이화여자대학교 대학원
2020학년도
박사학위 청구논문

Versatile Encountered-type Haptics for Immersive VR Environments Using Collaborative Robot

컴퓨터공학과

Yaesol Kim

2021

Versatile Encountered-type Haptics for Immersive VR Environments Using Collaborative Robot

이 논문을 박사학위 논문으로 제출함

2021 년 7 월

이화여자대학교 대학원

컴퓨터공학과 Yaesol Kim

Yaesol Kim 의 박사학위 논문을 인준함

지도교수 김영준

심사위원 민동보

오유란

류석창

전석희

김영준

이화여자대학교 대학원

Table of Contents

List of Figures

List of Tables

Abstract

Due to the surging interest in immersive virtual reality (VR), various types of haptic systems have been proposed and researched. Among the diverse form factors, the encountered-type haptic system provides free-to-touch and move-and-collide haptic sensation to a user while allowing natural interaction. According to the characteristics of the system that do not maintain contact with the user, the haptic manipulator's reachability with respect to the user's workspace needs to be considered for encountered-type haptic rendering. In addition, since the encountered-type haptic system allows a user to interact with virtual objects with one's bare hand, the system needs a haptic rendering algorithm that provides both kinesthetic and tactile feedback.

In this dissertation, an encountered-type haptic system that guarantees free and natural interaction while increasing the immersion to VR is introduced. To realize the research objective, a versatile haptic rendering method is proposed for representing various haptic forces and textures, active and passive haptic feedbacks, and haptic feedback for static and dynamic virtual objects. To provide encountered-type haptic feedback that renders various physical properties, a seven degrees of freedom (DoF) collaborative robot is used as a haptic manipulator, and a per-plane reachability map is proposed to check the reachability of a haptic manipulator. To enrich a VR experience by providing tactile feedback, a roughness synthesizing method for an encountered-type haptic display is proposed. For the robust haptic rendering, we also propose a contact prediction method to directly generate a haptic response by inferring the manipulator trajectory from the observation of human hand gestures.

Firstly, the encountered-type haptic system using a manipulator of seven-DoF is introduced to haptically and visually simulate an indoor VR environment with static and dynamic virtual objects such as vertical walls and revolving doors. In runtime, our system tracks user hand motion, and the robotic manipulator plans a trajectory for the end-effector with a rigid rectangular board attached 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 objects by holding the position of the rigid board corresponding to a vertical wall. For the dynamic objects, 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 a touch sensation to the surrounding environment.

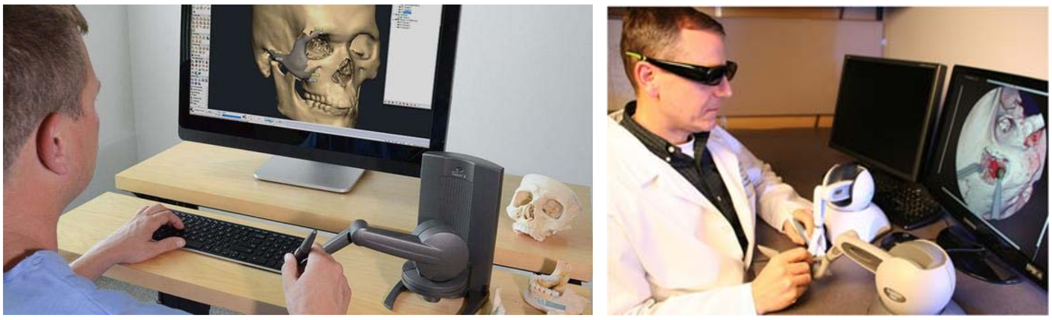
To provide cutaneous feedback along with kinesthetic feedback, inducing haptic-texture sensation without complicated tactile actuators is challenging for encountered-type haptic rendering. We propose a novel texture synthesizing method for an encountered-type haptic display using the spatial and temporal encoding of roughness, which provides both active and passive touch sensation requiring no complicated tactile actuation. Focused on macro-scale roughness perception, we geometrically model the textured surface with a grid of hemiellipsoidal bumps, which can provide a variety of perceived roughness as the user explores the surface with one’s bare hand. Our texture synthesis method is based on two important hypotheses. First, we assume that perceptual roughness can be spatially encoded along the radial direction of a textured surface with hemiellipsoidal bumps. Second, perceptual roughness temporally varies with the relative velocity of a scanning human hand with respect to the surface. To validate these hypotheses on our spatiotemporal encoding method, we performed psychophysical user tests and verified the main effects of spatiotemporal encoding of a textured model on the user’s roughness perception. Our empirical experiments imply that the users perceive a rougher texture as the surface orientation or the relative hand motion increases. Based on these findings, we show that our visuo-haptic system can synthesize an appropriate level of roughness corresponding to diverse visual textures by suitably choosing encoding values.

To generate a robust haptic response instead of relying on a deterministic contact prediction algorithm in VR, an end-to-end haptic response generation method is required. Since the haptic rendering system highly depends on the deterministic contact prediction between the virtual object and the user's hand, the failure of the contact prediction in VR leads to the failure of the entire haptic rendering system. We propose a learning from demonstrations method to predict the contact configuration solely by observing the human hand motion and no longer dependent on the conventional contact prediction approach, which uses ray shooting and projection in a VR environment. To generate the training data, which consists of human-robot interactions, we integrate three independent systems: a robot simulator, a VR environment, and a real-world tracking system. Then, we record the joint trajectories of the haptic manipulator simulated by the robot simulator and the cartesian trajectories of human fingertips and the center of the palm. The training data is represented within a basis function space and encoded as a probabilistic representation by learning from demonstrations. Using this probabilistic distribution as a prior, the system observes the partial trajectory of user hand motion and performs Bayesian filtering to infer the next hand motion and generate robot response. We successfully demonstrate the end-to-end encountered-type haptic response generation for VR users by referring only to the user's hand gesture. The result shows that the hand motion of touching a virtual object is enough to infer the haptic response spatially and temporally.

I. Introduction

A. Background

As a study on the sense of touch, haptics is an important area by itself and also crucial in physics-based animation and robotics. In particular, due to the recent surging interest in virtual reality (VR) environments, natural haptic interaction that provides both kinesthetic and tactile feedback is sought after more than ever before as part of efficient human-environment interaction. Since Massie proposed the three-DoF force-reflecting desktop device called PHANToM in 1993 [1], haptic interfaces have been around for a long time in their early stage. Still, the most prevalent forms of force feedback interface are stylus-type haptic devices (Figure I-1). This type of interface provides precise and sophisticated touch and manipulation of virtual objects with six degrees of freedom. Still, it may hinder the user's feeling of immersion in VR since it reduces the user's freedom of movement due to the limited workspace of the device.

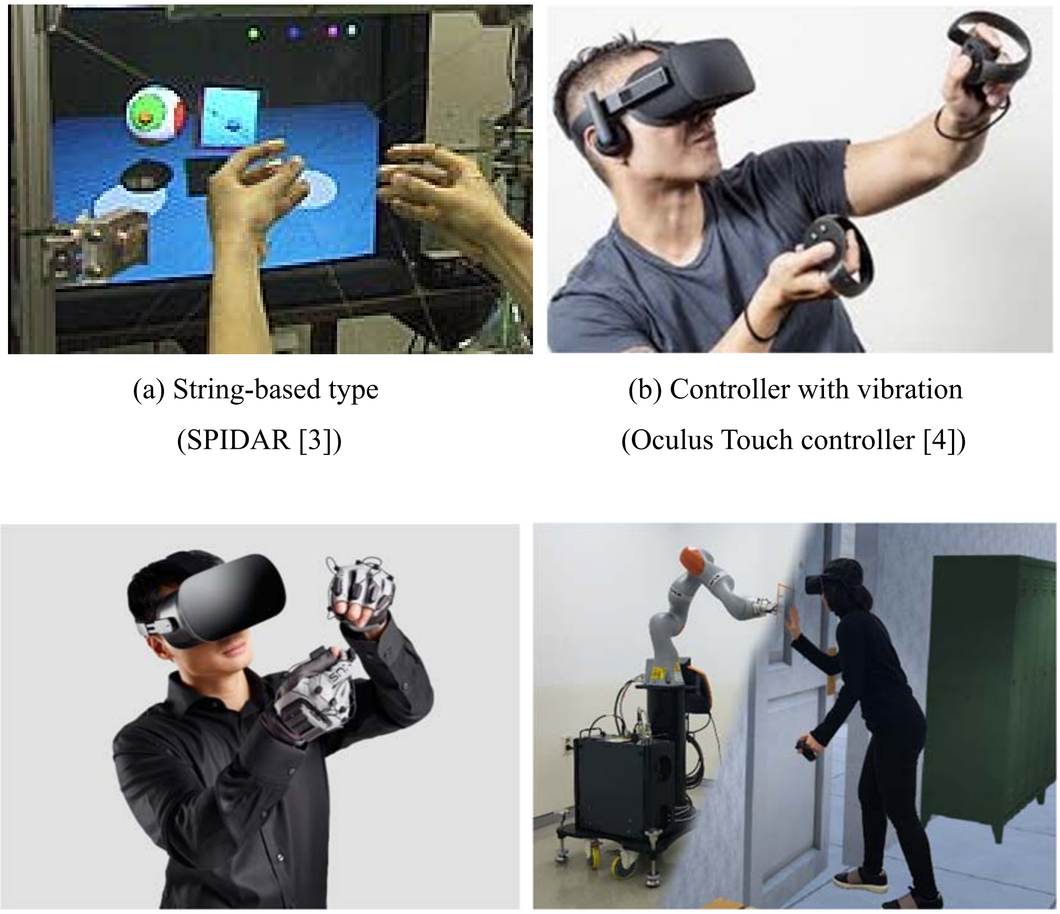


(a) Phantom Touch X

(b) Phantom Omni

Figure I-1. Conventional pen-shaped haptic devices

In recent years, a great deal of research has been carried out on effectively delivering fully immersive VR experiences with haptic interaction. In particular, due to the rise and availability of low-cost VR technology [2], disseminating and commercializing VR devices for general mass has drawn much attention from both academic and industrial sectors, which makes the interest in immersive VR research grow more than ever. Various form factors of the haptic interface have been proposed, such as a string-based type, VR controller with vibrational feedback, wearable type, and encountered type. Depending on the type of haptic interface, the user receives different types of feedback. For example, SPIDAR [3] could provide force feedback while the Oculus Touch controller [4] only provides vibration feedback.



(a) String-based type
(SPIDAR [3])

(b) Controller with vibration
(Oculus Touch controller [4])

(c) Wearable type
(Prime 2 Haptic [5])

(d) Encountered-type
(Ours [6])

Figure I-2. Various haptic interfaces according to device form factor

Among the cutting-edge haptic interfaces, an encountered-type haptic display or robotic graphics offers a VR interaction in a highly natural way by utilizing a robotic arm manipulator to deliver haptic feedback to users [7] without the need for conventional handheld interfaces to feel haptic feedback. Such an encountered-type haptic system allows users to experience “free-to-touch” and “move-and-collide” haptic sensations without requiring the user to wear a device or mount it on the user’s body. The natural interaction greatly improves VR immersion, but the existing encountered-type haptic system has a difficulty providing physical properties and textures of various virtual objects. To enable a fully immersive VR, the types of haptic feedback should be enriched and the range of haptic feedback should be sufficiently large in encountered-type haptics.

This dissertation proposes *the H-Wall system* to provide immersive VR through encountered-haptic feedback suitable for simulating indoor virtual environments using a robotic manipulator. Since 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. As a user explores the virtual environment at runtime, our system tracks the user’s head and hand movement and predicts the contact between the user's hand and a virtual object. Then, the robotic manipulator plans a trajectory for the end-effector with an attached rigid rectangular board to contact the hand to deliver a sense of touch. In the H-Wall system, we provide two types of virtual objects: static and dynamic. The force feedback is delivered only by the user’s active haptic exploration to represent the static objects while the haptic manipulator holds its configuration corresponding to the virtual object. For dynamic objects, the force feedback is delivered by both active and haptic exploration generated by altering the configuration of the objects.

Since the limited size of the haptic workspace can be a severe problem in 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. Our experiments show that this finite sampling of orientations works well, as the user is immersed in VR space. Also, the slight change in the user's orientation does not affect the believability of the H-Wall system. We successfully demonstrated to provide an illusion to the user in a virtual environment with touch sensation to the surrounding environment.

An encountered-type haptic display is often designed as a hybrid of kinesthetic and cutaneous mechanisms [8], delivering various physical characteristics of a virtual object to the user's bare hand. The haptic system allows users to haptically explore virtual objects and feel haptic feedback instantly and adequately through direct touch. Meanwhile, when a user uses one's bare hand for haptic interaction, the tactile information of a physical object can be immediately recognized by the user's nervous system. Therefore, to generate a truly immersive VR experience with haptic feedback, the encountered-type haptic display should provide force feedback with proper tactile sensation to the VR users.

A general approach for providing tactile feedback through an encountered-type haptic device can be taken by (1) preparing various physical textures in advance, (2) attaching one of the textures to the end-effector of a haptic device, and (3) switching the different haptic textures on demand [9], [10]. For this purpose, a special module can be attached to the end-effector, such as a tool-changing gripper or an extra tactile displaying device [11]. Although these are the feasible texture-rendering methods for an encountered-type haptic system, all the studies reported in the literature relied on adding extra modules to the manipulator or modifying the manipulator. Instead, in our approach, we attempt to deliver haptic-texture sensations by synthesizing tactile feedback through the spatiotemporal encoding of textures.

Accordingly, this dissertation aims to propose a novel approach for synthesizing surface roughness, which is one of the most prominent dimensions of perceptual space for tactile texture [12]. We model a textured surface by spatially encoding roughness along the radial direction of the surface. Based on the theory of roughness perception, we encode the roughness by using embossed bumps and modulating bump width and distance between bumps depending on the direction on the surface. Considering that ellipses vary in width and distance between them depending on their direction, we employ a grid of hemiellipsoidal bumps to model the surface texture and use the radius as a modeling parameter. We conducted an experimental study with 25 subjects to test our hypotheses that our roughness-encoded spatial model can render various perceptual levels of roughness, and the roughness can be temporally encoded by varying the velocity of a spatial model relative to the user's hand motion. As a result, we show that our encountered-type haptic system renders diverse roughness by adequately choosing the values of spatial (orientation) and temporal encoding (velocity) as rendering parameters. Texture roughness is provided both in a passive perceptual sense (i.e., the manipulator is dynamic while the user's hand is static) and in an active perceptual sense (i.e., the user's hand is dynamic while the manipulator is static) [13].

Since the encountered-type haptic rendering highly depends on the contact prediction process, the whole haptic rendering system stops working when the contact prediction fails in VR. Moreover, when the complexity of the VR environment or the object surface geometry is high, the contact predicting method using ray shooting and projection in VR miscalculates the haptic manipulator's trajectory. This dissertation addresses a method of generating a haptic manipulator's trajectory without relying on contact prediction in VR through learning from demonstrations. The contact prediction model is trained end-to-end, only with human-robot interaction data. By generating encountered-type haptic responses solely by partially observing the user's interactive motion, we could generate end-to-end haptic feedback without relying on a contact prediction in a VR environment, and the computational workload of the haptic system can also be reduced.

B. Research Objectives and Contributions

The main objective of the dissertation is to propose an encountered-type haptic system that improves the user's immersion into VR by enabling the user's free and rich visuo-haptic interaction and providing various haptic feedback. In order to provide users with a free and rich VR experience through the encountered-type haptic system, the haptic sensation coupled with large haptic space, various haptic sensations, and robust haptic response is required. To fulfill this research goal, the questions that we want to address are three-fold:

- How can we enlarge the encountered-type haptic system that provides various physical properties of virtual objects both haptically and visually?
- How can we provide tactile feedback on the textures of the various virtual objects?
- How can we generate encountered-type haptic responses with robust contact prediction for multiple virtual objects in VR?

To answer the first question, we proposed a haptic rendering method using an off-the-shelf collaborative robot of seven-DoF as a haptic device and enlarged the haptic feedback of virtual objects dramatically. By using a seven-DoF redundant robot as a grounded haptic manipulator, the range of stiffness we can provide is infinitely increased, and we can also vary the damping and inertia of the virtual objects. Furthermore, by solving the redundancy resolution problem, we can change the robot configuration with internal motion even if the end-effector is fixed. This allows the haptic system to enlarge the range of forces and speed, which the end-effector of the manipulator can generate, and the range of physical properties that can be provided to the user can be further increased. Since solving the redundancy resolution problem is beyond the scope of this study, we focus on providing haptic feedback within a predefined speed and force range. Also, to provide users seamless haptic feedback, we propose the per-plane reachability maps to check if the manipulator can reach the contact prediction point in real-time. In case the haptic manipulator is not reachable at the contact-predicted configuration, we restrict the user's hand movement in VR and minimize the negative effects of missing haptic feedback so as not to spoil the user's immersion.

In addition to providing the physical properties of various virtual objects, we propose a texture synthesizing method based on a robotic encountered-type haptic system to address the second question. For enriching the visuo haptic experience in VR, the encountered-type haptic system should provide tactile feedback that matches the visual texture. Given that the encountered-type haptic system allows for bare-handed interaction, the system should provide both kinesthetic and tactile feedback so as not to interfere with a user's immersion. While kinesthetic feedback is provided by allowing a user's free-to-touch and move-and-collide haptic feedback, tactile feedback representing the virtual object's surface information requires physical surface objects that correspond to each virtual surface or additional tactile device for tactile rendering. However, adding a tactile system to an encountered-type haptic system makes the system bulky and expensive. Using our texture roughness synthesizing method based on the spatiotemporal encoding of surface roughness, we can render multiple levels of roughness to the VR participants.

For the last question, we propose the robust contact prediction method. To provide haptic feedback corresponding to multiple objects with their own physical properties, it is important to predict which virtual object will be touched and the contact surface configuration. The existing encountered-type haptic response method relies on a deterministic contact prediction that uses ray casting and projection in a VR environment to locate the end-effector of the haptic device. To generate a haptic manipulator's trajectory robustly and independently from a contact prediction result in a VR environment, we train a contact prediction model using the human-robot interaction data as training data for learning for demonstrations. The system predicts contact configuration and determines the pose of the manipulator based on human hand observation using the trained model. For end-to-end learning, we recorded the encountered-type haptic interaction demonstrations in a simulation environment and leveraged them to encode a probability distribution model of all the DoF of interacting agents, including robot joint angles and human hand position. We utilize the Bayesian Interaction Primitives [14] and show that it is able to infer the contact configuration in runtime and does not violate the immersion of the user.

In summary, the main contributions are summarized as follows:

Various physical properties of virtual objects

- We propose an encountered-type haptic rendering system that provides various physical properties of virtual objects supporting both passive and active haptic feedback by using a seven-DoF human-robot collaborative robot.
- We propose optimization-based per-plane reachability maps for a robot manipulator.

Various textures of virtual objects

- We propose a novel texture synthesizing method by encoding spatiotemporal roughness and rendering using geometric grids of hemiellipsoidal bumps, which does not require complicated tactile actuation for encountered-type haptic displays.
- We empirically show evidence from psychophysical user studies that various levels of perceptual roughness can be rendered in a passive and active manner by controlling the orientation and velocity of the roughness encoded surface.

Robust contact prediction method for multiple virtual objects

- We generate an end-to-end contact prediction model that predicts the contact configuration by observing the user's hand gesture of touching a virtual object.

C. Organization

The remainder of this dissertation proceeds as follows. In Chapter II, we survey the related works on encountered-type haptics, the encountered-type haptic workspace considering the robot reachability and VR environment, texture rendering method for encountered-type haptics, and haptic response generation by learning from demonstration. In Chapter III, we describe our encountered-type haptic system by component. In Chapter IV, we describe per-plane reachability maps based on optimization-based computation and explain how to use the maps in runtime. The user experiments of our encountered-type haptic system using per-plane reachability maps are also described. In Chapter V, we describe the texture roughness synthesizing method using spatiotemporal encoding. Empirical experiments are presented to support our texture-encoding hypotheses, and our discussion is also described. In Chapter VI, we present the end-to-end contact prediction for encountered-type haptic response generation using learning from demonstration. We present the demonstration example that our system predicts the contact configuration by observing the user's hand motion. We conclude the dissertation in Chapter VII along with a plan for future work.

II. Related Work

In this chapter, we first explore an encountered-type haptic display, including grounded interface and non-grounded interface. We also survey works relevant to the encountered-type haptic workspace, mainly focusing on robot arm-type haptic manipulators. Then we study texture rendering methods for when humans touch the surface with their bare hands. Lastly, we briefly survey studies on learning from demonstrations focused on the method using probability distributions.

A. Encountered-type Haptics

1. Encountered-type Haptic Display

Encountered-type haptics or robotic graphics, originally conceived by McNeely [7], directly delivers haptic feedback to the user by physically presenting an object proxy, unlike conventional haptic devices relying on motor actuation. Tachi et al. proposed a similar concept using haptic shape display, which approximates the shape of *virtual haptic space* [15]. To support the free-to-touch and move-and-collide haptic sensation, Yokokohji et al. proposed a method for realizing visual/haptic interfaces, called the what-you-see-is-what-you-feel (WYSIWYF) display [16][17]. They used a PUMA 560 manipulator as an encountered-type haptic interface and vision-based tracking for controlling the robotic manipulator with a motion-command type haptic rendering algorithm (Figure II-1).

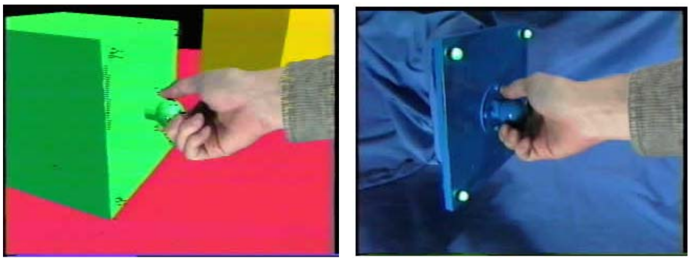
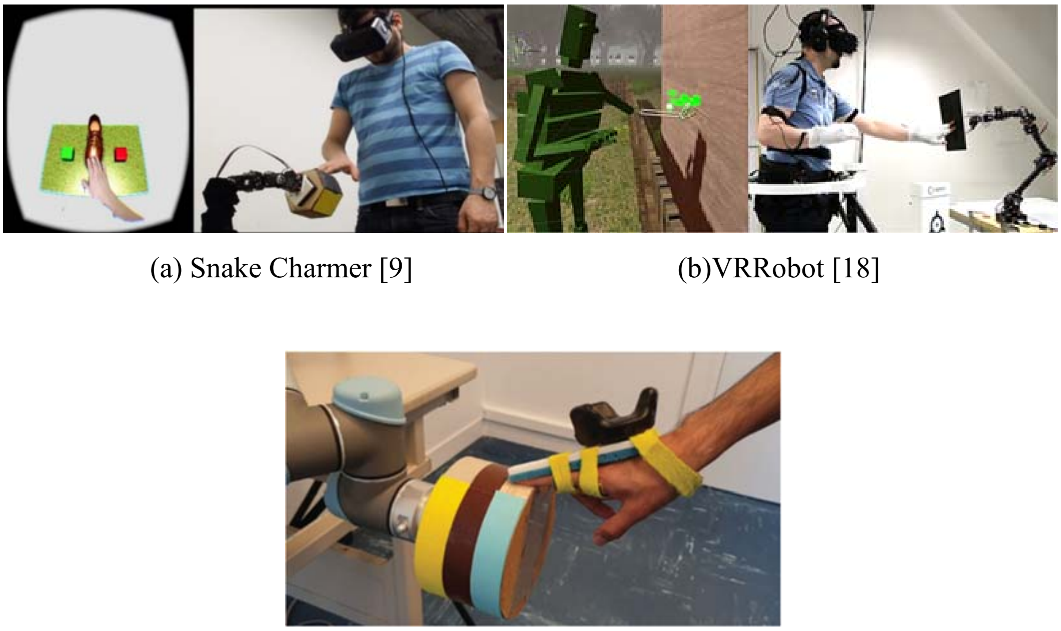


Figure II-1. WYSIWYF display [16]

Deriving from early theoretical proposals and implementations, robot manipulators provide prop-based haptic feedback while the user wears a head-mounted display (HMD) to experience VR. Araujo et al. [9] proposed the Snake Charmer to provide various tactile properties of objects such as shape difference, surface characteristics, and even temperature. Vonach et al. [18] presented a fully immersive VR system called VRRobot, which utilizes robot actuation to provide haptic feedback for static virtual objects in a passive sense. Mercado et al. proposed an approach for providing an infinite surface, called ENTROPiA, by rotating a cylindrical prop that serves as a virtual surface [19]. To locate the haptic props attached to the robotic arm, path planning and motion planning are proposed [20], [21].



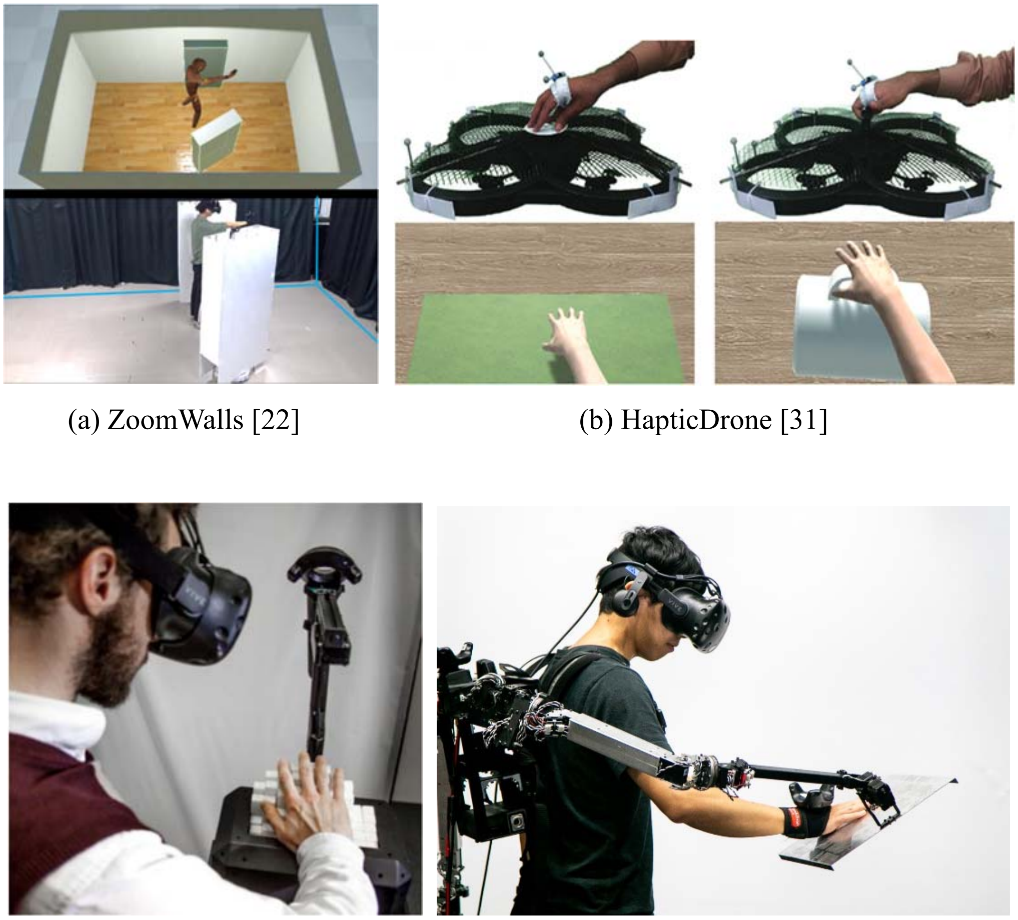
(a) Snake Charmer [9]

(b)VRRobot [18]

(c) ENTROPiA [19]

Figure II-2. Encountered-type haptic displays with a robot arm

Depending on the characteristics of the haptic feedback to be provided, various types of devices are used, such as a mobile robot [22]–[25], mobile shape display [26], [27], wearable devices [28], [29], drone or quadcopters [10], [30]–[32]. Taking advantage of robotic arm-type and wearable-type haptic displays, EncounteredLimbs [33] enables stiff feedback while maintaining the spatial freedom of haptic interaction using a wearable robotic limb.



(a) ZoomWalls [22]

(b) HapticDrone [31]

(c) Mediate [26]

(d) EncounteredLimbs [33]

Figure II-3. Various type of encountered-type haptic display

In this dissertation, we used a seven-DoF collaborative robot equipped with a panel on its end-effector as an encountered-type haptic display. This selection enables safe, large-scale, and both kinesthetic and tactile haptic rendering.

2. Passive and Active Haptic Exploration

Haptic exploration can be divided into two categories: active exploration and passive exploration. During active exploration, the user actively moves one's body to feel the haptic feedback, and in passive exploration, the haptic device is controlled to provide haptic feedback. Rodríguez et al. show active haptics is clearly superior to passive haptics for shape recognition tasks [34]. This is mainly due to the nature of purposive actions, which ease the perception of object solidity.

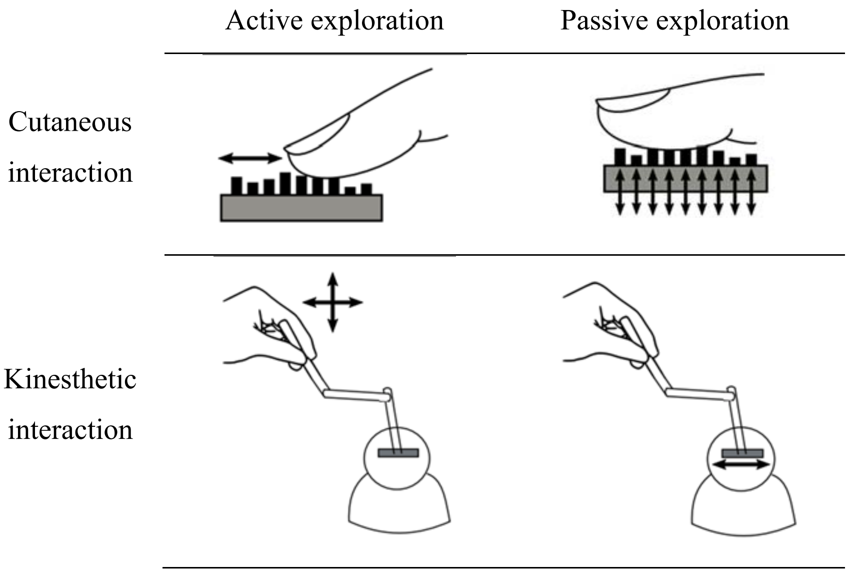


Figure II-4. Haptic interaction types and exploration modes [34]

The encountered-type haptics has been nurtured from active and passive haptic approaches as the user's direct touch delivers haptic feedback. The passive and active haptics are used separately for the robotic actuator to change pose just in time and allow the user to physically touch the standing proxy of virtual objects [9], [18]. The mobile shape display utilizes the mixed method [26], [27]. Mercado et al. used the hybrid method that the encountered-haptic device actively rotates its end-effector while the user also actively explores virtual surfaces [19]. Our work supports both active and passive haptic exploration in a hybrid way so that the types and range of haptic sensations that users can feel have been expanded.

B. Encountered-type Haptic Workspace

1. Robot Manipulator Reachability

The workspace of a robot manipulator can be classified into the dexterous workspace, reachable space, and orientation workspace [35]. Various workspace analysis methods have been proposed based on this classification. The sampling-based method is usually utilized to analyze reachable space. Zacharias et al. propose a capability map representing kinematic reachability and the directional structure information of a robot arm [36]. They sample points inside the working envelope and solve the reachability and directional structure of a point set. A manipulator can deduce places that are easy to reach and plan an optimal path using a capability map. Dong et al. propose 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 [37]. Reachable space generation using the Monte Carlo method is proposed by Guan and Yokoi [38]. They present a numerical approach using a random sampling method to avoid the conventional analytical method, which 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 [39]. They introduce the oriented reachability map based on inverse reachability data for the target pose. Zacharias et al. [40] propose an algorithm to position a mobile manipulator to generate 3D trajectories using a discrete representation of the reachable workspace.

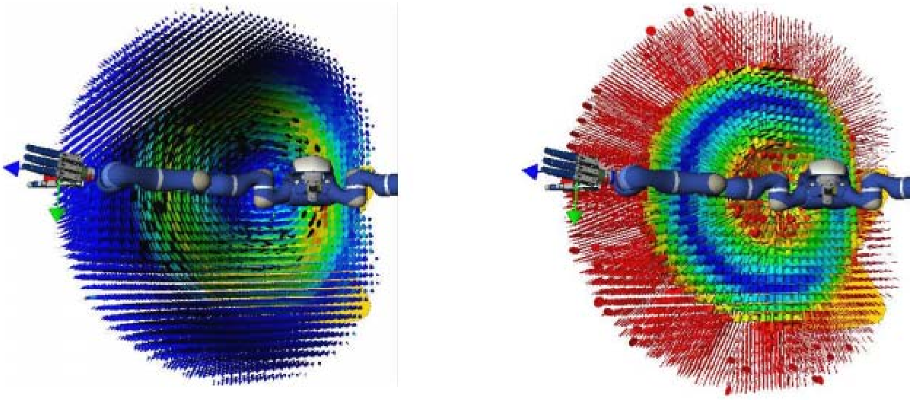


Figure II-5. Approximated structure of capability map using cone (left) and cylinder (right) [36]

In this study, we propose a new approach to calculate and construct robot reachability maps, called per-plane reachability maps. Our method approximates the reachability map to reachable boundary points and constructs the maps per pre-define orientation of the end-effector.

2. Visual Redirection

To fill the gap between the real-world and virtual world, methods of distorting VR are proposed, such as impossible space [41], flexible space [42], and redirected walking [43]–[46]. Sum et al. proposed a general method to match a given pair of virtual and physical worlds for immersive VR [47]. The limited amount of space distortion shown to be acceptable for VR navigation [48], [49] and haptic cues can be used to improve the effect of redirected walking [50], [51].

Since our system uses reachability maps optimized by a sampling-based method, there is a gap between robot reachable space in the real and virtual worlds. To remove this gap by distorting user's orientation in VR, we utilize the redirection approach.

C. Haptic Texture Rendering

1. Haptic Texture Display

Since the vibration of a haptic device can render the natural and rich texture information at a low cost, the vibration-based method is more widely used than other tactile rendering methods such as modulating static pressure, skin stretch, and friction [52]. To render the texture roughness with a tool, Romano and Kuchenbecker proposed a method for recording, modeling, and recreating texture contact vibration using a tablet-based haptic device [53]. Ujitoko et al. presented a cross-modal modulating method of vibrotactile roughness perception using pseudo-haptic effect [54]. For bare-finger interaction, a touchscreen with electrovibration [55] or conjunction of vibrotactile and electrostatic display [56] are proposed to render the texture roughness. Although these tactile displays could be integrated into the encountered-type haptic system, it may increase the system's complexity and cost.

To render the texture information within the encountered-type haptic system, multiple textures are attached to the haptic display [9], [10], [19]. Snake Charmer [9] uses a rotational palette with multiple textures attached. Also, this system allows the palette to be changed so that more types of textures can be provided. TeslaMirror [57] was proposed as a hybrid tactile display with multistimulus feedback by combining encountered-type haptic device and texture display with electrotactile and vibrotactile devices.

In this study, beyond providing a limited number of textures and avoiding using an additional tactile device, we propose the texture synthesizing method to provide a more diverse range of haptic textures in real-time.

2. Texture Roughness Perception

When the user touches a virtual object, it is known that the object's position alone is not sufficient to provide reliable tactile feedback without using additional physical cues [58]. Surface texture is essential for a human to perceive the property of an object that is constructed multi-dimensionally [59], [60]. Three prominent psychophysical features of tactile texture, including roughness (rough/smooth), hardness (hard/soft), and warmness (warm/cold), are used to define the perceptual dimensions of a tactile texture [12]. Among these dimensions, roughness is the most prominent one and thus has been intensively studied [61], [62].

In the roughness perception study, the size and spacing of tactile elements have been shown to play a crucial role in the subjective sensation of roughness. Lederman and Taylor showed that an increase in the width of *texton* leads to a decrease in perceived roughness [63]. The relationship between the texton spacing and the magnitude of subjective roughness is depicted as an inverted U-shaped curve [64] and peaks near a texton spacing of $3.5mm$. Dépeault et al. showed that one critical measure of roughness is the *dot spacing* along the scanning direction [65]. Connor and Johnson showed that the perceived roughness increases as the dot spacing increases along the scanning axis or across the scanning axis [62]. These results allude to a possibility that a single object can encode multiple roughness values by arranging elements with different inter-distances according to the scanning direction. Later in Chapter V, we will exploit this possibility to present and prove hypotheses and leverage them to spatially encode haptic textures for our system.

Although it is argued that user movement has a negligible effect on subjective roughness perception [66], [67], the influence of a temporal cue on roughness has been demonstrated in recent works. Gamzu and Ahissar showed that temporal cues might provide an alternative channel of information for bare-finger texture perception [68]. Smith et al. showed that kinetic friction, governed by a ratio of normal forces to tangential forces, is a significant determinant of roughness perception and implied that the temporal cues play a role in the roughness perception of macro-texture [69]. Under a precisely controlled condition of passive linear motion, Cascio and Sathian demonstrated that the perceived roughness also depends on scanning velocity [61].

Optimal roughness perception depends on the user's hand motion [70], [71], and is indispensable to neural activation in space and time [61]. The relative motion produced by an object or user's hand is required for roughness perception [66]. Lederman found that the estimated magnitude of roughness does not depend on whether the associated hand motion is active or passive [13]. In our system, we will take advantage of both active and passive motion of the user's hand to generate a different magnitude of roughness.

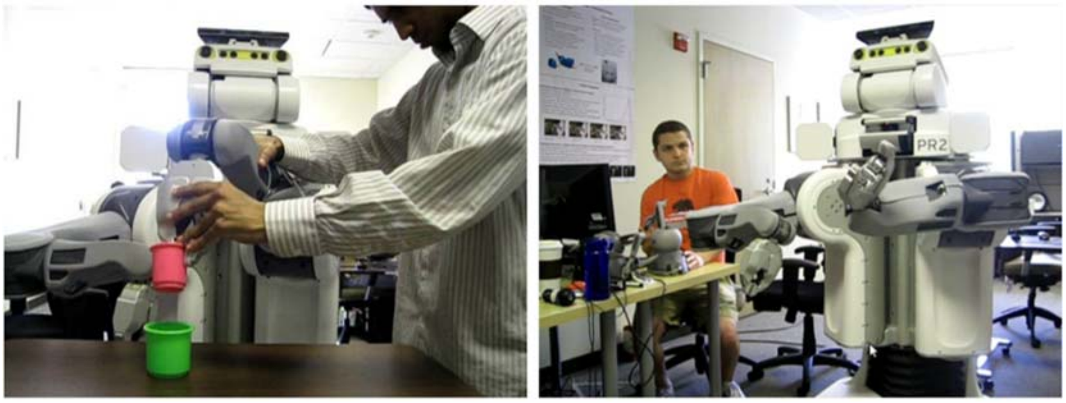
D. Contact Prediction for Encountered-type Haptics

The encountered-type haptic response is usually generated by the following process: 1) imports the user's tracking information into the VR environment, 2) predicts the contact point where the contact is most likely to occur in VR, 3) maps the contact predicted configuration back to the real world, and 4) places the haptic display in the corresponding configuration. To predict the contact point for encountered-type haptics, the distance between the user hand and VR object [9], [72], or the user's gaze direction with hand motion are often used [22], [73]. Recently, Yixian et al. proposed a contact prediction method using a collision detection approach with user intention inference [22]. They predict the contact surface by detecting collision between the virtual object and the bounding box around the user and by using a support vector machine (SVM) based on the user's head orientation and movement speed. The conventional approach for the encountered-type haptic response is the deterministic method in which the robot follows the contact prediction point.

By excluding the VR environmental information from the encountered-type haptic system, the haptic response could be generated robustly by defining the encountered-type haptic interaction as a simple human-robot interaction and teaching a robot to generate haptic feedback. Imitation learning or learning from demonstration has been proposed as a natural and straightforward way to teach robots [74]. From the teaching of a single robot motion based on the demonstrations of the desired behavior, learning from demonstrations is extended to the human-robot interaction scenario [75]–[77]. Interaction Primitive (IP) [78] generalizes the learning from demonstrations to a human-robot interaction scenario. As a special type of Dynamic Motor Primitives (DMPs) [79] representing a trajectory of human or robot movement, IP represents a joint activity of two interaction partners and learns probability distributions over joint actions between two interacting agents. IP is extended to the Bayesian Interaction Primitives (BIP) [14] and ensemble Bayesian Interaction Primitives (eBIP) [80]. By formulating IP as a simultaneous localization and mapping

(SLAM) problem, BIP shows performance improvement in space-time inference. We utilize BIP to encode the human-robot trajectory during the encountered-type haptic interaction and generate the end-to-end haptic response at runtime.

To generate training data by recording the HRI scenario, two interaction methods are generally used: kinesthetic teaching and teleoperation [80].



(a) Kinesthetic teaching

(b) Teaching by teleoperation

Figure II-6. Two interaction methods for robot teaching by demonstration [81]

The kinesthetic teaching method is an intuitive method for teaching a robot, but it is difficult to generate the desired motion when operating a robot with high DoF. Teaching by teleoperation, which is a challenging problem itself, requires a complicate master-slave teleoperation system. Especially for generating human-robot interaction data [82], two separate recordings are needed to get one demonstration trajectory, one for a human operator only and the other for a robot manipulator. In our system, we propose the demonstration generating method by using the robot in a simulation environment while tracking a user in the real world who explores the VR environment.

III. Encountered-type Haptic System for Large VR Environments

In this chapter, we describe the overall architecture of our H-Wall system that enlarges the encountered-type haptic experience for the VR environment.

A. System Overview

The encountered-type haptic system provides visual and haptic feedback to the user simultaneously. For our encountered-type haptic system, we build the visual rendering component and haptic rendering component considering the user in the loop interaction and control them asynchronously with three modules: user tracking, contact prediction, and reachability checking (Figure III-1).

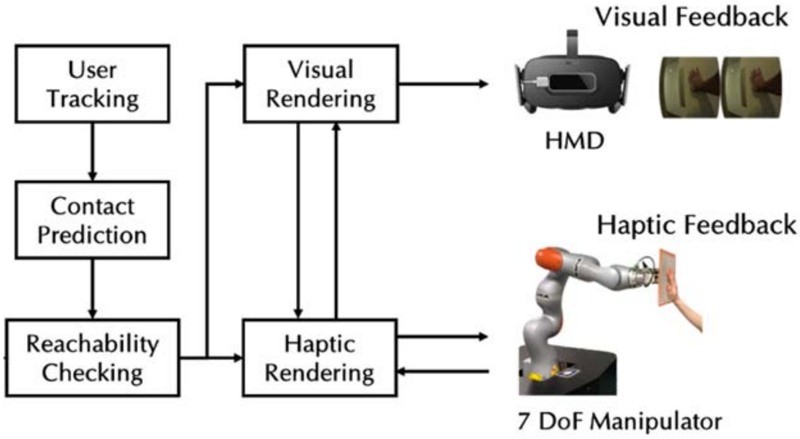


Figure III-1. Overview of an encountered-type haptic system

The user tracking module tracks the user's head and hand and transforms the data from the tracking sensor coordinate system to the VR coordinate system (Chapter III. B). Considering the user's virtual proxy hand and virtual objects in VR environment, the contact prediction module predicts when and which virtual object is most likely to be contacted by the user and estimates a contact configuration of the object, including contact point, contact normal, and surface texture roughness (Chapter III. C). Then, the system checks manipulator reachability to the estimated contact configuration, referring to the precomputed reachability maps (Chapter III. D). The reachability checking module ensures that the end-effector of the robot reaches the user hand position maintaining the configuration of the virtual wall before providing user visual feedback and haptic feedback.

For haptic rendering, a seven-DoF off-the-shelf collaborative robot manipulator operates as a haptic device. The manipulator relies on contact forces exerted by the user to provide haptic feedback corresponding to the estimated contact configuration (Chapter III. E). In our system, an indoor virtual environment consisting of furniture is rendered to the HMD that the user wears. Using user tracking data, the user's virtual proxy hands are also rendered in real-time (Chapter III. I.F). The states of a user avatar in virtual reality 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.

B. User Tracking

To determine the end-effector location of the manipulator and the location of an avatar hand, the user's head and hand are tracked in real-time and represented in the same frame in which the configuration of the haptic manipulator is represented. The user tracking system for the encountered-type haptic system can be classified into three categories: built-in, marker-based, and marker-independent tracking systems [83]. To increase the generality of the system, we build the marker-independent tracking system in two different ways through combinations of tracking sensors.

Firstly, we use an HMD tracker to track the six-DoF pose of the user's head and a fixed RGBD camera to identify the joint angles of the user's body, such as the head and hands (Figure III-2). More specifically, the user's geometry data is captured as a point cloud, and the head center and the palm of the hand are detected and tracked. Finally, the head and hand positions are transformed from the RGBD camera frame to the robot's base frame. Since the RGBD camera is located in a fixed position, this setup of the tracking system enables coordinate system transformation directly from tracking space to robot space and supports the tracking of the user's entire body as well as the user's hand tracking. However, the system may lose tracking when the human body is occluded by the robotic manipulator, or vice versa, and thus the tracking space is limited to the sensor's fixed field of view.

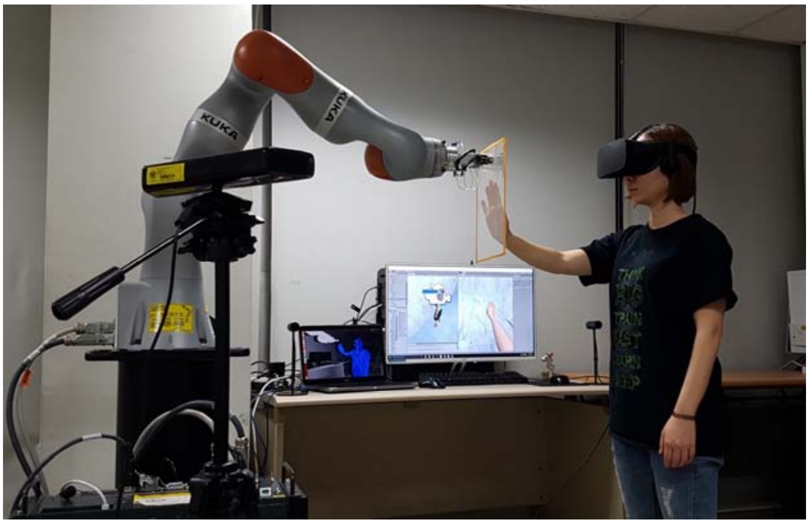


Figure III-2. An encountered-type haptic tracking system with
fixed RGBD camera and HMD tracking system

When the occlusion issue is critical or a large tracking space is needed, we could choose the tracking system with HMD and HMD mounted sensors such as the IR motion sensor [84] or monochrome cameras [4]. In this case, the user's head position is tracked by an HMD tracker and transformed from the tracking sensor frame to the robot's base frame. The user's hand position and direction of movement are tracked by an HMD-mounted sensor and transformed from the motion sensor frame to the HMD tracker frame and, finally, transformed to the robot's base frame.

To reduce the hand tracking sensor noise, we used the midpoint smoothing algorithm [85]. By referencing earlier four hand positions, two positions are determined after two steps of midpoint smoothing are performed and yield a direction vector, which is used as an estimate for the user scanning direction.

C. Contact Prediction

In order for the encountered-type haptic display to arrive at the location of the virtual object the user wants to touch before the user does and then wait for the user to touch the display, our system needs to predict when the user's hand will contact virtual objects. However, this problem poses a significant sensing and control challenge: predicting where the user will want to touch in the virtual environment as the hand approaches an object, such that the haptic device can position and shape itself as needed in order to provide the desired haptic experience [8].

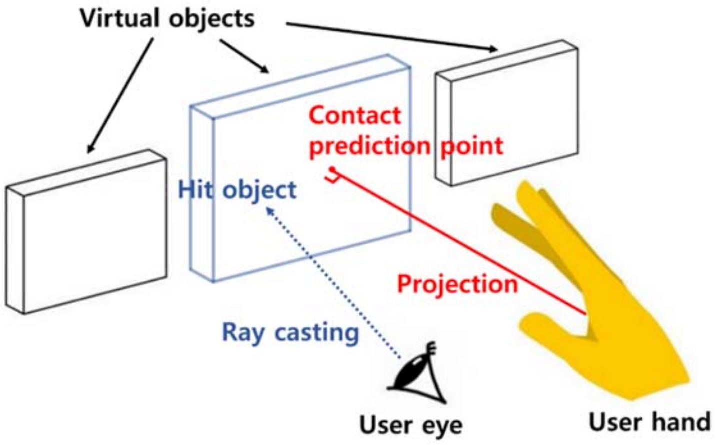


Figure III-3. Contact prediction using ray casting and projection

We will introduce the robust contact prediction proposed in Chapter VI, but here, we use a deterministic contact prediction method performed in a virtual world using user tracking data. To predict an object with a high probability of contact, we employ the concept of *a region of interest*. As shown in Figure III-3, we utilize ray casting for predicting the user's region of interest by shooting a virtual ray along the user's gazing direction. The closest hit object is considered a collision-imminent object, and its configuration along with associated surface texture roughness are retrieved. Then, the user's virtual hand is projected to the colliding object's surface so that the system can determine the contact position and orientation.

D. Haptic Manipulator Reachability

In our system, we precompute the reachability map of the encountered-type haptic manipulator (Chapter IV. B) and check the manipulator reachability in runtime. If the contact predicted configuration is inside the reachability maps, the system sets a goal pose of the haptic manipulator to the contact-predicted configuration. The position of the virtual hand is set in the virtual reality position as it is relative to the robot in the real world. When the contact predicted configuration is out of the reachability maps, the haptic system does not update the target position of the haptic manipulator and waits for the next contact prediction point.

E. Haptic Rendering

Two types of haptic rendering states can be distinguished: contact and non-contact. The contact state is determined by physical contact detection using a haptic manipulator. Torque sensors embedded in the manipulator's joints are used to detect the user's contact forces $\mathbf{f}$ in a standard way:

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

where $\mathbf{J}$ is the manipulator Jacobian. Then, we project $\mathbf{f}$ toward the vertical direction of the surface of the virtual object to obtain the orthogonal component $\mathbf{f}^{\perp}$ of $\mathbf{f}$. When $\mathbf{f}^{\perp}$ is higher than the pre-defined threshold, 3N in our implementation, the haptic state is set as a contact state. Otherwise, the system state is set as the non-contact state. While the non-contact state is maintained, the robotic manipulator is controlled only by position control. As shown in Figure III-4, the manipulator end-effector follows the target.

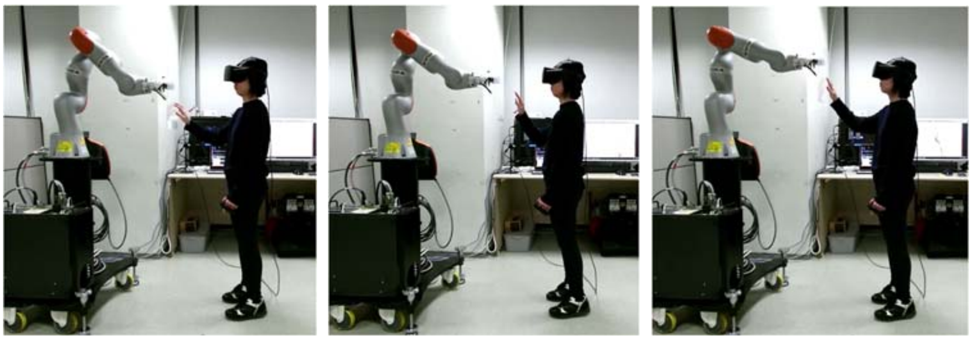


Figure III-4. Movement of haptic manipulator while in the non-contact state

During the contact state, the manipulator stops following the contact-predicting point and starts to provide the haptic feedback corresponding to the static or dynamic virtual object.

For the static object in VR, the haptic system provides user active haptic perception only. A seven-DoF manipulator relies on contact-force data measured from the user to hold its configuration for the user to touch a static virtual object.

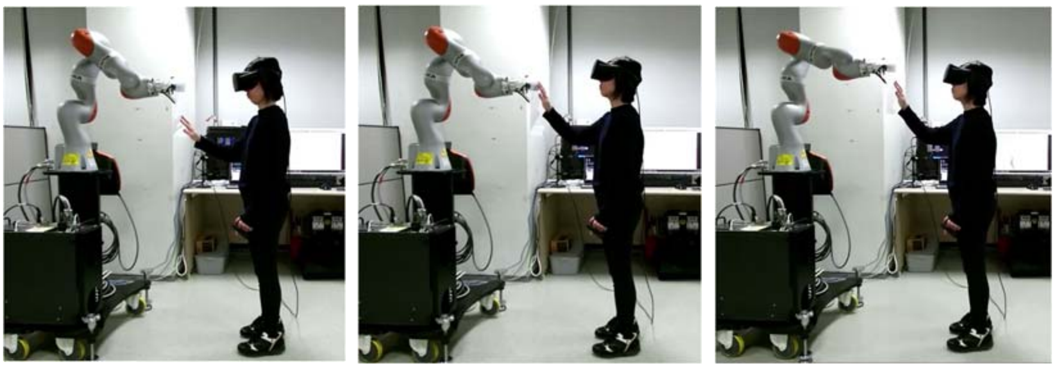


Figure III-5. Haptic interaction for the static object

For the dynamic object in VR, the haptic manipulator allows the user to push the virtual object. In this case, a seven-DoF manipulator is controlled by impedance control [86] and allows user active and passive haptic exploration. Figure III-6 shows an example of the dynamic haptic interaction where the user is trying to open the virtual door. The haptic manipulator holds its configuration corresponding to the virtual door until the contact force is detected, and when the user touches the haptic proxy of the virtual wall, the manipulator provides the user force feedback due to pushing the door open.

The haptic force rendered to a user is expressed by the following:

$$\mathbf{f} = k\delta\mathbf{x} + b\dot{\mathbf{x}}, \tag{III.2}$$

where $\mathbf{f}$ is the haptic force applied to the environment, k is the spring stiffness, $\mathbf{x}$ is the position of the haptic probe, b is the damping term [87]. In this dissertation, we simplify the force to Equation III.3 and provide dynamic object properties of the vertical surfaces by impedance control. The manipulator gives itself a margin $\delta\mathbf{x}$ in the normal direction of the attached panel. Then, the user can feel the impedance forces $\mathbf{f}$ in a normal direction of the virtual door:

$$\mathbf{f} = k\delta\mathbf{x} \ , \tag{III.3}$$

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

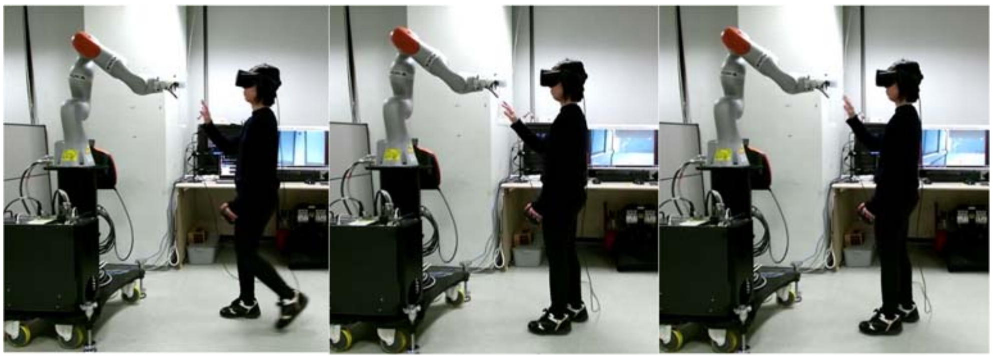


Figure III-6. Haptic interaction for the dynamic object

For the texture rendering, the passive and active haptic feedback is provided in a hybrid way. In this study, the texture roughness of the virtual object is rendered to the user (Chapter V.C). The haptic manipulator holds its configuration corresponding to the virtual object until the contact force is detected, and when the user touches and scans the virtual surface, the manipulator adjusts the orientation and velocity of the haptic manipulator as described in Chapter V.C.

F. Visual Rendering

In order to control a virtual avatar interactively, we employ a motion control interface using the Oculus Rift controller. Specifically, a user can control the avatar's position by joystick and rotate an avatar by pushing buttons in a controller. The user uses his/her left hand to operate a controller, while the user can feel haptic feedback freely by his/her right hand. The VR environment is visually rendered from the first-person view of the user through the HMD.

Inspired by redirected walking [43], [47], 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 $\mathbf{r_t}$ and $\mathbf{r_a}$ is calculated by corresponding β values, $\beta_a - \beta_t$. 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.

IV. Per-plane Reachability Maps for Encountered-type Haptic Manipulator

Just like any other haptic system, our encountered-type haptic system also has a finite workspace. However, the limited size of the 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 the 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 chapter, we explain our optimization-based methods to compute per-plane reachability maps for our encountered-type haptic system as well as how to use the maps at runtime.

A. 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 forward kinematics of the manipulator and is represented by a homogeneous matrix as follows:

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

where $\mathbf{R} \in \mathrm{SO}(3)$ 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:

$$\mathbf{p} = (p_\mathrm{x}, p_\mathrm{y}, p_\mathrm{z}) = \mathbf{f}_\mathbf{p}(q_1, q_2, \cdots, q_n), \qquad \text{(IV.2)}$$

$$\mathbf{r} = (\alpha, \beta, \gamma) = \mathbf{f}_\mathbf{r}(q_1, q_2, \cdots, q_n), \qquad \text{(IV.3)}$$

where $q_1, q_2, \cdots, q_n$ 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 $q_1, q_2, \cdots, q_n$ defines a three-dimensional reachability map of the manipulator.

In order to calculate reachability accurately, all possible positions and orientations in the 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 the reachability precisely, we merely approximate it by discretizing the orientations and positions and then extract and store its boundary points using optimization.

B. Per-plane Reachability Maps

In our encountered-type haptic 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 sub-workspace is sampled again into equidistant planes, called *per-plane reachability maps*, as illustrated in Figure IV-1. We also compute the boundary points of each per-plane reachability map.

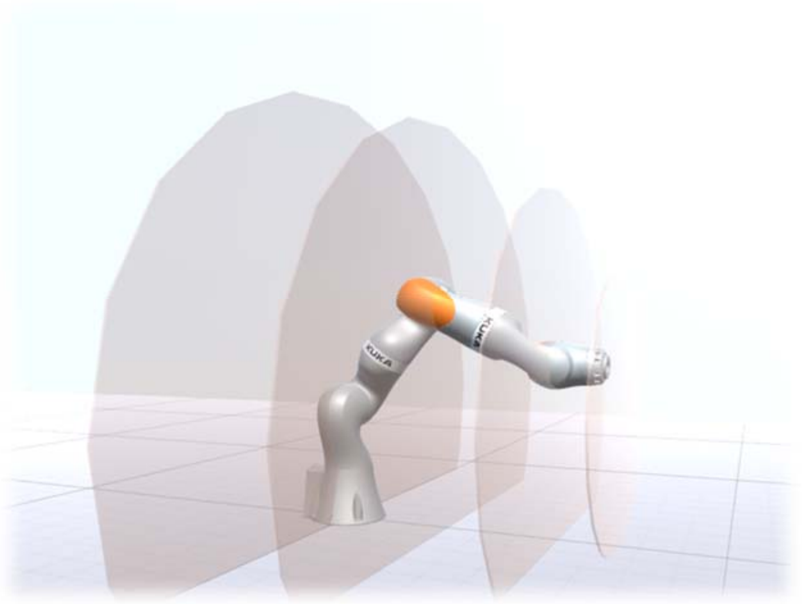


Figure IV-1. Illustration of four per-plane reachability maps of KUKA IIWA with a fixed end-effector orientation

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

$$\mathbf{T}_{\mathcal{E}} \cong \{\mathbf{T}_{\mathcal{E}}^{\mathbf{1}}, \mathbf{T}_{\mathcal{E}}^{\mathbf{2}}, \cdots, \mathbf{T}_{\mathcal{E}}^{\mathbf{i}}, \cdots, \mathbf{T}_{\mathcal{E}}^{\mathbf{n}_{\theta}}\}, \tag{IV.4}$$

where n_{θ} is the number of sampled orientations and

$$\mathbf{T}_{\mathcal{E}}^{\mathbf{i}} = \begin{bmatrix} \mathbf{R}(\alpha^{\mathrm{i}}, \beta^{\mathrm{i}}, \gamma^{\mathrm{i}}) & \mathbf{p} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \tag{IV.5}$$

with some fixed orientation $\alpha^{\mathrm{i}}, \beta^{\mathrm{i}}, \gamma^{\mathrm{i}}$ (Figure IV-2-(a)).

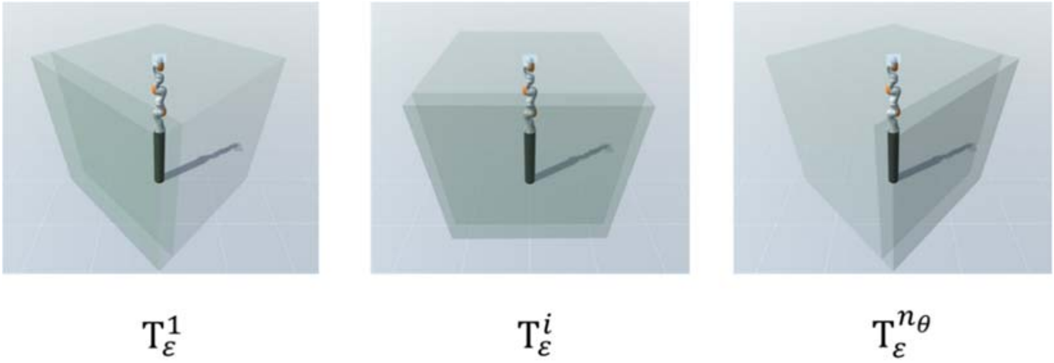
Then, $\mathbf{T}_{\mathcal{E}}^{\mathrm{i}}$ is further sampled into a discrete set of cardinality *M* as follows:

$$\mathbf{T}_{\mathcal{E}}^{\mathbf{i}} \cong \{\mathbf{T}_{\mathcal{E}}^{i_1}, \mathbf{T}_{\mathcal{E}}^{i_2}, \cdots, \mathbf{T}_{\mathcal{E}}^{i_k}, \cdots, \mathbf{T}_{\mathcal{E}}^{i_M}\}, \tag{IV.6}$$

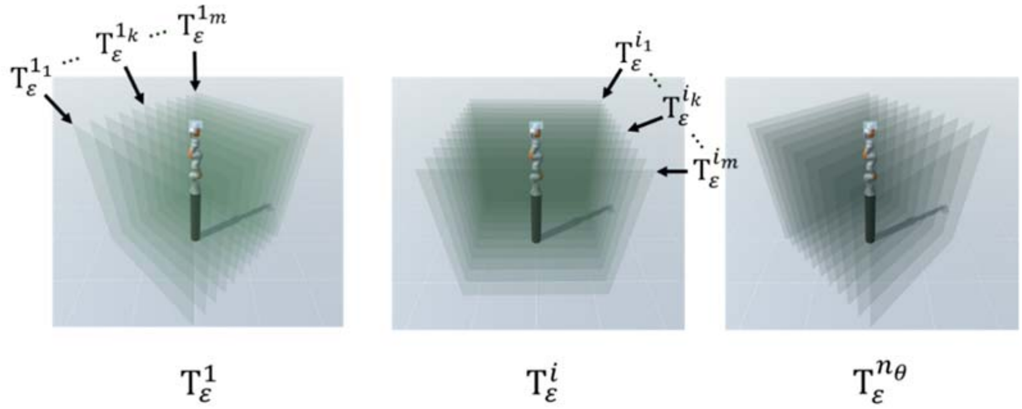
where *M* is the number of sampled planes and

$$\mathbf{T}_{\mathcal{E}}^{i_k} = \begin{bmatrix} \mathbf{R}(\alpha^{\mathrm{i}}, \beta^{\mathrm{i}}, \gamma^{\mathrm{i}}) & \begin{bmatrix} p_{\mathrm{x}} \\ p_{\mathrm{y}} \\ p_z^{i_k} \end{bmatrix} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \tag{IV.7}$$

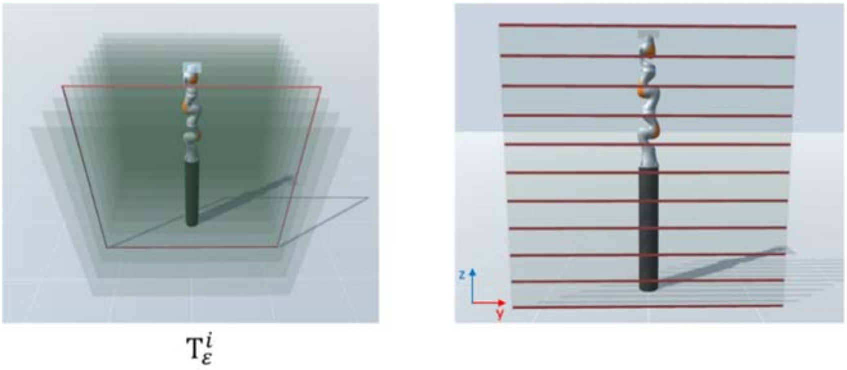
with some fixed value of $p_z^{i_k}$ (Figure IV-2.b). We refer to each of $\mathbf{T}_{\mathcal{E}}^{i_k}$ as a per-plane reachability map. In our implementation, we find the extremal values of $(p_{\mathrm{x}}, p_{\mathrm{y}}, p_z^{i_k})$ in $\mathbf{T}_{\mathcal{E}}^{i_k}$ and store them at a lookup table (Figure IV-2.c, Figure IV-2.d).



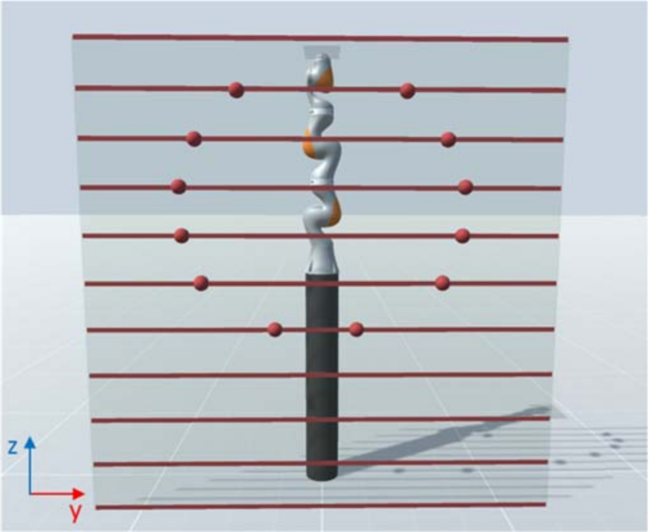
(a) Sample the workspace by n_θ different orientations and bound the workspace by a box



(b) Sample again into m equidistant planes



(c) Sample planes according to height



(d) Compute the boundary points by optimization

Figure IV-2. The concept of 3D workspace sampling

C. Optimization-based Computation

In order to calculate the per-plane reachability 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:

$$\mathcal{C}_{eq_i} = \begin{cases} \alpha = 0 \\ \beta = 0.5\pi \\ \gamma = \gamma^{\mathrm{i}} \\ p_{\mathrm{x}} - \tan(\gamma^{\mathrm{i}})\, p_{\mathrm{y}} = \mathrm{d_i} \\ p_{\mathrm{z}} = p_{\mathrm{z}}^{\mathrm{i_k}} \end{cases}, \qquad \text{(IV.8)}$$

where α and β are constants characteristic to represent vertical walls, γ^i defines 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 in 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:

$$p_{\mathrm{y}} = \mathbf{f}_{\mathrm{y}}(q_1, q_2, \dots, q_{\mathrm{n}}), \tag{IV.9}$$

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 multi-start method [88] to get optimal results. The multi-start 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 and divide the box by planes in parallel to the y- z plane of $\mathcal{W}$, and find the extremities of each plane. Since this boundary is conservative, some p_{y} contained in the box may not satisfy C_{eq}. We filter out these results from the reachability map by checking whether the optimization results satisfy the orientation constraints for each step of optimization.

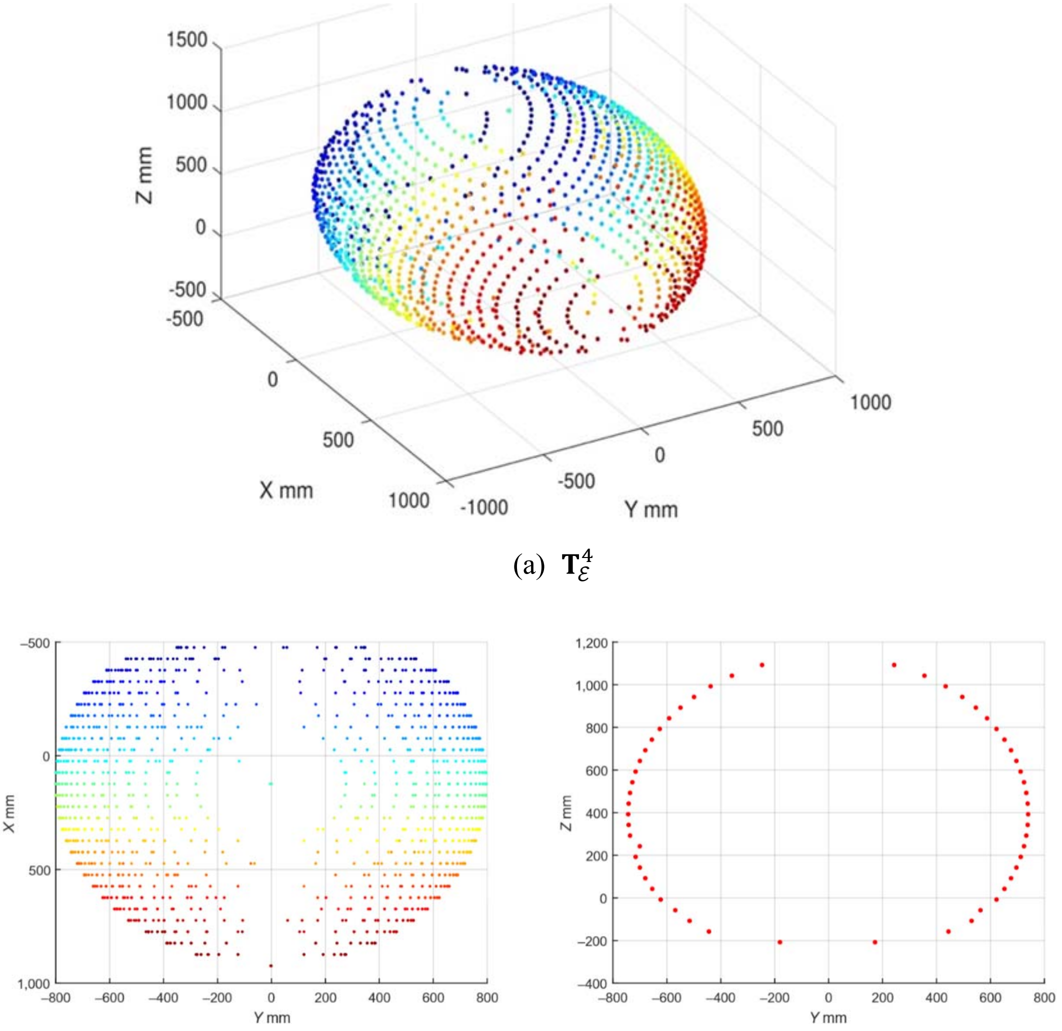
D. Runtime Reachability Check

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 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 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 positions and wait for the next update of an input value.

An example of a reachability map is shown in Figure IV-3. 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 colors and bounded by boundary points with the same color in Figure **IV-3**. 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 one hour to compute per orientation, each having 38 reachability planes per orientation.



(a) $\mathbf{T}_{\mathcal{E}}^{4}$

(b) Projection of $\mathbf{T}_{\mathcal{E}}^{4}$ onto the x-y plane

(c) $\mathbf{T}_{\mathcal{E}}^{4_{10}}$

Figure IV-3. Per-plane reachability maps with $\gamma = 0,\ i = 4$

E. User Experiment and Result

To verify the effectiveness of our H-Wall system, we conducted a user study. The experiment was divided into two parts to (1) test whether a user can distinguish between the real wall and the H-Wall and (2) evaluate the haptic feedback of our system. In both experiments, subjects wore a head-mounted display 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.

1. System Implementation

We implemented our encountered-type haptic system called H-Wall using two independent computing platforms. One is for visual rendering with the Windows 10 64bit operating system equipped with a 4.0GHz Intel Core i7 CPU, GeForce GTX 970 GPU, and 24GB RAM. The other is for haptic and robotic control with the Ubuntu 14.04 LTS 64bit operating system equipped with a 2.1GHz Intel Core i7 CPU and 8GB RAM.

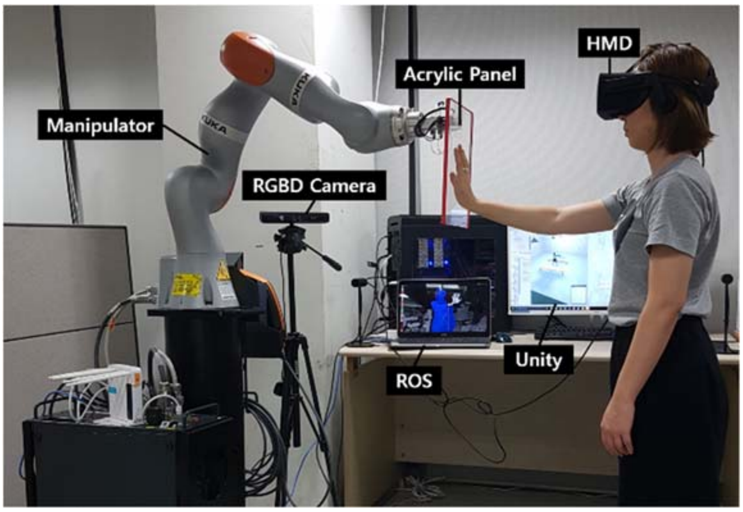


Figure IV-4. Experimental setup for the encountered-type haptic system with per-plane reachability maps

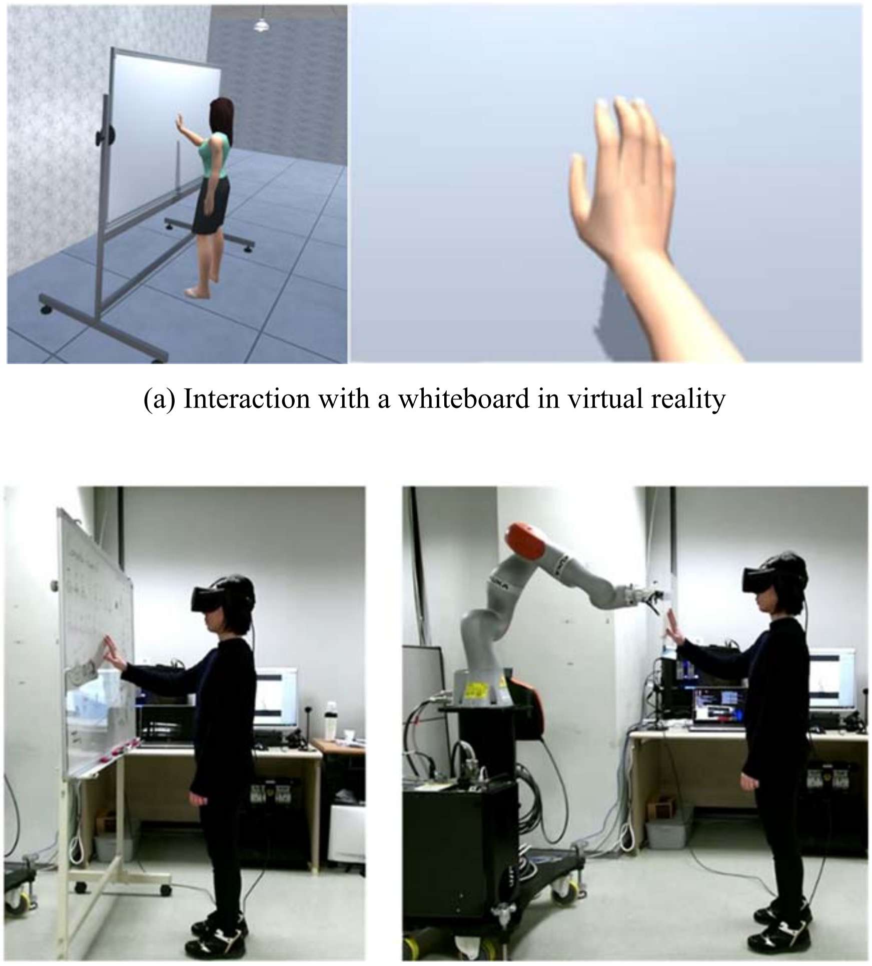
As illustrated in Figure IV-4, we used Kinect for Windows as an RGBD camera with OpenNI package to generate point cloud data and track the user's hand. KUKA LBR IIWA 7 R800 was used as a haptic manipulator, which has seven degrees of freedom and torque sensors are integrated into all seven joints. ROS indigo framework and Sunrise OS were used as a robotic software platform to perform haptic and tracking calculations and control the robot, respectively. Each of these components was programmed in C++ and Java programming languages and communicated over Ethernet via a message-passing protocol. Unity3D and Oculus Rift CV1 were used to visually render 3D virtual scenes and track the orientation of the user's head. Our rendering unit in Unity3D was programmed in C# and communicated with a ROS unit over a WiFi connection. MATLAB R2017 was used to compute the end-effector's pose using forward kinematics as well as solve nonlinear optimization with equality constraints to build per-plane reachability maps.

The reaction time of the robot manipulator is 22 ms and the latency of stereo visual rendering is 11.7 ms, which amounts to 45.45 Hz and 85.47 Hz, respectively, while the network latency is negligible. Also, the rendering bandwidth is one of the important aspects for encountered-type haptic rendering. Since the bandwidth of human movement is known to be about 10Hz [89], our haptic and visual response is fast enough to cover the human movement. 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, lightweight acrylic panel was attached to the IIWA end-effector and followed 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 from the RGBD camera.

2. Experiment 1: Distinguish Real Wall and H-Wall

The aim of this experiment was to evaluate whether the H-Wall is suitable for haptic feedback in comparison to a physical wall. We placed the virtual whiteboard in VR (Figure IV-5-(a)), a real whiteboard in the real world (Figure IV-5-(b)), and an H-Wall in real world (Figure IV-5-(c)). The subjects were asked to interact with a virtual whiteboard in a VR environment and were presented with an actual whiteboard and the H-Wall separately.



(a) Interaction with a whiteboard in virtual reality

(b) Physical whiteboard

(c) H-Wall

Figure IV-5. Experimental setup for the user study

The participants were asked to respond to the questions shown in Table IV-1.

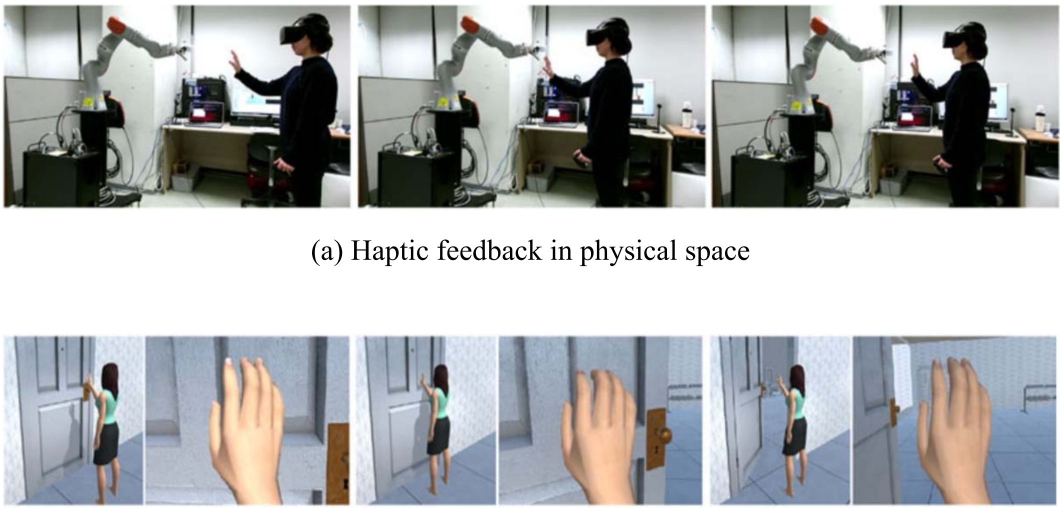
Table IV-1. Questionnaire for experiment 1: Distinguish wall

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.

In this experiment, seven of eight participants answered that it was hard to tell which wall felt more realistic. Only one subject said that the H-Wall was realistic.

3. Experiment 2: VR Interaction

The goal of Experiment 2 was to evaluate the level of immersion into VR with the H-Wall. The VR environment for this experiment was 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 with 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 **IV-6**.



(a) Haptic feedback in physical space

(b) Haptic feedback in virtual space

Figure IV-6. Active haptic interaction sequences

Then, the subjects were asked to score their subjective feelings of haptic feedback as shown in Table IV-2, using a 5-point Likert scale from 1 (worst) to 5 (best). The first question in Table IV-2 is about the naturalness of comprehensive interaction, including conventional VR avatar control using a joystick and the H-Wall haptic interaction using one's bare hand.

Table IV-2. Questionnaire for experiment 2: VR interaction

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

In this experiment, the participants gave positive feedback about the H-Wall system. The average response score to the question about the naturalness of interaction was 4.25. For the questions about the believability of static walls and dynamic doors, the scores were 4.125 and 4.375 on average, respectively.

F. Discussion

The experimental results show that most of the participants felt the H-Wall was realistic, just like a real whiteboard. Rather, only one participant answered that the H-Wall was more realistic than actual whiteboards. The participant who said that the H-Wall was realistic commented, *"In the case of an actual whiteboard, it shakes when I touch it and feels strange."* This comment implies that the H-Wall is realistic enough to replace the real object to provide providing encountered-type haptic feedback and even more suitable than the real thing as it provides well-controlled feedback.

The result of the first question in the second experiment shows that our haptic interaction can be provided together with the existing VR interaction method and does not interfere with the natural interaction. It is interesting to note that the H-Wall system provides even more realism for dynamic interaction cases than static ones, which suggests an interesting new future research direction for fully dynamic encountered-type haptic feedback.

Potential applications include an immersive VR game interface or a haptic feedback device for VR touch. As our encountered-type haptic system allows the user a dynamic virtual object interaction, it will further upgrade the sense of immersion in the horror VR game, such as Dreadhalls [90] from White Door Games, which includes opening door interaction (Figure IV-7).

Figure IV-7 Opening the door in Dreadhalls [90]

Even though the system uses precomputed robot reachability maps and checks the robot reachability in runtime, the haptic workspace is still limited by the hardware constraints. By integrating the robot arm with the mobile platform, the haptic workspace could be expanded. Since our system already checks the robot reachability in runtime, the mobile platform could easily integrate with our system by determining robot placement through reachability inversion [39].

V. Encountered-type Texture Rendering

A. Goal and Hypotheses

The main goal of this study is to synthesize the perceptual roughness through a rigid surface attached to the end-effector of an encountered-type haptic device using only the degrees of freedom that the device originally has. Extending the idea of using a collaborative robot as an encountered-type haptic device [6], we propose texture modeling and rendering methods to synthesize a variety of texture roughness.

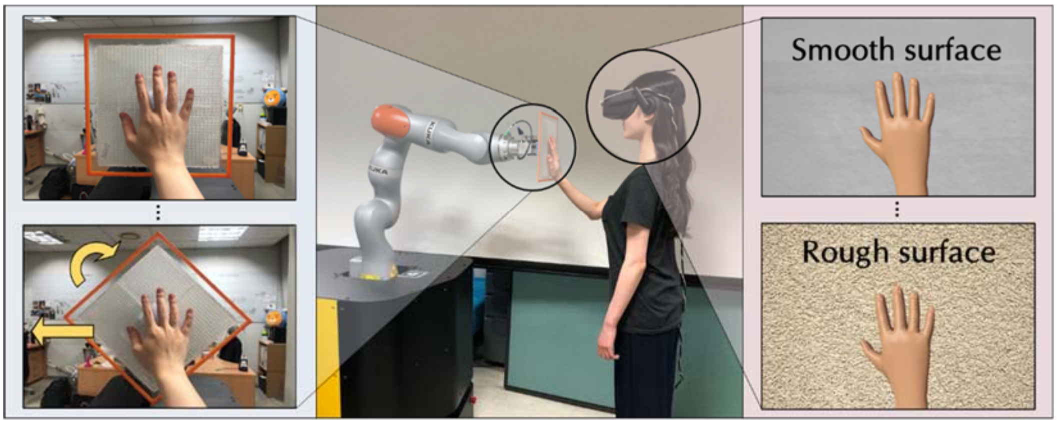


Figure V-1. Our encountered-type haptic rendering system can synthesize texture roughness using a collaborative robot manipulator (center). A rigid surface with spatially encoded bumps (top left) can give the VR user an illusion of touching surfaces with different levels of roughness from smooth (top right) to rough (bottom right) by changing the surface orientation and velocity (bottom left) [91]

By spatially encoding roughness along the radial direction on a rigid planar surface, the textured surface can represent various roughness as a change of touch orientation with respect to the surface when a user explores the surface with one’s bare hand. For instance, a user will feel different roughness whether she/he is rubbing the surface up and down or left and right.

In addition to the spatial encoding of roughness, a textured surface model tangentially moves at a certain velocity maintaining its contact with the hand to create temporally varying roughness. In other words, roughness can be rendered by modulating the orientation and velocity of the rigid surface as the user grazes over the surface model. Our encountered-type texture modeling and rendering method are based on the following hypotheses:

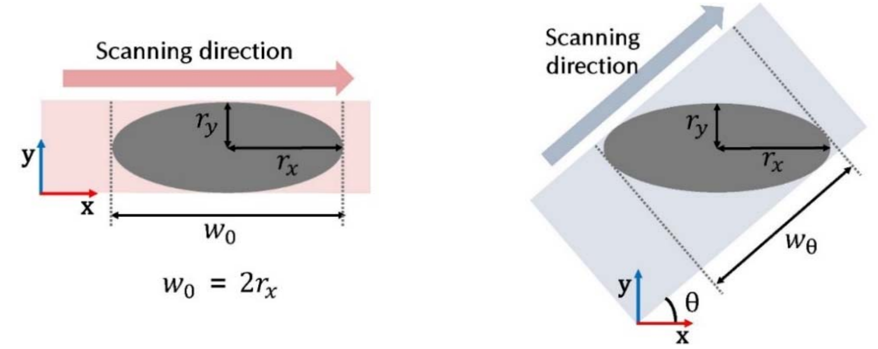
Hypothesis 1: When a user grazes over a rigid planar surface embossed with a grid of hemiellipsoidal bumps, the user can feel different levels of roughness depending on the *orientation* of the surface.

Hypothesis 2: When a user grazes over a rigid planar surface embossed with a grid of hemiellipsoidal bumps, the user can feel different levels of roughness depending on the *velocity* relative to the user's hand motion.

We validate these hypotheses and explain how to implement our haptic texturing system based on our results in Chapter V. D and Chapter V. E. Based on these hypotheses, we encode the roughness of textured surfaces spatiotemporally (Chapter V. B, Chapter V. C).

B. Textured Surface Modeling

Throughout this dissertation, we interchangeably use *dot* and *texton* to refer to the same tactile unit. Arranging an embossed dot pattern on a flat surface is a general and simple modeling method for spatially encoding tactile roughness. Similar to [92], we may use four parameters to dictate a dot pattern for representing perceived roughness: dot spacing, dot height, dot width, and dot angle. However, in our case, we select the dot width w_{θ} and the dot spacing d_{θ} as our main modeling parameters that can be modulated in runtime to generate difference roughness.



(a) Scanning direction is parallel to x-axis (i.e., $\theta = 0$)

(b) In general case when $\theta \neq 0$

Figure V-2. Perceptual width ($\boldsymbol{w_\theta}$) of an elliptic texton according to the scanning direction parameterized by $\boldsymbol{\theta}$

Meftah et al. [93] suggested that the human perception of surface roughness is a function of the spatial characteristics of the scanned surface along the scanning axis. Based on this observation, we define the texton width along a scanning direction, affecting roughness perception. When a circular texton such as a truncated cone or hemisphere is used for a dot pattern, the user will experience a constant perceptual width regardless of the scanning direction. In our case, however, we select an elliptically-shaped texton to control the texton width, as illustrated in Figure V-2. To define the dot width along the scanning direction w_θ, we consider the length of the texton projected onto the scanning direction. Then, w_θ is a function of the angle of scanning direction $\theta \in \left[0, \frac{\pi}{2}\right]$ as follows:

$$\boldsymbol{w_\theta = 2\sqrt{r_x^2 \, cos^2 \theta + r_y^2 \, sin^2 \theta}} \ , \qquad (\text{V.1})$$

where r_x and r_y are the x-radius and y-radius of an elliptical texton with $r_x > r_y > 0$, without loss of generality.

It is well known in the psychophysics field that the psychometric relationship between texton spacing and perceived roughness forms an inverted U-shape graph [64]. Since the perceived roughness increases with reduced texton spacing along the scanning axis when the inter-distance between textons is greater than 3.0 mm, we set the shorter axial radius of an elliptical texton to 1.5 mm so that the distance between textons is at least 3.0 mm.

In order to increase the effectiveness of the perceived roughness as θ increases, textons should be arranged in such a geometric way that their inter-distance (or spacing) d_θ decreases as θ increases. As illustrated in Figure V-3, we arrange the textons in a rectangular grid so that the encoded roughness increases as θ increases. Here, the distance d_θ is defined as a *ray distance* that is measured from the center of an ellipse to the nearest ellipse along the scanning direction with an angle of $\theta \geq \theta_{min}$. We designed the texton as hemiellipsoid since the textured surface model with hemiellipsoids provides a wider range of roughness compared to the model with elliptical truncated cones. We verified this in our additional user experiment, and the result will be discussed in Chapter V. F.

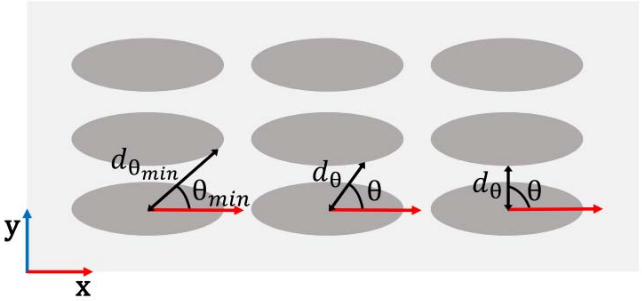


Figure V-3. Variation of the distance between the top of tactile elements ($\boldsymbol{d_\theta}$) according to a scanning direction parameterized by $\boldsymbol{\theta}$

3D modeling software, such as 3DS MAX, was used for this spatial encoding of textons, and we printed this model using a Form2 3D printer with 50-micrometer resolution. We will perform a user study and verify our spatial encoding scheme providing different roughness relative to the scanning direction in Chapter V. D.

C. Haptic Texture Rendering

Haptic texturing is rendered through our encountered-type haptic display. A textured surface, as designed in Chapter V. B, is attached to the end-effector of the haptic device and interacts with the user's bare hand. In runtime, the haptic system tracks the user's hand motion and controls the orientation and the velocity of the manipulator's end-effector depending on the surface roughness of the virtual object that the user tries to touch.

In order to haptically render the roughness of virtual objects, we encoded their roughness and mapped it to virtual roughness in our system. Specifically, for n objects' virtual textures that will interact with VR users, their roughness are first sorted in increasing order, and n distinguishable roughness are selected in increasing order from the m encoded roughness (assuming that $m \geq n$) and mapped to the virtual textures. Here, the exact roughness mapping from virtual to encoded textures is not crucial as long as the mapping is one-to-one since our objective here is to make the user feel different roughness of textures.

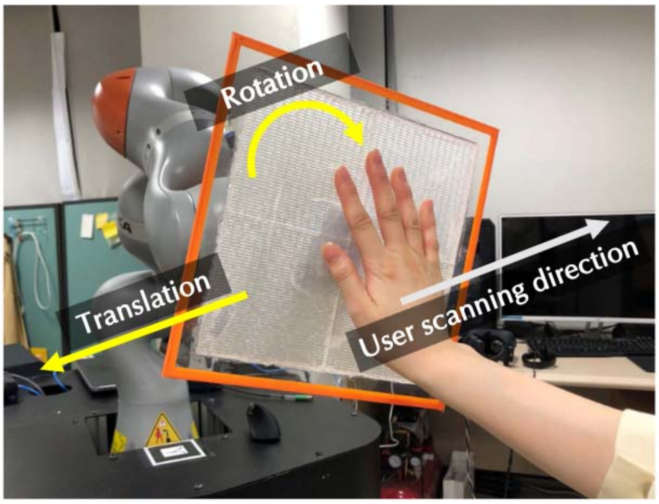


Figure V-4. Tactile rendering by rotating and translating the surface model relative to the user’s scanning direction.

The encoded roughness is associated with the orientation and velocity of the end-effector relative to the user's hand scanning direction in the offline process. At runtime, the orientation and velocity value corresponding to the virtual object's texture that the user intends to touch is chosen by our system and rendered to the user via the manipulator to deliver haptic feedback. As shown in Figure V-4, the end-effector rotates and translates a textured surface model to provide a proper relative velocity between the user's hand and the surface, representing the roughness of a virtual object that the user is touching in VR.

D. Preliminary Study and Result

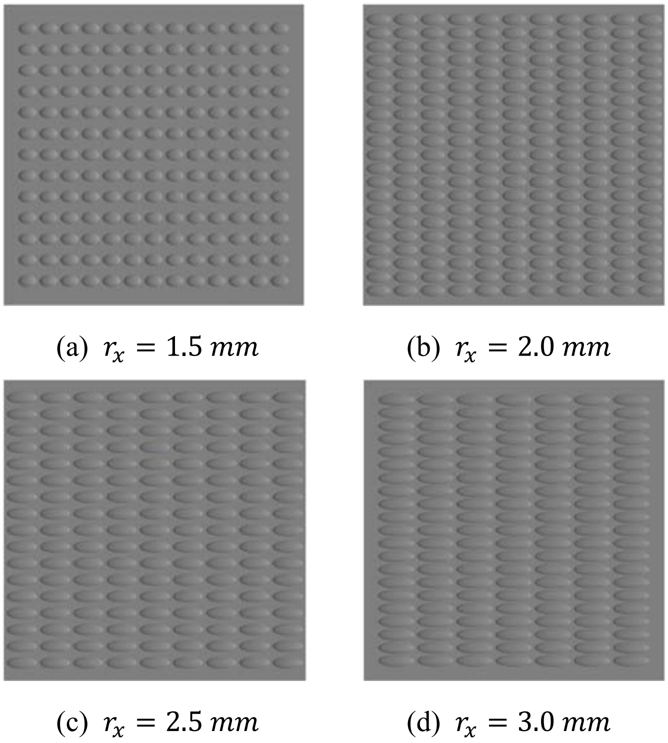
Prior to testing our hypotheses that our surface textured model with various scanning direction and velocities can provide a wide range of perceived roughness, we conducted a preliminary user study to (1) examine if altering users' roughness perception by changing the scanning direction of our model is feasible and (2) decide on the design of surface texture model, specifically the radius in x-axis of each bump, for simulating a range of perceived roughness as wide as possible.

1. Participants

For this study, we recruited seven female volunteers from a women's university that the paper authors are affiliated with. All the participants were right-handed, and their average age was 26.1 ranging from 24 to 29.

2. Experimental Conditions

To provide as wide a range of perceived roughness as possible with the varying scanning directions, we found the most effective radius of hemiellipsoidal bumps on a surface. We investigated four different radii in x-axis (denoted as r_x) starting from 1.5 mm to 3.0 mm with the interval of 0.5 mm while the radius in y-axis was fixed to 1.0 mm, as shown in Figure V-5. The tested angles for the scanning directions were chosen in terms of the linearly-increasing slope of scanning direction, as shown in Figure V-6, which were: $0(\arctan 0)$, $45(\arctan 1)$, $63.43(\arctan 2)$, $71.57(\arctan 3)$, and $90(\arctan \infty)$ degrees.



(a) $r_x = 1.5\,mm$

(b) $r_x = 2.0\,mm$

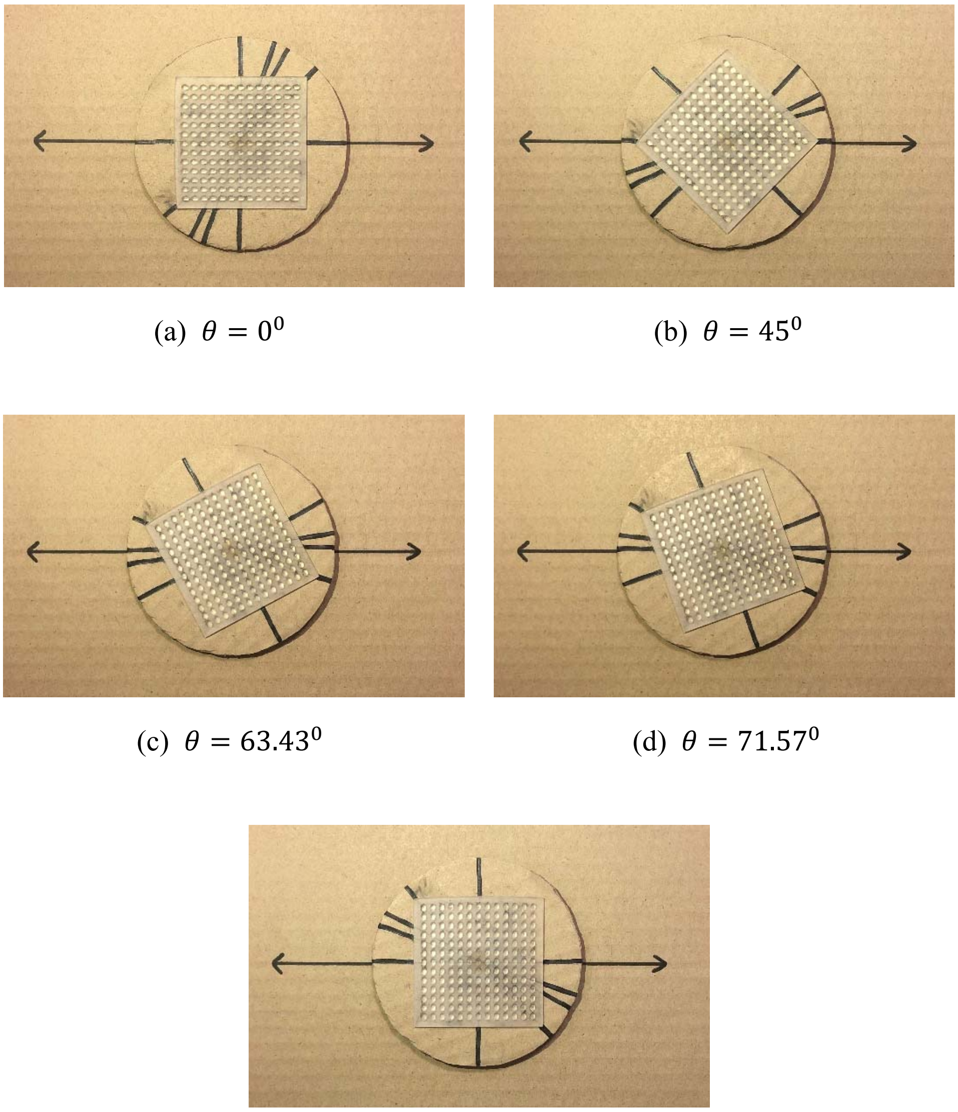
(c) $r_x = 2.5\,mm$

(d) $r_x = 3.0\,mm$

Figure V-5. Four different roughness encoded models with various radii on the x-axis

3. Apparatus

For each of the four roughness encoded models, we prepared a rigid polymer surface patch (five $cm \times$ five cm) using a 3D printer. Each patch was firmly attached to round cardboard with tacks. To test different scanning directions, instead of asking subjects to change their scanning directions, an experimenter rotated each patch to change its orientation according to the five predefined angles that were marked on the cardboard. In addition, we had an opaque cloth for covering the patch area during the experiment to help subjects focus on their sense of touch while preventing them from seeing the surface texture details.



(a) $\theta = 0^0$

(b) $\theta = 45^0$

(c) $\theta = 63.43^0$

(d) $\theta = 71.57^0$

(e) $\theta = 90^0$

Figure V-6. Five different scanning angles used for the preliminary study

4. Procedure

After a brief introduction about the study procedure, we asked subjects to use their index finger of the dominant hand to rub a patch with a specific scanning direction under five different orientations, while each patch was covered by an opaque cloth, as shown in Figure V-7. We followed the ratio scaling method of magnitude estimation, which allows subjects to select their range of numbers and make a numerical estimation of perceived magnitudes [94].

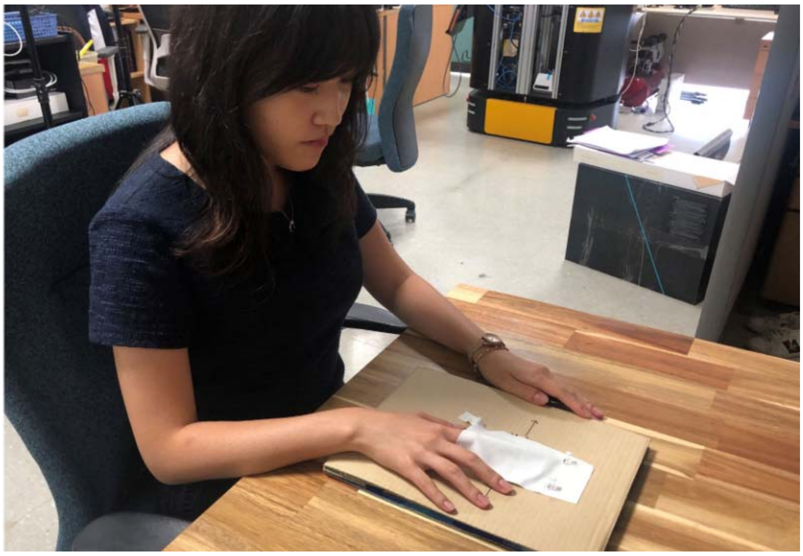


Figure V-7. Experimental setup for the preliminary study

Subjects were asked to rate the perceived roughness with any positive number, with a higher value indicating a rougher texture. Subjects were allowed to choose their own scale when reporting their subjective roughness. The presentation order was first randomized by patches with different radii, then by scanning directions for each round, and the test was repeated three times. In total, each subject performed 60 trials (four radii in x-axis × five scanning orientations × three rounds).

5. Data Analysis

Since subjects were free to choose any numerical value for reporting the magnitude of perceived roughness, the captured raw data needed to be normalized for relative comparisons. Thus, following the magnitude estimation method [94], [95], perceived roughness for each patch and orientation across three trials per subject was averaged and normalized by dividing each subject's responses by the grand mean of all subjects' averaged data. The normalized averaged data were then re-scaled by being multiplied by the grand mean of all subjects' averaged data.

6. Results

As shown in Figure V-8, our results suggest that it is feasible to change the perceived level of roughness by varying the scanning direction using our model. Moreover, it seems that the perceived level of roughness tends to increase as the scanning angle increases from 0 to 90 degrees, except when r_x is set to 1.5 mm. Although further investigation is needed, it could be that the impact of increasing the scanning angle is relatively small, and the difference between the radii of x and y is smaller than a certain threshold. Based on the results, we chose a textured surface model with $r_x = 3.0\,mm$ for our encountered-type haptic system, which showed the widest range of average perceived roughness to maximize the effect.

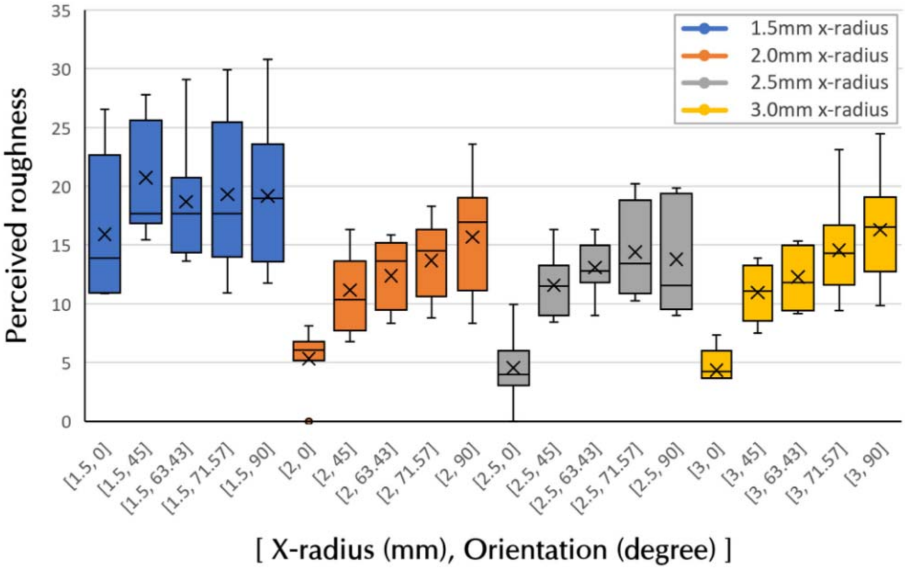


Figure V-8. The normalized roughness estimates varying in a surface patch with different x-radius (r_x) and scanning direction

E. Main Study and Result

To test our hypotheses that a surface textured model with various grazing orientations and velocity can provide a wide range of perceived roughness, we designed and conducted a within-subject study where participants were asked to report perceived roughness after scanning using our encountered-type haptic texturing system.

1. System Implementation

The implementation of the haptic texture rendering system followed the implementation detail described in Chapter IV. E. 1, but the tracking space was expended by mounting an infrared (IR) motion sensor on the user's HMD. Also, we replaced the acrylic panel with the textured surface model to synthesize multiple surface roughness and serve as a proxy of the virtual object.

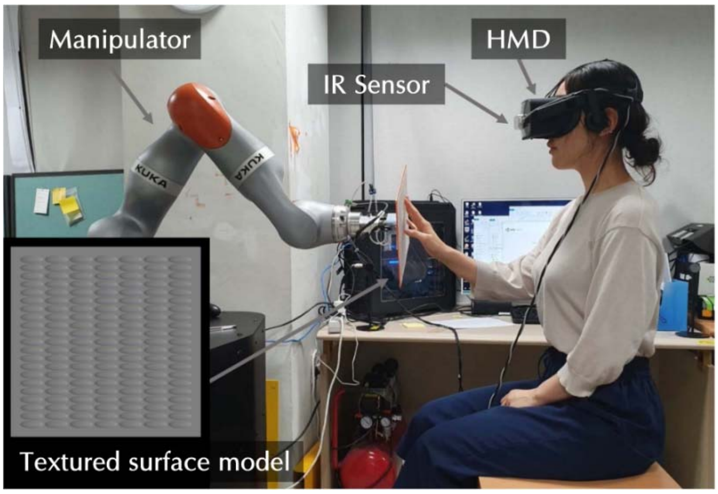


Figure V-9. Experimental setup for the main user study

The response time of the robot manipulator was measured at 22 ms including 0.125 ms controller PC latency, and the latency of stereo visual rendering was 11.7 ms. We used Leap motion as a motion sensor to track the user's hand, and its frame latency was 8.33 ms, while the network latency was negligible. As a haptic proxy for a virtual object, a textured surface model is attached to the IIWA's end-effector and follows the human hand in highly real-time providing a sense of illusion of touching a virtual object.

2. Participants

Twenty-five human subjects, consisting of 20 females and 5 males, participated in our main study. The ages of the subjects varied between 19 and 50, with an average age of 26.6. All subjects were right-handed, and four of them had participated in our preliminary study.

3. Experimental Conditions

The orientation conditions were the same as in our preliminary study: 0, 45, 63.43, 71.57, and 90 degrees. The velocity conditions included 0, 20, 40, 60, 80 (mm/sec), which were distinguishable velocities observed from an internal study. The x-radius of the hemiellipsoidal bump (r_x) was set to 3.0 mm based on findings from the preliminary study.

4. Apparatus

We prepared a rigid polymer surface patch (20 cm × 20 cm) with $r_x = 3.0\,mm$ using a 3D printer. As the printable size of patches is limited by hardware constraints, we divided the model into four sub-patches of a squared shape (10 cm × 10 cm) and printed them separately. Then we attached the surface models to a rigid acrylic panel that was fastened to the end-effector of the manipulator. Also, we printed a custom robotic gripper model that can be connected to the end-effector by bolts to firmly attach the acrylic panel to the robot.

We used HMD and IR sensors for tracking the user's head and hand motions and preventing the subject from seeing the surface detail. In order to focus on the haptic effect on roughness perception, subjects were asked to wear an HMD during the experiment, as shown in Figure V-9, and no visual images were provided through the HMD. The sequence of experimental conditions was programmatically set per subject in advance so that the experimenter could prepare for the next trial remotely using an Oculus controller.

5. Procedure

Before the experiment began, the subjects were informed that their task was to rate the roughness of each series of system conditions without considering other texture attributes, such as warm/cold, soft/hard, or flat/bumpy. The magnitude estimation method in Chapter V. E. 7 was used to numerically represent estimation data for the subject's perceived roughness.

The subjects were comfortably seated on a chair facing the haptic manipulator and wore an HMD that was displaying nothing. An IR motion sensor was mounted on top of the HMD to track users' hand movement, as described in Chapter V. Then, subjects were asked to raise their dominant hand to their shoulder height and extend the hand forward to touch a textured surface model. When they felt contact with the model, they were instructed to touch the surface from side to side, keeping both the contact force and the scanning speed constant during the entire experiment. Since keeping these parameters constant depends on the user's capabilities, the subjects repeated the exercises until they could experiment correctly. The subjects evaluated surface texture roughness using only a positive number, with a higher number indicating a rougher texture. The textured surface was cleaned with alcohol after each subject completed their experiment.

6. Data Analysis

By combining five different orientations and five different velocity values for the surface texture, 25 different levels of conditions were tested for each subject. In total, each subject performed 150 trials (five orientations × five velocity levels × six rounds). Each subject spent 36.3 minutes for the entire experiment procedure on average. The perceived roughness reported by each subject was normalized across rounds, as in our preliminary study.

So as not to violate the sphericity before conducting analysis, we apply Aligned Rank Transform (ART) [96] to our non-parametric data. Then we conduct a two-way repeated-measures ANOVA with factors of orientation (5-level) and velocity (5-level). Pairwise t-tests were performed for post hoc analysis with Bonferroni adjustments.

7. Results

Figure V-10 shows the results of the roughness estimation experiment using our texture synthesizing system. The x-axis in the graph represents the combinations of orientation and velocity values, and the y-axis represents the rescaled normalized roughness. The curves in the graph are shown in different colors to distinguish the different conditions on orientation.

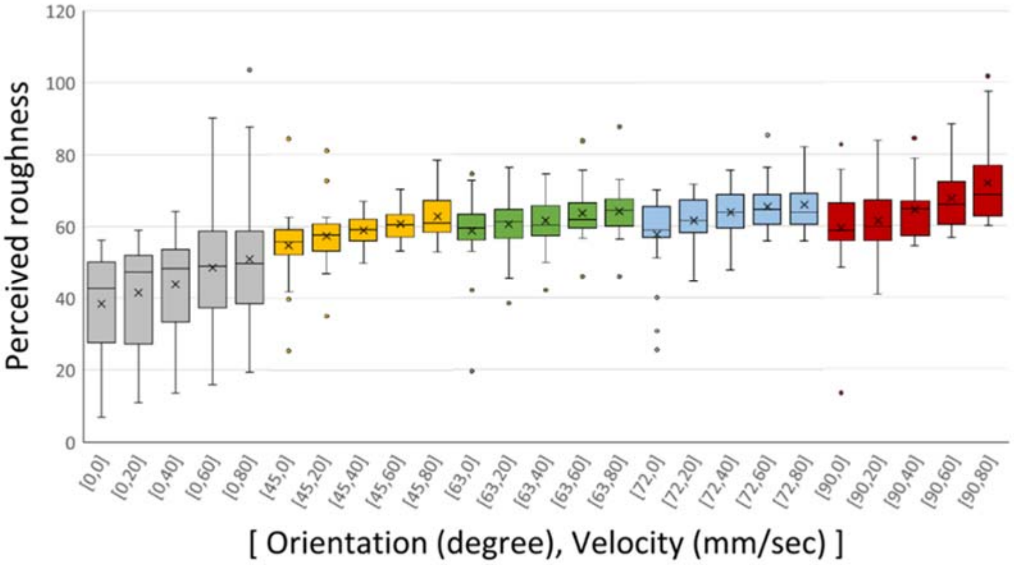


Figure V-10. The normalized magnitude estimation of roughness with respect to the orientation and velocity of an end-effector

Overall, subjects perceived the surface to be rougher as orientation angle and velocity increased. Moreover, changes in orientation seemed to have a greater influence on perceived roughness compared to changes in velocity. To investigate the impact of orientation and velocity on perceived roughness and their interaction effect, we performed a two-way analysis of variance (ANOVA) with repeated measures. As a result, we found a significant main effect and large effect size of orientation ($F_{(4)} = 71.67,\ p < 0.001,$

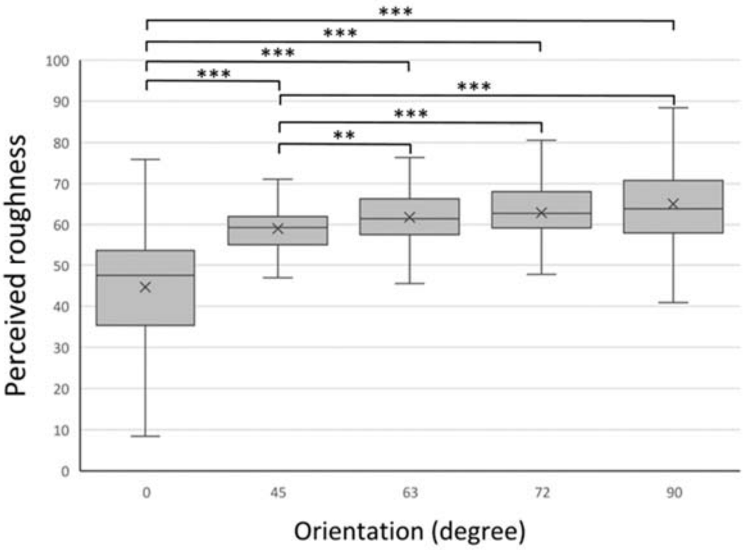
$\eta_p^2 = 0.3323$) and significant main effect and medium effect size of velocity ($F_{(4)} = 15.87$, $p < 0.001$, $\eta_p^2 = 0.0993$) on roughness estimation confirming both of our hypotheses (Table V-1).

Table V-1. Result of the two-way ANOVA with repeated measures

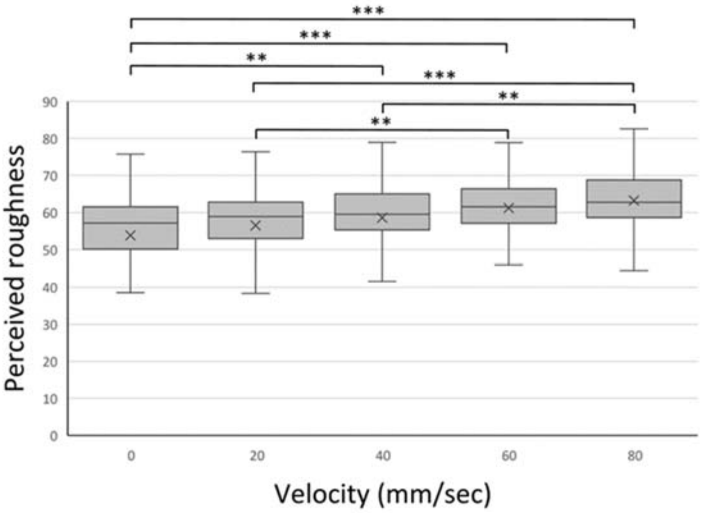
Factor	DoF	F-value	p-value
Orientation	4	71.66748	≈ 0
Velocity	4	15.87291	≈ 0
Orientation × Velocity	16	0.43698	0.9726

Figure V-11-(a) shows the post hoc analysis with Bonferroni adjustments. Stars indicate p values from pairwise post hoc analysis results: '*' for p < 0.05, '**' for p < 0.01, and '***' for p < 0.001. The results show that the perceived roughness grows significantly as the orientation angle increases, in general. For example, the perceived roughness of the scanning orientation angle of 0 and 45 is significantly less than other angles (p ≤0.001 for all). However, there was no statistically significant difference in perceived roughness at angles of 63.43, 71.57, and 90 degrees.

Similarly, subjects tend to perceive our surface model to be rougher as the velocity increases, as shown in Figure V-11-(b). While no significant differences in perceived roughness between every pair with 20 mm/sec differences (*e.g.*, 20 and 40 mm/sec and 60 and 80 mm/sec) were found, the perceived roughness differences between all pairs with at least 40 mm/sec difference were statistically significant. This suggests that our surface model can be used to simulate various levels of perceived roughness by tuning either of the parameters of orientation and velocity. Meanwhile, no interaction effect between orientation and velocity was found to be significant.



(a) Perceived roughness with respect to the end-effector orientation



(b) Perceived roughness with respect to the end-effector orientation

Figure V-11. Boxplots showing users' perceived roughness on average when varying orientation and velocity

F. Discussion

In this chapter, we discuss how our results can contribute to the implementation of roughness synthesis and suggest potential applications for virtual environments.

1. Supporting Roughness Changes with One Surface

Our results demonstrate that our encountered-type textured surface model with hemiellipsoidal bumps can be used for changing users' perception of roughness levels simply by changing orientation and velocity without using body-mounted or hand-held devices. Moreover, compared to other encountered-type haptic systems with an additional tactile display, our simple method can represent various levels of perceived roughness using one rigid surface model. Therefore, we can dynamically change the level of perceived roughness in real-time without switching to a different textured surface each time. Our method is expected to reduce the system complexity and the cost of an encountered-type haptic system that provides textural information.

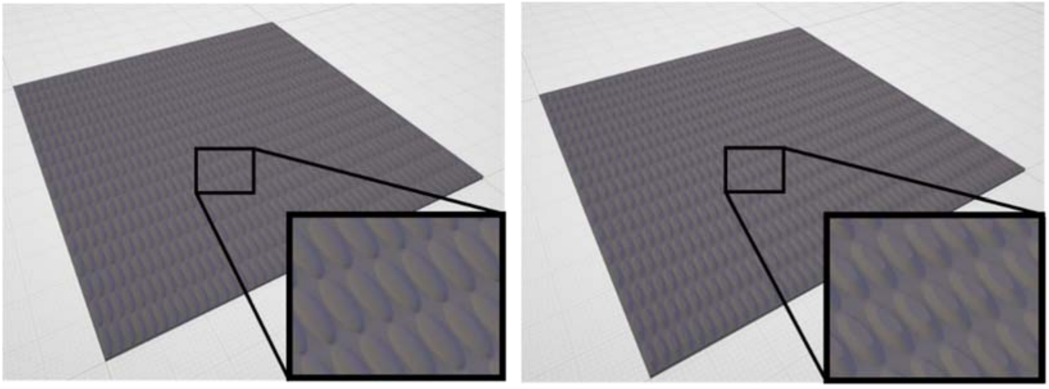


Figure V-12. Textured surface models with hemiellipsoids (left)
and truncated cones (right)

To vary the range of perceived roughness, a hemiellipsoid is used as an elliptical texton in our texture model. We also conducted user experiments for cross verification using the truncated-cone model [62] and our hemiellipsoidal model, and verified that our model more clearly shows the increase in roughness with respect to the increase in angle, providing a wider range of roughness (Figure V-13). In this experiment, r_x, r_y, and the height of elliptical truncated cones are constant and set to the same values as our hemiellipsoid model and the dot angle of truncated cones is set to 45 degrees. These experiments follow the same procedure as the main user study with eight subjects.

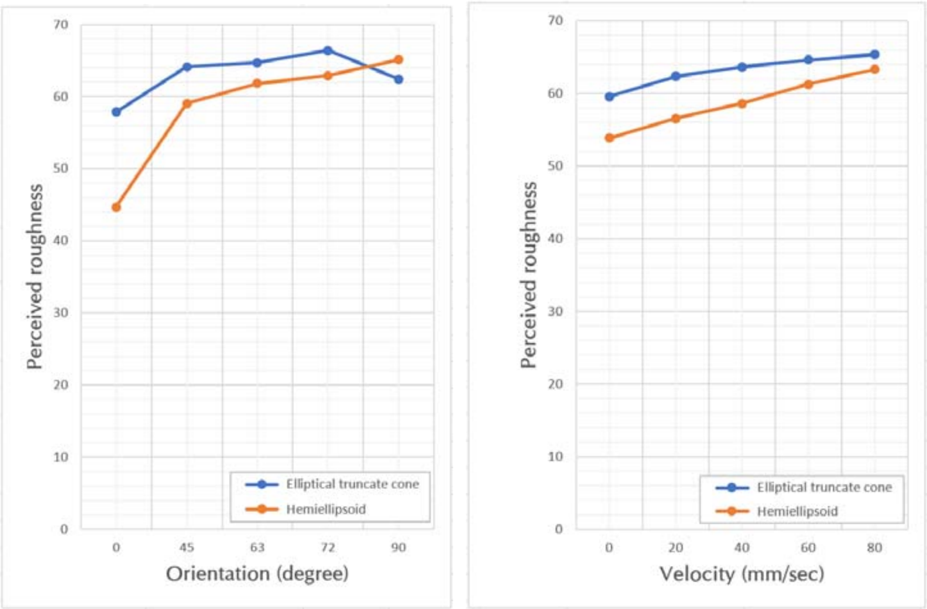


Figure V-13. The perceived roughness according to the orientation change (left) and velocity change (right). The blue graph represents the model with elliptical truncated cones and the orange graph represents our model with hemiellipsoids

The increasing effect of roughness proportional to the increase in angle is more evident in our hemiellipsoid model than in the model with truncated cones (Figure V-13). In particular, when a user scans the surface of the model with truncated cones along its sharp edge, the proportional relationship between the orientation and perceived roughness may not be maintained.

2. Simulating Various Levels of Roughness

Our findings confirm that roughness perception is affected by temporal cues that depend on scanning velocity, similarly to prior work [61]. As our textured model has a large inter-distance between textons, experimental conditions with a larger velocity resulted in higher roughness, which is consistent with the result of Connor and Johnson [62]. Whereas the earlier work was focused on a passive touch experiment for studying the effect of temporal cues under a precisely controlled condition, our result shows that the temporal cues also have a significant influence on roughness perception during active touch. In addition, we show that spatial cues offered by our encountered-type texture surface model with different orientations can also be used to alter users' perception of texture roughness. Most of all, we demonstrate that our model can simulate a particular level of perceived roughness by tuning two parameters: orientation angle and velocity. For instance, gradual changes in perceived roughness can be implemented by changing the velocity, which is almost linear in terms of roughness level, as shown in Figure V-13. Thus, once we identify the range I of roughness change a priori, we can continuously map a certain level of roughness to an orientation-velocity pair (θ, v) in two steps: first we find an orientation θ_I that contains the roughness range I, then change the velocity value v continuously to cover I.

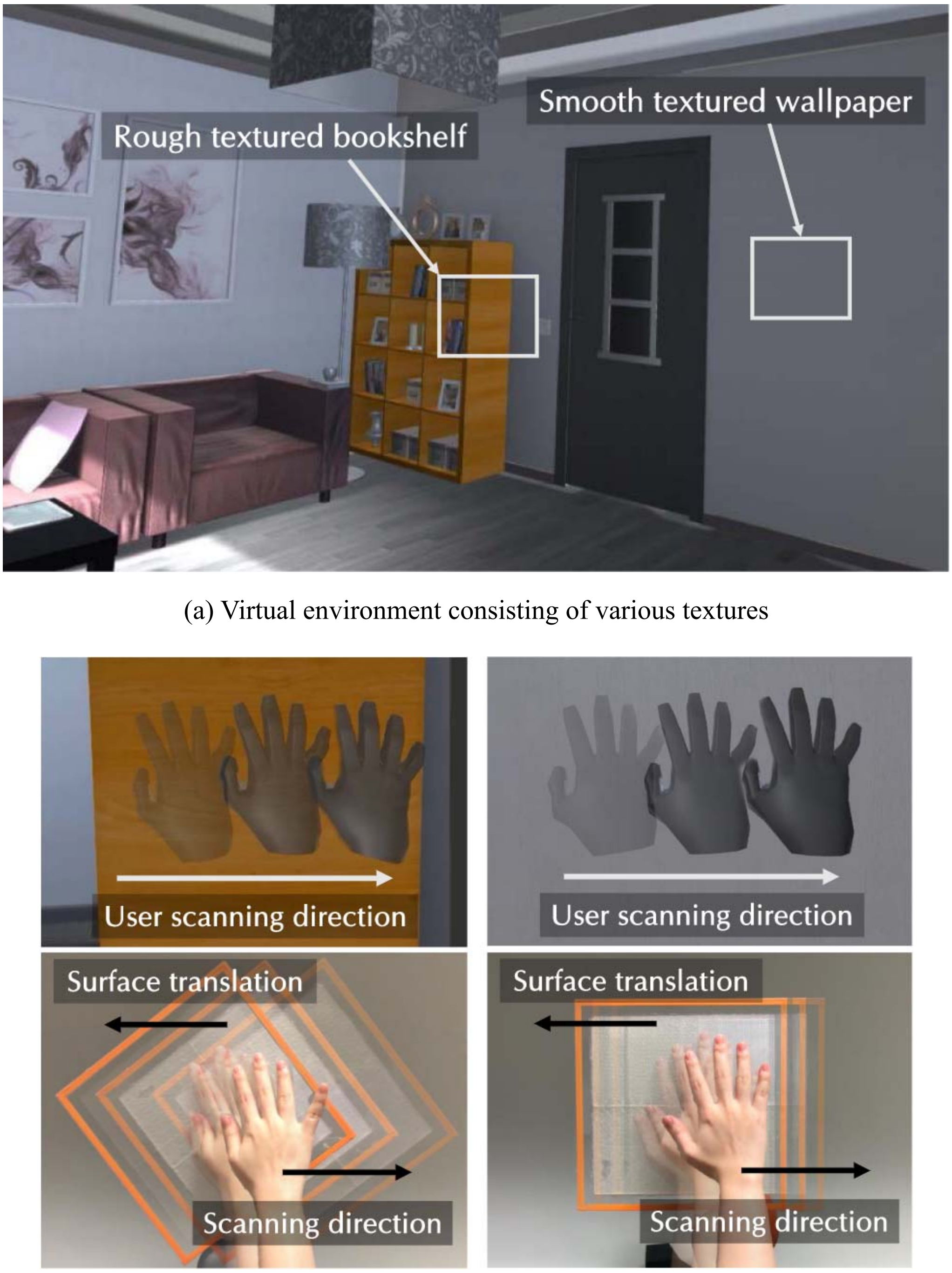
3. Active and Passive Touch

Our texture roughness rendering method provides both active and passive touch simultaneously. The encoded spatial cue of roughness is represented by the user's active touch. As the haptic manipulator holds its orientation relative to the user's hand motion, there is no relative rotational motion between the user's hand and the textured surface model. On the other hand, both passive and active touch is involved in delivering temporal cues. While the user actively scans the surface of the model, a haptic manipulator translates the model with a predefined velocity determined by the roughness of the virtual object. This means that the textured surface moves in the opposite direction to the user's scanning direction, regardless of the user's scanning speed.

While previous studies demonstrated the little effect of relative speed [66], [67], our findings show the significant effect of scanning speed on perceived roughness. Based on the human brain research [97] that shows roughness perception differs depending on the scanning velocity during a passive touch, we conjecture that the simultaneous active and passive touch makes the experimental result differ from previous research. When the haptic system provides passive feedback during the user's active touch in VR, the system might deceive the human perception process and substantiate the effect of scanning speed. To identify the exact cause and neural process, additional in-depth research in the sensory and neural recognition field is needed, which we leave as future work.

4. Potential VR Applications

While deepening our understanding of how we can manipulate roughness perception with a spatiotemporally encoded surface model, our findings suggest implications for potential VR applications with bare hand interaction in particular. An encountered-type haptic system with a limited number of textures can use our nine levels of distinctive roughness. If the order of texture roughness is determined in the target application, one can use our spatiotemporal encoding of textures to cover the variation of roughness. For example, the player of an immersive VR game can feel the distinctive roughness corresponding to the texture of the game object using the encountered-type haptic system. Figure V-14 shows our VR prototype implementation of the indoor environment consisting of furniture, such as a door and bookshelf. As the user grazes at the surfaces of the furniture with different textures in the virtual environment, she or he can feel different levels of roughness by scanning the textured surface using the encountered-type haptic display.



(a) Virtual environment consisting of various textures

(b) Rough texture

(c) Smooth texture

Figure V-14. Prototype of VR application using our roughness synthesizing method

Although we limited the user's movement to a straight line in our user study, our prototype implementation can provide the user with a non-linear movement. Further study on verifying the significance of roughness rendering supporting non-linear scanning could contribute to improving VR realism. This system will be even more powerful if the manipulator is mobile, as this will extend the limited haptic workspace. This is our base implementation platform, although our current haptic system does not exploit mobility.

5. Limitations and Future Work

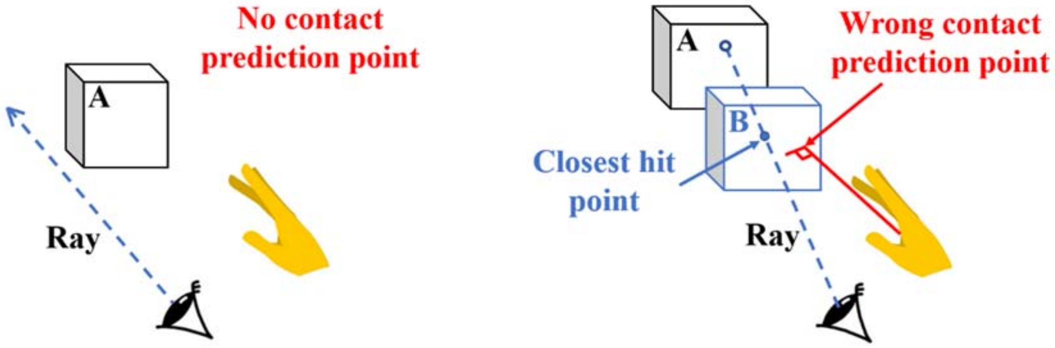
As a single-session controlled lab study, limitations exist. First of all, since we imposed a motion constraint to control the contact force and velocity of hand motion, the subjects were not completely free in their hand motion during the experiment. However, users should be able to freely change the contact force and the speed of rubbing in virtual environments with tactile feedback. Thus, the uncertainties caused by the active movement of the user should be considered when applying our method to an actual VR application. Since the majority of our subjects were females in their twenties, another limitation is that our results on the ranges of perceived roughness may have differed if we had recruited a different group of participants. Although a previous work [98] shows that there is no difference in tactile spatial acuity by gender, it should be noted that there could be a gender or age difference that we have not considered.

Moreover, we did not consider synthesizing the surface of an actual object since the goal of the main study was to maximize the range of roughness synthesis with a single encoded model. Future research should consider an analytical model that can be mapped to a specific object to simulate the same texture but with a relative difference in terms of roughness. Also, other modeling parameters, such as texton height, could be studied to provide a more realistic texture.

Although we focused on macro-scale roughness perception, a further study for synthesizing fine roughness would be interesting. In order to do this, a micro-scale modeling method and finer velocity control during texture rendering will be needed. Also, beyond the distinctive level of roughness, a more serious study on a continuous level of roughness synthesis is deserved to represent the diverse range of visual or physical textures. In this case, the real-time system reaction is much more critical. Although no subject complained about the abnormality of system latency during the user experiment, a reduction in reaction time both in haptic rendering and visual rendering is still desirable.

VI. Contact Prediction for Encountered-type Haptics Using Learning from Demonstrations

The contact prediction method is described in Chapter III. C utilizes the region of interest concept, which is simple and intuitive. This approach successfully predicts the contact configuration in a simple VR environment but is too sensitive to the user's gaze direction, so contact prediction becomes intractable in a complex VR environment. For example, since this approach cast the ray from the user's viewpoint, when the object that the user tried to touch is missed by the ray, the contact prediction module fails to predict the contract configuration (Figure VI-1-(a)). Also, when objects are occluded from the user's field of view (Figure VI-1-(b)), the system fails to predict the contact point and cannot generate an adequate haptic response.



(a) Virtual object is not hit by ray (b) Virtual object is occluded by other objects

Figure VI-1. Examples of contact prediction failure

Instead of calculating the contact prediction point deterministically and thereby placing the encountered-type haptic probe at the predicted configuration, we propose an alternative method to directly generate a haptic manipulator response by inferring the manipulator trajectory from the observation of human hand gesture and circumvent the issues with the contact prediction. To this end, we encode the prior probability distribution of interaction between a human and a robot by learning from demonstrations, and at runtime, based on the observation of human motion, generate the manipulator response by Bayesian filtering using BIP [14].

A. Goal and Method

The main goal of this study is to generate an end-to-end robot manipulator response for encountered-type haptic feedback based on the observation of the hand gesture when the user is about to touch a virtual object. Conventionally, the encountered-type haptic response is generated deterministically in consideration of the real world where the user and haptic manipulators exist and the virtual world where virtual objects exist. In detail, after the user's tracking information is imported into the VR environment, the system finds the point where the contact is most likely to occur considering the interaction between the user and the virtual object. Then the system maps the contact predicted configuration back to the real world to place the haptic display in the corresponding configuration. Considering the interaction between the haptic manipulator and human hand, without VR environmental information, the encountered-type haptic feedback could be provided by learning from demonstrations. To this end, we simulate encountered-type haptic interactions in VR to obtain desired demonstration data and conduct learning from demonstrations using BIP [14].

The BIP infers the trajectory of the controlled agent by imitating the interaction demonstrations of multiple agents utilizing IP [78]. More specifically, BIP learns probability distributions over joint actions between interaction agents and infers the controlled agent action as a function of observed agent behavior. Following the previous methods [14], [80], we produce the HRI demonstration data of encountered-type haptic interaction and use a haptic manipulator as a controlled agent and the human hand as an observed agent. The demonstrations are used to train the model of dynamics of interaction represented within a basis function space, encoded as a prior distribution of agents' joint interaction. In runtime, the system performs the Bayesian filtering to human observation data. In this way, the trained model infers the next human movement and generates an encountered-type haptic robot response.

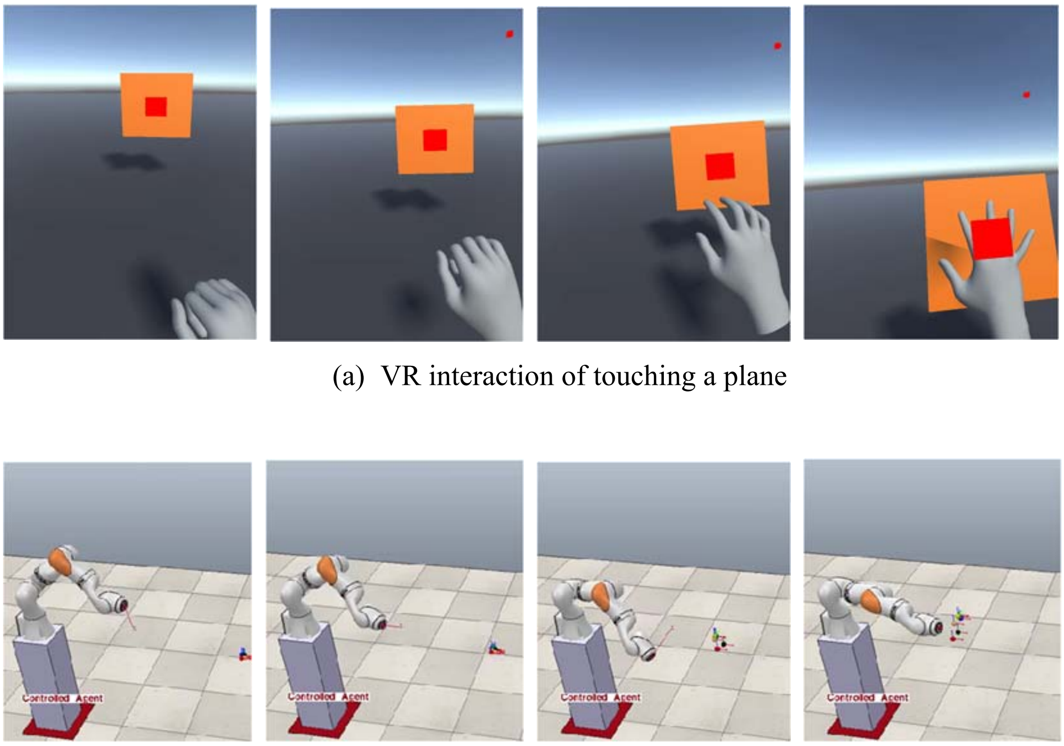
For the training BIP model, we generate demonstrations using VR and simulation world, represent the training data within a basis function space by decomposition, and then calculate the joint distribution. For inference and response generation, the system conducts Bayesian filtering to the observed human movement in runtime. Through Bayesian filtering, the system predicts the next human movement, determines the current condition with a learned model, and then generates a robot response. The method is described in the following chapters (Chapter VI. B).

B. End-to-end Learning for Encountered-type Haptic Rendering

1. Generating Training Data

To learn the contact prediction model from demonstrations, the encountered-haptic demonstration data, including robot trajectory and human motion trajectory, should be captured. However, it is hard to generate haptic interaction data especially considering the three independent systems: robotic haptic manipulator, VR environment, and real-world tracking system. Also, synchronizing temporally and spatially separated data is a challenging problem. To handle these issues, we propose a method for generating training data using a robot simulation environment. We track the user, who is interacting with VR objects, and the user's movement is captured by the real-world tracking system. The system then performs coordinate transformation from tracking system to simulation environment coordinate system. The trajectory of the encountered-haptic manipulator corresponding to the user motion is generated by the simulator, while the user tracking data is represented in the simulation environment in real-time. By capturing the user tracking data and robot trajectory simultaneously, our method synchronizes the separate human-robot interaction data spatiotemporally.

To generate encountered-type haptic interaction data, we randomly sample the point in per-plane reachability maps and use this point as a target contact point during the interaction. As shown in Figure VI-2-(a), the small size plane with target configuration in a VR environment is rendered via HMD. In the simulation environment, the haptic manipulator moves to the target configuration by calculating pseudo inverse kinematics. As shown in Figure VI-2-(b), the hand tracking data is shared in the simulation environment, and the robot joint angle and the human hand position are recorded simultaneously.



(a) VR interaction of touching a plane

(b) Encountered-type haptic simulation in robot simulator

Figure VI-2. Generating demonstration of the encountered-type haptic interaction using (a) user's VR interaction and (b) robot simulation

From the internal study, we confirm that the user's hand motion for touching a flat object usually starts with a clenched hand and switches to an open hand when touching the object's surface. Based on these findings, we generate training data that reflects the hand movement of the user's touch motion.

2. Bayesian Interaction Primitives

To model the human-robot interaction by extracting the interaction dynamics based on the observation of the demonstration, we utilize IntPrim framework [14]. Using BIP, the interactions of two agents, including human and haptic manipulator, are transformed into the latent space by basis function decomposition, and a joint model of both space and time variables is learned and then encoded within the same probabilistic representation. In runtime, the trained model is used to generate the trajectory of the haptic manipulator based on the partial observation of human movement. In this chapter, we introduce the BIP proposed by Campbell et al. and show how we use the BIP.

BIP is a learning-from-demonstration framework for human-robot interaction. Using the training demonstrations of human-robot interactions, BIP models the relationship between the interacting agents. Firstly, the interaction demonstration $\boldsymbol{X}_{1:T}$ is represented as follows:

$$\boldsymbol{X}_{1:T} = [\boldsymbol{x_1}, \dots, \boldsymbol{x_T}] \in \mathbb{R}^{(\boldsymbol{D_o}+\boldsymbol{D_c}) \times \boldsymbol{T}} \quad , \qquad \text{(VI.1)}$$

where D_o and D_c represents DoF of the observed agent and DoF of the controlled agent, respectively. For practical reason, $\boldsymbol{X}$ is decomposed into latent space. Each dimension $d \in (D_o + D_c)$ of $\boldsymbol{X}$ is approximated to the weighted linear combination of the time-dependent basis function.

$$\boldsymbol{X^d} = [x_1^d, \dots, x_t^d], \qquad \text{(VI.2)}$$

$$x_t^d = \boldsymbol{\phi}_{\emptyset(t)}^d \boldsymbol{w}^d + \epsilon_x, \qquad \text{(VI.3)}$$

where $\boldsymbol{\phi}_{\emptyset(t)}^d \in \mathbb{R}^{1 \times B^d}$ is a row vector of basis function B^d, $\emptyset(t)$ represents the relative phase of time, $\boldsymbol{w}^d \in \mathbb{R}^{B^d \times 1}$ is weights that can be calculated by simple linear regression, and ϵ_x is i.i.d. Gaussian noise. Then we can represent the basis transformation h as follows:

$$\boldsymbol{x_t} = h(\phi(t), \boldsymbol{w}). \qquad \text{(VI.4)}$$

For the simultaneous estimation of the phase and phase velocity, the state vector is represented as $\boldsymbol{s} = [\phi, \dot{\phi}, \boldsymbol{w}]$, where $\dot{\phi}$ represents the phase velocity, and its probability distribution is defined as follows:

$$p(\boldsymbol{s}_t | \boldsymbol{X}_{1:t}, \boldsymbol{s_0}) \propto p(\boldsymbol{x}_t | \boldsymbol{s}_t) p(\boldsymbol{s}_t | \boldsymbol{X}_{\boldsymbol{1:t-1}}, \boldsymbol{s}_0) \qquad \text{(VI.5)}$$

The goal of BIP is to infer the $\boldsymbol{s}$ with underlying latent model $\boldsymbol{w}$, by calculating the posterior density $p(\boldsymbol{s}_t | \boldsymbol{X}_{1:t}, \boldsymbol{s}_0)$. The posterior density is calculated by a two-step recursively, state prediction and measurement update with an extended Kalman filter. Based on Markov assumption, the state prediction density is defined as follows:

$$p(\boldsymbol{s}_t | \boldsymbol{X}_{\boldsymbol{1:t-1}}, \boldsymbol{s}_0) = \int p(\boldsymbol{s}_t | \boldsymbol{s}_{t-1}) \, p(\boldsymbol{s}_{t-1} | \boldsymbol{X}_{1:t-1}, \boldsymbol{s}_0) d\boldsymbol{s}_{t-1}. \qquad \text{(VI.6)}$$

By assuming the estimated errors with the Kalman filter is normally distributed,

$$p(\boldsymbol{s}_t | \boldsymbol{X}_{1:t}, \boldsymbol{s}_0) = N(\boldsymbol{\mu}_{t|t}, \boldsymbol{\Sigma}_{t|t}), \qquad \text{(VI.7)}$$

$$p(\boldsymbol{s}_t | \boldsymbol{X}_{1:t-1}, \boldsymbol{s}_0) = N(\boldsymbol{\mu}_{t|t-1}, \boldsymbol{\Sigma}_{t|t-1}), \qquad \text{(VI.8)}$$

$$\boldsymbol{\mu}_{t|t} = \boldsymbol{G} \boldsymbol{\mu}_{t-1|t-1} \,, \qquad \text{(VI.9)}$$

$$\boldsymbol{\Sigma}_{t|t-1} = \boldsymbol{G} \boldsymbol{\Sigma}_{t-1|t-1} \boldsymbol{G}^{\boldsymbol{T}} + \boldsymbol{Q}_t, \qquad \text{(VI.10)}$$

$$\boldsymbol{G} = \begin{bmatrix} 1 & \Delta t & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}, \tag{VI.11}$$

where **Q** is the state transition noise, e.g., discrete white noise.

By linearizing the nonlinear observation function h(·) via first-order Taylor approximation,

$$\boldsymbol{H_t} = \frac{\partial h(\boldsymbol{s}_t)}{\partial \boldsymbol{s}_t} \tag{VI.12}$$

we finally get the measurement update as follows:

$$\boldsymbol{K}_t = \boldsymbol{\Sigma}_{t|t-1}\boldsymbol{H}_t^{\boldsymbol{T}}\left(\boldsymbol{H}_t\boldsymbol{\Sigma}_{t|t-1}\boldsymbol{H}_t^{\boldsymbol{T}} + \boldsymbol{R}_t\right)^{-1}, \tag{VI.13}$$

$$\boldsymbol{\mu}_{t|t} = \boldsymbol{\mu}_{t|t-1} + \boldsymbol{K}_t\left(\boldsymbol{x}_t - \boldsymbol{h}(\boldsymbol{\mu}_{t|t-1})\right), \tag{VI.14}$$

$$\boldsymbol{\Sigma}_{t|t} = (\boldsymbol{I} - \boldsymbol{K}_t\boldsymbol{H}_t)\boldsymbol{\Sigma}_{t|t-1}, \tag{VI.15}$$

where $\boldsymbol{K}$ is Kalman gain, and $\boldsymbol{R}_t$ is the Gaussian measurement noise about the sensor observation $\boldsymbol{x}_t$.

In order to show the encountered-type haptic response generation using BIP, we use a seven-DoF robot manipulator as a controlled agent, and the observed agent is a human hand with 18-DoF. We track the five fingertips and center of the palm point, and each point has three-DoF. We generate 47 training trajectories, and we conduct the basis spaces selection comparing the Gaussian function, sigmoidal function, and polynomial functions. We select the polynomial function with both the lowest Bayesian Information Criterion and Aikake Information Criterion

C. Demonstration and Result

1. System Implementation

The hardware setup is the same as described in Chapter V. E, except for the HMD and IR sensor. Since the Oculus Quest with four cameras on the sides tracks the user's hand even in the user’s peripheral vision, we replace the Oculus Rift and Leap Motion sensor mounted on the HMD with the Oculus Quest and its embedded monochrome cameras. We maintain the two independent computing platforms: one for visual rendering and the other for haptic and robotic control.

We also inherit the software system from Chapter V. E, and the computing component for haptic simulation is added. To this end, the IntPrim Python library with IntPrim ROS framework [14] is used, and the robot simulator CoppeliaSim was controlled via IntPrim ROS framework. We use C++, Python, and Lua programming languages for the robot manipulator simulation and for generating a BIP model.

2. Experiment

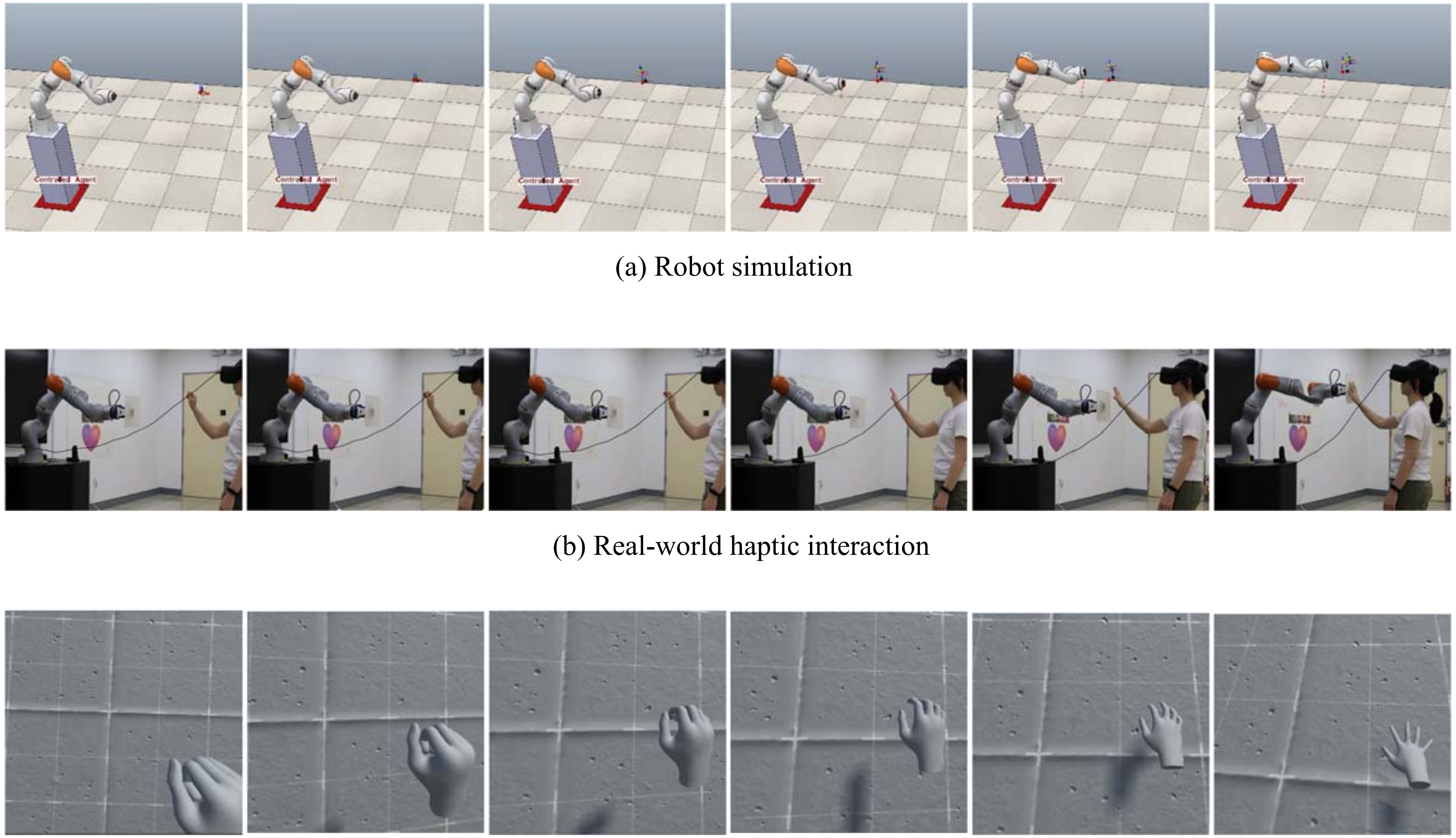
We conduct the experiment to verify the accuracy of the contact prediction model. We tested the online scenario where the user attempts to touch a static virtual panel in VR, in the same way as generating the training data in Chapter VI.B.1. In VR, a virtual panel is provided to the participant, and then the participant reaches out to touch a panel. The system predicts the contact point and we calculate the discrepancy between the predicted contact point and the location of the virtual panel. One virtual panel is provided to the user for a single contact prediction test. We sample 30 points in the manipulator’s per-plane reachability maps and use the points as virtual panels’ location, the contact point.

During the experiment, a vertical flat panel positioned at the sampled point is visually rendered, and the user wearing an HMD performs a hand gesture to touch the given panel (Figure VI-2). Then the user’s hand position is represented in a robot simulation environment, and the haptic manipulator in the simulation environment moves along the contact prediction point in response to the user. Since this experiment aims to verify the accuracy of the contact prediction, we do not provide the user with physical haptic feedback during the experiment, and the contact point is predicted purely from the user’s hand trajectory.

3. Result

The experimental results show that the contact prediction model successfully predicts the contact point. All 30 tests succeeded in contact prediction to provide encountered-type haptic feedback to the user. The absolute mean error of contact prediction in x, y, and z direction w.r.t. the robot base frame was 3.6194cm, 5.4275cm, 6.951cm, respectively. The largest errors in the x, y, and z direction were 9.9579cm, 12.826cm, and 12.942cm, respectively. Since we use $20cm \times 20cm$ rigid panel as an encountered-type haptic display, we assume that the system could provide a haptic response according to the contact even if there is an error.

As shown in Figure VI-3., we also test the scenario consists of simply touching a virtual wall with physical haptic interaction. Based on the observation of the partial trajectory of the participant's hand, the system generates a robot trajectory using a probability distribution model (Figure VI-3.-(a)). When the robot moves in response and detects the contact force, the robot holds its configuration to deliver haptic feedback to the user (Figure VI-3.-(b)).



(a) Robot simulation

(b) Real-world haptic interaction

(c) VR interaction

Figure VI-3. Inferencing encountered-type haptic manipulator response based on observation of user hand gesture

4. Discussion

The demonstration shows that our encountered-type haptic system successfully generates the end-to-end haptic response only by observing the human hand gesture without relying on the previous contact point prediction method in VR.

Our approach to generating haptic training data using VR and simulation environments results in a less time-consuming process. Unlike the general HRI training data generation methods, such as kinesthetic teaching and teleoperation, we utilize a simulation environment where robot simulation data and user hand tracking data coexist. Consequently, all the states can be recorded at once in a simulation environment that does not require a spatial and temporal synchronization of tracked data.

Interestingly, our system successfully generated encountered-type haptic feedback using BIP without utilizing any of the geometric information of virtual objects that is required by the conventional contact prediction method. This result indicates that the hand motion for touching a virtual object, specifically the hand gesture with fingers spread out, is enough to infer the haptic response spatially and temporally. Although more complicated scenarios with multiple virtual objects need to be tested, our method surely reduces the computational cost compared to the contact point prediction and robot control.

When we tested the hand motion with the palm open, the trained model failed to predict the contact point. A possible cause of this phenomenon is that the contact prediction can be difficult without finger movement data, as finger movement plays an important role in inferring the interaction phase and haptic response. Thus, there must be specific hand gestures suitable for inferring a haptic response depending on the interaction task. Based on this observation, providing appropriate robot interaction without object information or object-centric calculation can be possible only if the task-specific hand motion is defined, not only for virtual reality but also for real-world interaction.

Even though we successfully demonstrated haptic interaction using contact prediction based on BIP, a few centimeters of contact prediction errors still exist. Furthermore, as the contact point gets closer to the ground, the contract prediction error has increased. We believe this is due to the lack of sufficient interactions in training data that include such scenarios. To increase the contact prediction accuracy, more demonstration data for training the contact prediction model is needed and the multimodal learning for contact prediction using head orientation, eye-tracking data, and contact force might be a possible solution. Also, it will be useful to verify end-to-end haptic response generation via leave-one-out cross-validation to examine the accuracy by comparing the haptic manipulator trajectories of demonstration data and generated data by the BIP model. Lastly, a statistical analysis on how quickly the system can provide an encountered-type haptic response compared to the previous contact prediction method will be desired.

VII. Conclusions and Future Work

In this dissertation, we proposed the encountered-haptic system that provides various physical properties of virtual objects by kinesthetic/tactile haptic feedback representing static/dynamic virtual objects, that allows user active/passive haptic exploration in a large workspace. To achieve this goal, we used an off-the-shelf collaborative robot as an encountered-type haptic device and controlled the manipulator by position, force, and impedance control. We proposed the per-plane reachability maps that store the robot reachability for each virtual plane and used precomputed per-plane reachability maps for real-time encountered-type haptic rendering. We successfully demonstrate our system to provide an illusion to the user in a virtual environment with touch sensation to the surrounding environment.

For texture rendering without an additional actuator, we propose a macro-scale roughness-synthesizing method for an encountered-type haptic display using the spatiotemporal encoding of roughness. The encoding scheme is based on a geometric grid of hemiellipsoidal bumps with the capability of changing orientation and velocity relative to user motion. Our psychophysical experiments show that both the orientation and the velocity impact perceived roughness, which suggests that users feel the same surface of our model differently in terms of roughness when scanning orientation or velocity changes. Moreover, we demonstrate that simulating a wide range of perceived roughness with a specific target roughness level is possible through both active and passive touch sensation.

To generate an end-to-end haptic response of the manipulator for multiple virtual objects, we propose the contact prediction method by observing the VR user’s interacting motion using Bayesian Interaction Primitives. We also proposed a method for generating haptic demonstration data using real-world tracking data and robot simulation data, which reduces the cost of obtaining HRI data and avoids spatiotemporal data synchronization. We show that our system can be successfully integrated with IntPrim framework [14] and generate an encountered-type haptic response to the VR participants.

By addressing the following limitations and future work, it is expected to improve the encountered-type haptic system so that VR participants are immersed in the VR.

- **Force feedback considering various physical properties:** In this dissertation, we focused on providing stiffness and constrained damping through impedance control. By considering other physical terms such as mass, inertia, and damping of the virtual objects, the physical properties of the virtual object that can be provided through the encountered-type haptic system can be further expanded.

- **Generate various tactile feedback:** While our texture synthesizing method provides various levels of macro-scale roughness using the plane surface display, the other field of tactile information may play a more critical role depending on the VR interaction task. Consequently, further research that considers the various cutaneous feedback, such as surface geometry or surface temperature, is required. Since our texture rendering is highly dependent on the user scanning direction, which determines the bump scanning frequency, further works should consider the uncertainty originated from the user's movement, and an analytical model for the bump placement is necessary. Also, future work on the problem of providing both force feedback and tactile feedback is needed. This dissertation focused on texture rendering with force feedback for a static object, but additional research is needed for force and texture rendering for dynamic objects.

- **Contact prediction for multiple virtual objects and various interactions:** We proposed the contact point prediction using learning from demonstrations and show our method successfully provides the encountered-type haptic feedback of touching a static object. Contact prediction for multiple virtual objects with different physical properties remains an open question. Also, it is required to evaluate and statistically analyze the accuracy, speed, computational cost of our method. To generate haptic feedback for various haptic interactions, future work for training the interactions with multiple task-specific hand gestures such as grasping or pushing objects is needed. Also, a future search could examine the multimodal learning for contact prediction considering user gaze direction or eye-tracking data with hand tracking data and study how to determine the weight of each data set.

Bibliography

[1] T. H. Massie, “Design of a three degree of freedom force-reflecting haptic interface,” B.S. thesis, MIT, May 1993.

[2] P. Cipresso, I. A. C. Giglioli, M. A. Raya, and G. Riva, “The past, present, and future of virtual and augmented reality research: A network and cluster analysis of the literature,” *Frontiers in Psychology*, vol. 9, pp. 2086, 2018.

[3] Y. Hirata and M. Sato, “3-dimensional Interface Device For Virtual Work Space,” *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 889–896, 1992.

[4] Facebook, “Oculus Quest.” https://www.oculus.com/quest/

[5] Manus, “Prime 2 Haptic.” https://www.manus-vr.com/haptic-gloves

[6] Y. Kim, H. J. Kim, and Y. J. Kim, “Encountered-type haptic display for large VR environment using per-plane reachability maps,” in *Computer Animation and Virtual Worlds*, vol. 29, no. 3–4, p. e1814, 2018.

[7] W. A. Mcneely, “Robotic Graphics: A New Approach to Force Feedback for Virtual Reality,” in *Proc. of IEEE Virtual Reality Annual International Symposium*, Sep 1993, pp. 336-341.

[8] H. Culbertson, S. B. Schorr, and A. M. Okamura, “Haptics: The Present and Future of Artificial Touch Sensation,” *Annu. Rev. Control Robot. Auton. Syst.*, vol. 1, pp. 385–409, 2018.

[9] B. Araujo, R. Jota, V. Perumal, J. X. Yao, K. Singh, and D. Wigdor, “Snake charmer: Physically enabling virtual objects,” In *Proc. of the TEI'16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction*, pp. 218-226, 2016.

[10] P. Abtahi, B. Landry, J. J. Yang, M. Pavone, S. Follmer, and J. A. Landay, “Beyond the force: Using quadcopters to appropriate objects and the environment for haptics in virtual reality,” in *Proc. of the CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1-13.

[11] P. Abtahi and S. Follmer, “Visuo-haptic illusions for improving the perceived

performance of shape displays," *In Proc. of the 2018 CHI Conference on Human Factors in Computing Systems, pp. 1-13. 2018.*

[12] S. Okamoto, H. Nagano, and Y. Yamada, "Psychophysical dimensions of tactile perception of textures," *IEEE Transactions on Haptics*, vol. 6, no. 1, pp. 81–93, 2013.

[13] S. J. Lederman, "The perception of surface roughness by active and passive touch," *Bulletin of the Psychonomic Society*, vol. 18, pp. 253–255, 1981.

[14] J. Campbell and H. ben Amor, "Bayesian interaction primitives: A slam approach to human-robot interaction," in *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 379–387, 2017.

[15] S. Tachi, T. Maeda, R. Hirata, and H. Hoshino, "A construction method of virtual haptic space," *Proceedings of the 4th International Conference on Artificial Reality and Tele-Existence (ICAT'94)*, 1994.

[16] Y. Yokokohji, R. L. Hollis, and T. Kanade, "WYSIWYF Display: A Visual/Haptic Interface to Virtual Environment," *Presence*, vol. 8, no. 4, pp. 412-434, 1999.

[17] Y. Yokokohji, R. L. Hollis, and T. Kanade, "What you can see is what you can feel-development of a visual/haptic interface to virtual environment," in *Proc. of the IEEE 1996 Virtual Reality Annual International Symposium*, March 1996, pp. 46-53.

[18] E. Vonach, C. Gatterer, and H. Kaufmann, "VRRobot: Robot actuated props in an infinite virtual environment," in *2017 IEEE Virtual Reality (VR)*, pp. 74–83, March 2017.

[19] V. Mercado, M. Marchal, and A. Lécuyer, "ENTROPiA: Towards Infinite Surface Haptic Displays in Virtual Reality Using Encountered-Type Rotating Props," in *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 3, pp. 2237–2243, Mar. 2021.

[20] Y. Yokokohji, J. Kinoshita, and T. Yoshikawa, "Path planning for encountered-type haptic devices that render multiple objects in 3d space," in *Proceedings IEEE Virtual Reality 2001*, 2001, pp. 271–278.

[21] K. Shigeta, S. Yuji, and Y. Yokokohji, “Motion planning of encountered-type haptic device for multiple fingertips based on minimum distance point information,” in *Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC’07)*, 2007, pp. 188–193.

[22] Y. Yixian, K. Takashima, A. Tang, T. Tanno, K. Fujita, and Y. Kitamura, “ZoomWalls: Dynamic walls that simulate haptic infrastructure for room-scale VR world,” in *UIST 2020 - Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, Oct. 2020, pp. 223–235

[23] Y. Wang, Z. Chen, H. Li, Z. Cao, H. Luo, T. Zhang, K. Ou, J. Raiti, C. Yu, S. Patel, and Y. Shi, “MoveVR: Enabling Multiform Force Feedback in Virtual Reality using Household Cleaning Robot,” In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1-12, 2020.

[24] E. J. Gonzalez, P. Abtahi, and S. Follmer, “REACH+: Extending the reachability of encountered-type haptics devices through dynamic redirection in VR,” in *UIST 2020 - Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, Oct. 2020, pp. 236–248

[25] R.Suzuki, H. Hedayati, C. Zheng, J. L. Bohn, D. Szafir, E. Y. L. Do, M. D. Gross, and D. Leithinger, “RoomShift: Room-scale Dynamic Haptics for VR with Furniture-moving Swarm Robots,” In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems,* pp. 1-11, 2020.

[26] D. Fitzgerald and H. Ishii, “Mediate: A spatial tangible interface for mixed reality,” in *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018, pp. 1–6.

[27] A. F. Siu, E. J. Gonzalez, S. Yuan, J. B. Ginsberg, and S. Follmer, “Shapeshift: 2D spatial manipulation and self-actuation of tabletop shape displays for tangible and haptic interaction,” in *Conference on Human Factors in Computing Systems - Proceedings*, Apr. 2018, vol. 2018-April.

[28] X. de Tinguy, T. Howard, C. Pacchierotti, M. Marchal, and A. Lécuyer, “Weatavix: wearable actuated tangibles for virtual reality experiences,” in *International*

Conference on Human Haptic Sensing and Touch Enabled Computer Applications, 2020, pp. 262–270.

[29] R. Kovacs *et al.*, “Haptic PIVOT: On-Demand Handhelds in VR,” in *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, New York, NY, USA: Association for Computing Machinery, 2020, pp. 1046–1059.

[30] M. Abdullah, M. Kim, W. Hassan, Y. Kuroda, and S. Jeon, “HapticDrone: An Encountered-Type Kinesthetic Haptic Interface with Controllable Force Feedback: Initial Example for 1D Haptic Feedback,” in *Adjunct Publication of the 30th Annual ACM Symposium on User Interface Software and Technology - UIST ’17*, pp. 115–117, 2017.

[31] M. Abdullah, M. Kim, W. Hassan, Y. Kuroda, and S. Jeon, “HapticDrone: An Encountered-Type Kinesthetic Haptic Interface with Controllable Force Feedback : Example of Stiffness and Weight Rendering,” in *2018 IEEE Haptics Symposium (HAPTICS)*, pp. 334–339, 2018.

[32] M. Hoppe, P. Knierim, T. Kosch, M. Funk, L. Futami, S. Schneegass, N. Henze, A. Schmit, and T. Machulla, “VRHapticDrones: Providing haptics in virtual reality through quadcopters,” in *Proc. of the 17th International Conference on Mobile and Ubiquitous Multimedia,* 2018, pp. 7–18.

[33] A. Horie, M. Y. Saraiji, Z. Kashino, and M. Inami, "EncounteredLimbs: A Room-scale Encountered-type Haptic Presentation using Wearable Robotic Arms," *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*, 2021, pp. 260-269.

[34] J. L. Rodríguez, R. Velázquez, C. Del-Valle-soto, S. Gutiérrez, J. Varona, and J. Enríquez-Zarate, “Active and passive haptic perception of shape: Passive haptics can support navigation,” *Electronics (Switzerland)*, vol. 8, no. 3, Mar. 2019.

[35] B. Siciliano and O. Khatib, *Springer handbook of robotics*. Springer, 2016.

[36] F. Zacharias, C. Borst, and G. Hirzinger, "Capturing robot workspace structure: representing robot capabilities," *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 3229-3236

[37] J. Dong and J. C. Trinkle, “Orientation-based reachability map for robot base

placement," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 1488–1493.

[38] Y. Guan and K. Yokoi, "Reachable space generation of a humanoid robot using the monte carlo method," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 1984–1989.

[39] N. Vahrenkamp, T. Asfour, and R. Dillmann, "Robot placement based on reachability inversion," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 1970–1975.

[40] F. Zacharias, W. Sepp, C. Borst, and G. Hirzinger, "Using a model of the reachable workspace to position mobile manipulators for 3-d trajectories," *2009 9th IEEE-RAS International Conference on Humanoid Robots*, 2009, pp. 55-61.

[41] E. A. Suma, Z. Lipps, S. Finkelstein, D. M. Krum, and M. Bolas, "Impossible spaces: Maximizing natural walking in virtual environments with self-overlapping architecture," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 4, pp. 555–564, 2012.

[42] K. Vasylevska, H. Kaufmann, M. Bolas, and E. A. Suma, "Flexible spaces: Dynamic layout generation for infinite walking in virtual environments," in *2013 IEEE Symposium on 3D User Interfaces (3DUI)*, 2013, pp. 39–42.

[43] S. Razzaque, Z. Kohn and M. C. Whitton, "Redirected walking," *Proc. EUROGRAPHICS*, pp. 105-106, 2001.

[44] T. Nescher, Y.-Y. Huang, and A. Kunz, "Planning redirection techniques for optimal free walking experience using model predictive control," in *2014 IEEE Symposium on 3D User Interfaces (3DUI)*, 2014, pp. 111–118.

[45] M. A. Zmuda, J. L. Wonser, E. R. Bachmann, and E. Hodgson, "Optimizing constrained-environment redirected walking instructions using search techniques," *IEEE transactions on visualization and computer graphics*, vol. 19, no. 11, pp. 1872–1884, 2013.

[46] F. Steinicke, G. Bruder, K. Hinrichs, J. Jerald, H. Frenz, and M. Lappe, "Real walking through virtual environments by redirection techniques," *JVRB-Journal of*

Virtual Reality and Broadcasting, vol. 6, no. 2, 2009.

[47] Q. Sun, L. Y. Wei, and A. Kaufman, "Mapping virtual and physical reality," in *ACM Transactions on Graphics*, Jul. 2016, vol. 35, no. 4.

[48] G. Bruder, P. Lubos, and F. Steinicke, "Cognitive resource demands of redirected walking," *IEEE transactions on visualization and computer graphics*, vol. 21, no. 4, pp. 539–544, 2015.

[49] N. C. Nilsson, S. Serafin, and R. Nordahl, "Establishing the range of perceptually natural visual walking speeds for virtual walking-in-place locomotion," *IEEE transactions on visualization and computer graphics*, vol. 20, no. 4, pp. 569–578, 2014.

[50] K. Matsumoto, Y. Ban, T. Narumi, T. Tanikawa, and M. Hirose, "Curvature manipulation techniques in redirection using haptic cues," in *2016 IEEE Symposium on 3D User Interfaces (3DUI)*, 2016, pp. 105–108.

[51] K. Matsumoto, Y. Ban, T. Narumi, Y. Yanase, T. Tanikawa, and M. Hirose, "Unlimited corridor: Redirected walking techniques using visuo haptic interaction," In *ACM SIGGRAPH 2016 Emerging Technologies*, pp. 1-2. 2016.

[52] S. Choi and K. J. Kuchenbecker, "Vibrotactile display: Perception, technology, and applications," *Proceedings of the IEEE*, vol. 101, no. 9, pp. 2093–2104, 2013.

[53] J. M. Romano and K. J. Kuchenbecker, "Creating realistic virtual textures from contact acceleration data," *IEEE Transactions on Haptics*, vol. 5, no. 2, pp. 109–119, 2012.

[54] Y. Ujitoko, Y. Ban, and K. Hirota, "Modulating Fine Roughness Perception of Vibrotactile Textured Surface using Pseudo-haptic Effect," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 5, pp. 1981–1990, 2019.

[55] Y. Vardar, A. Isleyen, M. K. Saleem, and C. Basdogan, "Roughness perception of virtual textures displayed by electrovibration on touch screens," in *2017 IEEE World Haptics Conference, (WHC 2017)*, June 2017, pp. 263-268.

[56] K. Ito, S. Okamoto, H. Elfekey, H. Kajimto, and Y. Yamada, "A texture display using vibrotactile and electrostatic friction stimuli surpasses one based on either type of

stimulus," *2017 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2017*, vol. 2017-Janua, pp. 2343–2348, 2017.

[57] A. Fedoseev, A. Tleugazy, L. Labazanova, and D. Tsetserukou, "TeslaMirror: Multistimulus Encounter-Type Haptic Display for Shape and Texture Rendering in VR," In *ACM SIGGRAPH 2020 Emerging Technologies*, pp. 1-2. 2020.

[58] K. S. Hale and K. M. Stanney, "Deriving haptic design guidelines from human physiological, psychophysical, and neurological foundations," *IEEE Computer Graphics and Applications*, vol. 24, no. 2, pp. 33–39, 2004.

[59] S. J. Lederman and R. L. Klatzky, "Relative Availability of Surface and Object Properties during Early Haptic Processing," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 23, no. 6, pp. 1680–1707, 1997.

[60] L. A. Jones, *HAPTICS.* The MIT Press, 2018.

[61] C. J. Cascio and K. Sathian, "Temporal cues contribute to tactile perception of roughness," *The Journal of Neuroscience,* vol. 21, no. 14, pp. 5289–5296, 2001.

[62] C. Connor and K. Johnson, "Neural coding of tactile texture: comparison of spatial and temporal mechanisms for roughness perception," *The Journal of Neuroscience*, vol. 12, no. 9, pp. 3414–3426, 1992.

[63] S. J. Lederman and M. M. Taylor, "Fingertip force, surface geometry, and the perception of roughness by active touch," *Perception & Psychophysics*, vol. 12, no. 5, pp. 401–408, 1972.

[64] C. Connor, S. Hsiao, J. Phillips, and K. Johnson, "Tactile roughness: neural codes that account for psychophysical magnitude estimates.," *The Journal of Neuroscience,* vol. 10, no. 12, pp. 3823–3836, 1990.

[65] A. Dépeault, E. M. Meftah, and C. E. Chapman, "Tactile perception of roughness: Raised-dot spacing, density and disposition," *Experimental Brain Research*, vol. 197, no. 3, pp. 235–244, 2009.

[66] M. M. Taylor and S. J. Lederman, "Tactile roughness of grooved surfaces: A model and the effect of friction," *Perception & Psychophysics*, vol. 17, no. 1, pp. 23–36, 1975.

[67] S. J. LEDERMAN, "Tactual roughness perception: spatial and temporal determinants," *Canadian journal of psychology*, vol. 37, no. 4, pp. 498–511, 1983.

[68] E. Gamzu and E. Ahissar, "Importance of temporal cues for tactile spatial-frequency discrimination," *Journal of Neuroscience*, vol. 21, no. 18, pp. 7416–7427, 2001.

[69] A. M. Smith, C. E. Chapman, M. Deslandes, J. S. Langlais, and M. P. Thibodeau, "Role of friction and tangential force variation in the subjective scaling of tactile roughness," *Experimental Brain Research*, vol. 144, no. 2, pp. 211–223, 2002.

[70] M. Meenes and M. J. Zigler, "An Experimental Study of the Perceptions Roughness and Smoothness," *The American Journal of Psychology*, vol. 34, no. 4, pp. 542–549, 1923.

[71] S. J. Lederman and R. L. Klatzky, "Hand movements: A window into haptic object recognition," *Cognitive Psychology*, vol. 19, no. 3, pp. 342–368, 1987.

[72] S. Jeon, "Haptic rendering of curved surface by bending an encountered-type flexible plate," *IEICE Transactions on Information and Systems*, vol. 99, no. 7, pp. 1862–1870, 2016.

[73] L.-P. Cheng, E. Ofek, C. Holz, H. Benko, and A. D. Wilson, "Sparse Haptic Proxy: Touch Feedback in Virtual Environments Using a General Passive Prop," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 2017, pp. 3718–3728.

[74] S. Schaal, "Is imitation learning the route to humanoid robots?," *Trends in cognitive sciences*, vol. 3, no. 6, pp. 233–242, 1999.

[75] Z. Wang, M. P. Deisenroth, H. ben Amor, D. Vogt, B. Schölkopf, and J. Peters, "Probabilistic modeling of human movements for intention inference," *Proceedings of Robotics: Science and Systems, VIII*, 2012.

[76] S. Ikemoto, H. ben Amor, T. Minato, B. Jung, and H. Ishiguro, "Physical human-robot interaction: Mutual learning and adaptation," *IEEE robotics & automation magazine*, vol. 19, no. 4, pp. 24–35, 2012.

[77] D. Lee, C. Ott, and Y. Nakamura, "Mimetic communication model with compliant physical contact in human—humanoid interaction," *The International Journal of*

Robotics Research, vol. 29, no. 13, pp. 1684–1704, 2010.

[78] H. ben Amor, G. Neumann, S. Kamthe, O. Kroemer, and J. Peters, “Interaction primitives for human-robot cooperation tasks,” in *2014 IEEE international conference on robotics and automation (ICRA)*, 2014, pp. 2831–2837.

[79] S. Schaal, “Dynamic movement primitives-a framework for motor control in humans and humanoid robotics,” in *Adaptive motion of animals and machines*, Springer, 2006, pp. 261–280.

[80] J. Campbell, S. Stepputtis and H. Ben Amor, "Probabilistic multimodal modeling for human-robot interaction tasks," *Robotics: Science and Systems*, 2019.

[81] B. Akgun and K. Subramanian, “Robot learning from demonstration: kinesthetic teaching vs. teleoperation,” *Unpublished manuscript*, pp. 7, 2011.

[82] K. Bagewadi, J. Campbell, and H. B. Amor, “Multimodal Dataset of Human-Robot Hugging Interaction,” *arXiv preprint*, 2019.

[83] V. R. Mercado, M. Marchal, and A. Lecuyer, “‘Haptics On-Demand’: A Survey on Encountered-Type Haptic Displays,” *IEEE Transactions on Haptics*, 2021.

[84] Ultraleap, “Leap Motion Controller,” https://www.ultraleap.com/product/leap-motion-controller/

[85] D. Cohen-Or, C. Greif, T. Ju, N. J. Mitra, and H. Zhang, “A Sampler of Useful Computational Tools for Applied Geometry, Computer Graphics, and Image Processing.” CRC Press, 2015.

[86] Neville Hogan, “Impedance Control: An Approach to Manipulation,” in *1984 American Control Conference*, 1984, pp. 304–313.

[87] M. C. Lin, M. Otaduy, M. C. Lin, and M. Otaduy, “Haptic Rendering: Foundations, Algorithms and Applications.” AK Peters, Ltd., 2008.

[88] Z. Ugray, L. Lasdon, J. Plummer, F. Glover, J. Kelly, and R. Marti, “Scatter Search and Local NLP Solvers: A Multistart Framework for Global Optimization,*”* *Informs J Comput*. vol. 19, no. 3 pp. 328-840, 2007.

[89] T. L. Brooks, “Telerobotic response requirements,” in *1990 IEEE International Conference on Systems, Man, and Cybernetics Conference Proceedings*, 1990, pp.

113–120.

[90] W. D. Games, “Dreadhalls.” [Online]. Available: https://www.dreadhalls.com

[91] Y. Kim, S. Kim, U. Oh, and Y. J. Kim, “Synthesizing the Roughness of Textured Surfaces for an Encountered-Type Haptic Display Using Spatiotemporal Encoding,” *IEEE Transactions on Haptics*, vol. 14, no. 1, pp. 32–43, Jan. 2021.

[92] J. M. Goodman and S. J. Bensmaia, “A variation code accounts for the perceived roughness of coarsely textured surfaces,” *Scientific Reports*, vol.7, p. 46699, April 2017.

[93] E. M. Meftah, L. Belingard, and C. E. Chapman, “Relative effects of the spatial and temporal characteristics of scanned surfaces on human perception of tactile roughness using passive touch,” *Exp. Brain Res*, vol. 132, pp. 351–361, 2000.

[94] L. A. Jones and H. Z. Tan, “Application of psychophysical techniques to haptic research,” *IEEE Transactions on Haptics*, vol. 6, no. 3, pp. 268–284, 2013.

[95] S. S. Stevens, “On the psychophysical law,” *Psychological Review*, vol. 64, no. 3, pp. 153–181, 1957.

[96] J. O. Wobbrock, L. Findlater, D. Gergle, and J. J. Higgins, “The Aligned Rank Transform for nonparametric factorial analyses using only ANOVA procedures,” in *Proc. of the SIGCHI Conference on Human Factors in Computing Systems,* pp. 143-146, 2011.

[97] S. Ballesteros, F. Muñoz, M. Sebastián, B. García, and J. M. Reales, “ERP evidence of tactile texture processing: Effects of roughness and movement,” *Proceedings - 3rd Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, World Haptics 2009*, pp. 166–171, 2009.

[98] R. M. Peters, E. Hackeman, and D. Goldreich, “Diminutive digits discern delicate details: Fingertip size and the sex difference in tactile spatial acuity,” *Journal of Neuroscience*, vol. 29, no. 50, pp. 15756–15761, 2009.

국문초록

가상현실 몰입감 향상을 위한 협업 로봇 기반 조우형 햅틱스

김예솔
컴퓨터공학과
이화여자대학교 대학원

몰입감 있는 가상현실(VR)에 대한 관심이 높아짐에 따라 다양한 유형의 햅틱 시스템이 제안 및 연구되고 있다. 다양한 햅틱 인터페이스의 폼팩터 중 조우형(Encountered-type) 햅틱 시스템은 사용자와의 불필요한 접촉을 유지하지 않고, 자유로운 터치와 움직임에 따른 충돌 역감을 제공하여 가상현실에서의 자연스러운 상호작용을 보장한다. 조우형 햅틱 시스템의 특성에 따라, 사용자의 작업 공간에 대한 햅틱 매니퓰레이터의 도달가능성(reachability)이 고려되어야 하며, 맨손으로 조우형 햅틱 피드백을 느끼는 사용자를 위해 운동 감각(Kinesthetic feedback)과 촉각(Tactile feedback)을 동시에 제공하는 햅틱 렌더링 알고리즘이 필요하다.

본 논문에서는 자유로운 상호작용을 보장하면서도 가상현실로의 몰입감을 증대시키는 조우형 햅틱 시스템을 소개한다. 연구 목적을 이루기 위해, 다양한 역감 및 질감, 능동적 및 수동적 햅틱 피드백, 정적 및 동적 가상 객체에 대한 햅틱 피드백을 제공하기위한 다목적 랜더링을 제안한다. 다양한 물리적 특성을 나타내는 조우형 햅틱 피드백을 제공하기 위해 7 자유도의 협동 로봇을 햅틱 햅틱 매니퓰레이터로 사용하였으며, 햅틱 매니퓰레이터의 도달가능성을 확인하기 위해 평면별

도달가능성 맵을 제안하였다. 또한, 촉감을 제공함으로써 풍부한 가상현실 경험을 제공하기 위해 조우형 햅틱 디스플레이를 위한 거칠기 합성 방법을 제안하였으며, 강경한 햅틱 랜더링을 위해 사용자의 손 움직임을 관찰하여 접촉 예상 지점을 추론하는 방법을 제안한다.

첫째로, 본 논문에서는 7 자유도의 협업 로봇을 햅틱 디바이스로 사용하여 수직인 벽과 회전하는 문 등의 정적 및 동적 가상 객체로 구성된 실내 VR 환경을 사용자에게 시각적, 촉각적으로 제공하기 위한 조우형 햅틱 시스템을 제안한다. 본 시스템은 사용자의 실시간 손 추적 데이터에 기반하여 직사각형의 단단한 보드가 부착된 로봇 말단의 궤적을 계획함으로써 사용자에게 가상 물체와의 접촉에 따른 역감을 전달한다. 정적 가상 물체의 경우, 사용자와 햅틱 디바이스의 접촉이 유지되는 동안 햅틱 디바이스가 가상 물체 표면에 해당되는 상태를 유지함으로써 역감을 전달한다. 동적 가상 물체의 경우 임피던스 기반으로 햅틱 디바이스를 제어하여 가상 물체를 이동시키는 역감을 전달한다. 또한, 제한된 작업 공간에 따른 문제를 다루기 위해 햅틱 매니퓰레이터의 평면 별 도달가능성 맵을 계산하여 사용자가 만지려고 하는 가상 객체의 각도에 따라 햅틱 피드백의 제공 가능성을 실시간으로 확인하는 방법을 제안한다. 새롭게 제안한 최적화된 평면 별 도달가능성 맵을 사용한 본 시스템은, 사용자 평가 실험을 통해 가상현실 참여자에게 몰입감 있는 가상현실 체험을 제공함을 확인하였다.

복잡한 촉각 제공 장치(tactile actuator)를 추가하지 않고 조우형 햅틱 랜더링을 통해 사용자에게 표면 질감 등의 촉감을 제공하는 것은 어려운 문제이다. 본 논문에서는 조우형 햅틱 디스플레이를 위한 새로운 질감 합성 방법으로 시공간적 거칠기 인코딩을 제안한다. 제안하는 방법은 사용자에게 능동적 터치 감각과 수동적 촉감을 모두 제공하며, 추가적인 촉각 제공장치를 필요로 하지 않는다. 본 연구에서는 거시적 거칠기 인식에 초점을 맞추어, 사용자가 맨손으로 표면을 탐색할 때,

표면을 만지는 방향과 속도에 따라 다양한 거칠기를 느낄 수 있도록 반 타원구 돌기를 이용하여 물체 표면을 기하학적으로 모델링한다. 제안하는 질감 합성 방법은 두 가지 중요한 가설을 기반으로 한다. 첫째, 사람이 인식하는 거칠기는 반 타원형 돌기로 구성된 표면의 방사 방향을 따라 공간적으로 인코딩 될 수 있다. 둘째, 사람이 인식하는 거칠기는 사람 손이 표면을 쓰다듬는 속도에 따라 다르다. 본 연구에서는 시공간적 거칠기 인코딩 방법에 대한 두 가설을 검증하기 위해 심리물리학적 사용자 평가 실험을 수행하였고, 질감 모델의 시공간적 거칠기 인코딩 방법이 사용자의 거칠기 인식에 미치는 주요 효과를 확인하였다. 또한, 본 연구의 경험적 실험을 통해, 사용자가 거칠기를 인코딩한 표면 물체를 쓰다듬는 방향과 각도가 증가할 수록 질감을 더 거칠게 인식한다는 것을 확인하였다. 사용자 평가 결과를 바탕으로, 거칠기 인코딩 인자를 적절하게 선택하여 다양한 시각적 질감에 해당하는 적절한 수준의 거칠기를 합성할 수 있음을 보였다.

가상현실에서의 결정론적 접촉 예측 알고리즘에 의존하지 않고 강경한 햅틱 반응을 생성하기 위해서는 종단간 햅틱 반응을 생성 방법이 필요하다. 햅틱 랜더링 시스템은 가상 객체와 사용자 손 사이의 결정론적 접촉 예측 방법에 크게 의존하기 때문에 가상현실에서의 접촉 예측의 실패는 전체 햅틱 랜더링 시스템의 실패로 이어진다. 본 논문에서는 광선 슈팅과 투사를 이용하여 접촉을 예측하는 기존 방법에 의존하지 않고, 사람 손의 움직임만을 관찰하여 접촉을 예측하는 모방학습 방법을 제안한다. 먼저, 훈련 데이터(training data)로 사용되는 인간-로봇 상호작용 시범 데이터를 생성하기 위해 로봇 시뮬레이터, 가상현실 환경, 실제 환경의 추적 시스템을 통합하여 훈련 데이터를 얻기 위한 환경을 구성하였다. 이후, 시뮬레이션을 통해 움직이는 매니퓰레이터의 관절 경로와 가상 물체와 상호작용하는 사람의 손가락 끝 점 및 손바닥의 중앙 지점의 경로를 기록하였다. 훈련 데이터는 모방학습을 통해 기저 함수 공간에서 확률 분포로 표현된다. 생성된 확률 분포를 사전분포로 사용하여 시스템 실행 시간에 관측한 사용자의 손 움직임에 따라 다음

사용자 손 움직임을 추론하고 로봇 반응을 생성하는 베이지안 필터링을 수행한다. 본 논문에서는 가상현실 참여자의 손 움직임만을 고려하여 종단간 햅틱 반응을 성공적으로 생성하는 것을 보였다. 본 연구의 결과를 통해 가상 물체를 만지기 위한 손의 동작을 통해 시공간적으로 적절한 햅틱 반을 추론할 수 있음을 확인하였다.

감사의글

대학원에 입학한 지 5년 6개월이라는 시간이 흘러 어느 새 논문을 마무리하고 박사학위를 취득하게 되었습니다. 쏜살같이 지나간 듯하면서도, 막막함을 견디며 나아가야 했던 길고도 힘든 시간이었습니다. 어려운 기간을 지나는 중에 많은 분들의 도움을 받아 학위를 마칠 수 있었기에, 감사한 분들께 마음을 전합니다.

먼저, 저에게 연구자의 길을 열어 주신 김영준 지도교수님께 감사의 말씀을 드립니다. 호기심에 이끌려 연구실 인턴십을 시작했던 학부생 때부터 직접 지도해주시고, 연구를 위해서라면 언제나 아낌없는 지원과 격려를 해 주신 덕분에 연구에 대한 흥미와 열정을 잃지 않고 박사학위까지 마무리할 수 있었습니다. 학생들에게 학문에 몰두할 수 있는 환경을 마련해주시고 가까이서 지도해주시는 교수님께 배울 수 있어서 연구에 매진할 수 있었고, 논리적인 사고방식과 문제 해결 능력도 많이 기를 수 있었습니다. 교수님께서 보여주신 연구에 대한 열정과 통찰력, 교수님께서 지도해주신 연구에 대한 자세를 잊지 않고 앞으로 나아가겠습니다.

연구에 대한 어려움이 있을 때 도움을 주신 오유란 교수님께 감사드립니다. 연구 주제와 관련하여 조언을 여쭈었을 때에도 시간을 내어 지도해 주셨고, 특히, 사용자 평가 실험의 설계 및 분석에 있어 큰 도움을 주셔서 연구 결과를 잘 도출할 수 있었습니다. 학위논문 심사위원으로도 참석해주시고 조언해주신 덕분에 논문도 잘 정리할 수 있었습니다. 마음 깊이 감사드립니다.

또한, 바쁘신 가운데 학위 논문 심사위원을 맡아주신 민동보 교수님, 류석창 교수님, 전석희 교수님께 감사드립니다. 제 연구에 대해 관심을 가져 주시고, 아낌없는 조언과 격려를 나누어 주신 덕분에 마지막까지 잘 정리할 수 있었습니다.

그리고 연구실 생활을 함께했던 GLAB 식구들 모두에게 감사드립니다. 연구에 대한 고민을 나눌 수 있었던 송다은, 민혜정 언니, 한경민 박사님, 서정렬 박사님, 졸업한 후배 김현정, 김정민, 김지수, 김지선, 김시연, 김민영 모두 감사드립니다.

연구 시작단계에서 많은 지도와 도움을 주셨던 졸업하신 이영은 언니, 김윤형 언니, 김여진 언니께 정말 감사드립니다. 선배 언니들의 지도와 조언으로 즐겁게 배우고 가장 많이 성장했던 시간이었습니다.

언제나 곁에서 격려해준 친구 이미지에게 감사를 전합니다. 자신감을 잃고 마음이 많이 흔들릴 때마다 곁에서 위로해주고 버팀목이 되어준 덕분에 무사히 학위를 마무리할 수 있었습니다. 항상 응원해준 친구들 이민주, 정지모, 박광빈, 현정원, 이주경, 박근주, 박서현, 구자은에게도 감사를 전합니다.

사랑하는 엄마, 아빠, 할머니, 내 동생 미솔이에게 감사의 마음을 전합니다. 오래 공부하는 조카를 항상 챙겨 주시는 이모, 삼촌들 모두 감사드립니다. 묵묵히 곁을 지켜 주시고 지지해주신 가족이 있어 지금까지 연구할 수 있었습니다. 항상 변함없는 사랑을 주셔서 늘 감사드립니다.

이렇게 많은 분들의 도움으로 박사학위를 마무리할 수 있었습니다. 많은 분들이 주신 도움이 더욱 빛나도록 다른 사람에게 도움을 나누는 사람이 되겠습니다.