Haptic Rendering of Curved Surface by Bending an Encountered-Type Flexible Plate

Seokhee JEON(a), Member

SUMMARY An encountered-type haptic interface generates touch sensation only when a user’s hand “encounters” virtual objects. This paper presents an effective encountered-type haptic interface that enables rendering of surfaces with variable curvature. The key idea is to systematically bend a thin elastic plate so as to create a curved surface with desired curvature, which becomes a contacting end effector that follows the user’s finger and becomes an interface a user can touch when needed. The pose of the curvature is controlled in a way that it corresponds to the curved surfaces of virtual objects and user’s finger position. The idea is realized by attaching two commercial haptic interfaces to both edges of a thin acryl plate and squeezing the plate. This setup allows us to generate a cylindrical object with curvature up to 0.035 mm$^{-1}$ and gives 3DOF position control and 1DOF rotational control of the curved surface. Achievable workspace and curvature range are analyzed, and the feasibility and physical performance are demonstrated through a visuo-haptic grabbing scenario. In addition, a psychophysical experiment shows perceptual competence of the proposed system.

key words: encountered-type haptics, curvature rendering, flexible end effector

1. Introduction

One of the goals of a haptic system is to synthetically generate physical signals for constructing virtual feeling of various attributes of an object, such as shape, stiffness, friction, and surface haptic texture [1]. Among the haptic attributes, the shape of an object is perceived by humans through two exploratory procedures (EP); the multi-finger enclosure such as grasping and the contour following usually done by stroking with a finger [2], [3]. The former supplies the sense of global shape, while the latter provides exact shape data. However, due to the large variety of haptic signals involved in shape perception in the both EPs, developing a general haptic interface for shape display is a challenging task. Common tool-mediated interfaces where a user holds a grip that corresponds to a contacting avatar in virtual environments are usually not suitable for supplying rich set of tactile information such as irregular pressure distribution due to high curvature, shear deformation of skin, and contact location change due to curved surface. This results in the loss of important cues for shape perception [4], [5].

An alternative is “encountered-type” haptic interfaces. First introduced by McNeely et al. [6] and Yokokoji et al. [7], and followed by several researchers, e.g., [8]–[11], the concept allows a user not to be always in contact with the device. Instead, the user’s body part is in contact with the device only when there is an interaction with virtual objects. A pre-shaped end effector is placed at the future trajectory of the user’s hand, leading to an encounter between the user’s hand and the end effector. This approach provides direct touch without any mediating tool, allowing a natural interaction with appropriate tactile sensation. In addition, no need of mechanical coupling between the device and a user’s body part makes a multi-finger interaction more immediate and realistic.

Nevertheless, previous approaches on encountered-type haptic interfaces suffer from limited generality. Customized and pre-shaped end effectors can only represent a specific object, limiting the flexibility of the system. Manually or automatically replacing the end effector can be one option, but it is still not feasible when surface properties should be frequently changed. An ideal solution is to introduce a “universal end effector” where its haptic properties, such as shape and stiffness, can be systematically altered to effectively cope with virtual objects with different haptic property.

This paper presents my very initial steps towards developing the “universal end effector” for encountered-type haptic interfaces. I present a new method providing variable curvature with rich tactile and kinesthetic cues. The core mechanism of the approach is to bend a thin flexible acryl plate so as to create a curved surface where a user can touch. Controlling the degree of bending allows us to systematically alter the curvature of the surface. I realized the idea by attaching two haptic interfaces to both edges of the plate. The two haptic interfaces are position-controlled, resulting in the systematic control of the degree and direction of bending, and the location of the bended surface. That is, just simple position controlling of the two interfaces is enough for controlling both the pose and property of the end effector. This allows a unified control algorithm optimal for encountered-type haptic rendering scenario.

As an initial work, this paper focuses on demonstrating the feasibility of the idea. All the hardware parts needed for realizing the concept are implemented with commercial haptic interface, including the plate squeezing mechanism, plate behavior modeling, and modules for finger tracking (Sects. 3 and 4). As one of the core building blocks towards the long-term goal, software modules are also developed for curvature rendering, device controlling, and simulation for
the interaction with virtual objects (Sect. 5). The system is thoroughly evaluated physically (Sect. 6) and perceptually (Sect. 8) and is combined with visual display to show potentials of the concept (Sect. 7). As a proof-of-concept study, I admit that more efforts should be followed for the concept to be practically applied, in terms of workspace, achievable curvature, and variability of rendered shape, which are extensively discussed in Sect. 9.

2. Related Work

The concept of property-changing interface first drew attention to researchers in the field of human computer interface. Much effort has been made to develop a new flexible interaction device. Among them, most of the work has focused on shape, where the contour of an interface surface is systematically changed to provide flexible feeling of virtual object shape. Pin/rod-array based structure was the most common mechanism to realize it, e.g., [12]–[17]. An array of mechanically actuated pins or rods, sometimes covered by a flexible sheath, controls the contour of the overall surface that can be touched by a user. Besides the pin-array structure, Coelho et al. provided a comprehensive survey of materials that can be used in designing such interfaces[18]. However, effort to apply those shape-changing interfaces to an encountered-type haptic framework is rather scanty. Most shape changing interfaces are either too heavy or need bulky actuation modules, which prevents them from being used as an end-effector of an encountered-type haptic interface.

Shape, in general, is a variation of surface curvature. If the focus is narrowed down to a haptic device capable of displaying variable curvature, there exists a couple of efforts in the haptics research community. The main issue was how to merge a tactile perception of contact location changes due to varying curvature into a kinesthetic perception (absolute force and position perception). Roughly, two distinct approaches exist. One approach embedded a small rotating ball [19] or a moving flat plate [20], [21] under a fingerpad that is attached to the end effector of a large force-reflecting kinesthetic haptic interface. The pose of the small plate or ball is configurable, so that it can be instantaneously positioned and oriented to the contact point to bring it in contact with the fingertip (in [21] the finger is always in contact with the plate). Another approach is based on the encountered-type haptic principle. A large flat plate [22] or a deformable sheet [23] is mechanically attached to one or multiple robotic manipulator, and the pose of the plate is continuously configured to be along the tangential direction at the contact point with a virtual object. Both approaches effectively display contact location changes due to a curved surface. However, the former is mechanically and computationally complex and cannot display frictional feedback since the plate follows the finger. In addition, since a flat plate is used, both approaches were not able to create pressure distribution cue within the contact area, which is indeed an important tactile cue for curvature perception [4], [5].

3. Hardware Setup

Two parallel-mechanism impedance-type haptic interfaces (Novint Technologies Inc.; Falcon) were used as a prototype. The overall configuration is shown in Fig. 1 (a). Two Falcons are configured to face each other, holding two end edges of a thin acrylic plate. The original hand grip of the Falcon is replaced with a 2-DOF universal joint, giving rotational freedom along two axes to the plate as shown in the figure. The dimension of the plate is 120 × 120 mm with 0.7 mm thickness. The plate is thin enough to be bent with relatively small force but shows quasi-elastic behavior where it is completely flattened again after force removal.

Core idea of variable curvature stems from the flexible plate. The hardware setup enables us to exert squeezing force parallel to the plate, eventually bending the plate. The bent plate generates smooth curved surface of which the curvature can be systematically controlled by adjusting the amount of squeezing force. A user can actively touch the surface to feel variable curvature of virtual objects.

Bending a plate by compressing it from its sides needs squeezing force large enough to make the plate “buckle”. In general, when a slender beam is under compression from its sides, it stays flat until the compressive force reaches a certain value called “critical load.” Very little deflection is ob-

![Fig. 1 Hardware configuration for flexible curvature rendering.](image)
served in this stage. More compressive force makes the plate buckle, eventually bending the plate. Since my system relies on the behavior after buckling, the maximum squeezing force from the two Falcons should exceed the critical load. For this, I employ two design factors. First, I deployed the two Falcons facing each other where their exertable forces are at the maximum. Second, I designed the holding joints in a way that the squeezing force lays along a line being off from the axial line (a line passing through plate’s center of mass). This eccentric compression, in general, significantly decreases the critical load as compared to an axial compression. As a result, the squeezing force from the Falcons sufficiently surpasses the buckling, which also allows avoiding instability due to buckling. In addition, this off-axis design fixes the direction of bending; torque due to the eccentric load makes the plate always bend outward.

In order to make an encounter of the plate with a user’s finger, the position of the finger should be tracked. To this end, an infrared-marker based position tracker (Natural Point Inc.; TrackIR 5) is employed as shown in Fig. 1 (b). Three retroreflective markers are geometrically arranged and attached on the nail of the index finger. This structured marker and an external IR camera allows us to track the 3D position and 3D orientation of the fingertip in the IR camera coordinate system. The transformation between the two coordinate systems (the Falcon and the IR camera coordinate systems) are estimated based on the algorithm introduced in [24] and used for collocating the coordinate systems. According to the tracking performance experiment, the horizontal/vertical position resolution of tracking was 1 mm when marker-camera distance is 30 mm, and depth direction position resolution was 2 mm. The update rate was 120 Hz and the latency was measured as about 12 ms. In order to keep pace with faster sampling rate of haptic rendering (1 kHz), the tracking data are linearly extrapolated and up-sampled to 1 kHz. I should admit that marker-based tracking that needs an attachment to the finger is against the advantage of an encountered-type haptics. A lot of more advanced marker-less tracking techniques, e.g., the Kinect and the Leap Motion, can be also applied as alternative through which any attachment to the finger can be avoided.

4. Curvature Measurement and Modeling

Assuming that the plate is merely compressible, the curvature and overall geometry of the bent plate are directly related to the distance between the two holding joints. Revealing this relation ensures a precise control of the curvature. To this end, I measure plate’s geometry changes under various joint-to-joint distances and build a distance-geometry model. This model will be used for the rendering in Sect. 5.

For measurement of the geometry of a curved surface, I employ another stylus-type haptic interface (Sensible Technologies Inc.; PHANToM Omni). Its excellent position sensing capability allows us to accurately sample 3D points lying on the surface. Using this device, I sampled points on the surface under various joint-to-joint distances.

Starting from a completely flat configuration (120 mm), the distance decreased in 5 mm step until it reaches 50 mm. The two Falcons were position-controlled so as to set the distance to desired value. In every step, a curved line connecting two holding joints is stroked using the Omni, and 3D position samples lying on the line are gathered. The sampled points for each distance step are shown in Fig. 2.

Next step models the curvature distribution of the curves. To this end, a polynomial function is fitted to the curve for each distance step. According to the curve-fitting tests with 20 different forms of function, ninth-order polynomials showed the best result without too much over-fitting. A polynomial is also computationally simple enough to be used in a 1 kHz haptic loop. The local curvature $\kappa$ of the polynomials is derived using the definition of “first curvature” introduced in [25]. Finally, the relation between joint-to-joint distance and curvature distribution is modeled. Figure 3 shows the curvature at the center of the plate with respect to the distance. The curvature data are also fitted to 8th-order polynomial in order to fill the gaps between adjacent samples as shown by a dashed-line in the figure.

5. Rendering Curvature

As a proof-of-concept study, I only consider surfaces with cylindrical shape. In other words, a surface is bent only along one direction (maximum curvature direction) at a
To realize the encountered-type interaction, two different device control strategies should be used; one for actual rendering when the fingertip is in contact with the surface and the other for contact preparation during no collision state. The former, namely collision state, should provide solid perception of a curvature of a contacting virtual object, and thus the plate should be fixed in place during the user’s stroking. On the other hand, during no-collision state, the main focus of the system is to configure the plate ready for a future contact with the finger by continuously controlling the plate to be in appropriate location and curvature configuration based on the user’s finger position.

A rendering frame begins with detecting a collision between the fingertip and the virtual mesh. For this, the H-COLLIDE algorithm is utilized [26]. The H-COLLIDE algorithm efficiently detects the collision between a point and a mesh, and commonly used in many haptic rendering frameworks. It decomposes the workspace into uniform grids, constructing an oriented bounding box hierarchy of the grids, and efficiently find a contacting polygon and position based on frame-to-frame coherence. As a result, the algorithm declares a collision when the user’s finger position \( \mathbf{p}_s \) (computed by the IR tracking module) is inside the mesh contour. Otherwise, no-collision is declared. According to the result of the collision detection, the algorithm branches into two parts; no-collision state and collision state.

5.1 No-Collision State

In the no-collision state, the center of the plate or the target position (3D vector, \( \mathbf{p}_t \)) should continuously follow the most probable future contact point. Note that \( \mathbf{p}_t \) is a point on the mesh surface and never be the same as \( \mathbf{p}_s \) during the no-collision state. In my system, I assume that an optimal future contact point is the point closest to the fingertip. This assumption is valid since a user normally approaches the object along a normal direction of the surface. Thus, in every frame in the no-collision state, we need to find a point closest to \( \mathbf{p}_s \) among the points on the mesh. To this end, I utilize again the H-COLLIDE algorithm, which efficiently gives a closest point from a certain point outside a mesh. \( \mathbf{p}_s \) is updated to this proximity query result.

After determining the target position, \( \mathbf{p}_t \), the target curvature magnitude (scalar, \( \kappa \)) and target curvature direction (3D unit vector, \( \mathbf{u}_t \)) should be also determined. The H-COLLIDE algorithm also gives us a polygon that contains the closest point. Thus, based on \( \kappa \) and \( \mathbf{u}_t \), we can easily obtain \( \kappa \) and \( \mathbf{u}_t \) of \( \mathbf{p}_t \) using barycentric interpolation. In actual implementation, the proximity query in the H-COLLIDE algorithm is conducted on a part of mesh within a certain range (30 mm) in order to facilitate the query process.

Under a given target position \( \mathbf{p}_t \) and curvature direction \( \mathbf{u}_t \), the desired positions \( \mathbf{p}_1 \) and \( \mathbf{p}_2 \) can be derived by

\[
\mathbf{p}_1 = \mathbf{p}_t - \frac{d}{2} \mathbf{u}_t, \quad \mathbf{p}_2 = \mathbf{p}_t + \frac{d}{2} \mathbf{u}_t.
\]
where \( \mathbf{u}_i \) is a unit surface normal vector at the target position \( \mathbf{p}_i \), \( h \) is the deflection of \( \mathbf{p}_i \) from the initial flat configuration, and \( d \) is the distance from \( \mathbf{p}_1 \) to \( \mathbf{p}_2 \). The equations tell us that unknown variables, \( \mathbf{u}_i \), \( h \), and \( d \) should be estimated.

First step is to estimate \( d \). Under a given target curvature \( \kappa \), the inverse of the curve from Fig. 3 gives us the desired distance from \( \mathbf{p}_1 \) to \( \mathbf{p}_2 \). However, mathematical derivation of the inverse of an 8th-order polynomial is not easy. My approach is to up-sample the curve using the polynomials and to build a hash table with curvature values as keys and distances as data. The whole curvature range (from 0 mm\(^{-1}\) to 0.035 mm\(^{-1}\)) is evenly up-sampled with 0.0001 mm\(^{-1}\) step size, resulting in 350 bins. The output of the hash table is again linearly interpolated to fill the gap.

Next step is for \( \mathbf{u}_n \). It can be derived from \( \mathbf{u}_i \) and the line along the holding edge (\( l \) in the figure). Let a unit vector representing up-direction of \( l \) be \( \mathbf{u}_i \), \( \mathbf{u}_n \) can be derived by

\[
\mathbf{u}_n = \frac{\mathbf{u}_i \times \mathbf{u}_l}{|\mathbf{u}_i \times \mathbf{u}_l|}.
\]

(2)

\( \mathbf{u}_i \) can be derived as follows. The two universal joints holding the plate does not provide rotation of the plate about the \( \mathbf{u}_i \) direction. This results in \( \mathbf{u}_i \) always lying in a plane perpendicular to \( z \) axis (see the coordinate frame in the figure), resulting in the \( z \)-axis component of \( \mathbf{u}_i \) being zero. Among unlimited number of vectors with \( z \)-axis component being 0, \( \mathbf{u}_i \) can be determined using the following two characteristics of \( \mathbf{u}_i \); \( \mathbf{u}_i \) is perpendicular to \( \mathbf{u}_c \) and is a unit vector

\[
\mathbf{u}_i \cdot \mathbf{u}_c = 0, |\mathbf{u}_i| = 1.
\]

(3)

With Eqs. (3) and (2), \( \mathbf{u}_n \) can be calculated.

The final step is to derive the deflection, \( h \), of \( \mathbf{p}_i \) due to bending. For this, data based approach is adapted again. The surface geometry data in Fig. 2 already possess the information of the deflection for each distance value. Thus, for each curve, the deflection at the center is taken, and a distance-deflection curve is built as shown in Fig. 5. Then, the samples are again fitted to a 9th-order polynomial, which estimates \( h \) under a given distance \( d \) from the first step. All the unknown variables in Eq. (1) are determined and the desired positions, \( \mathbf{p}_1 \) and \( \mathbf{p}_2 \), can be derived using Eq. (1).

Finally, the two Falcons are position-controlled based on \( \mathbf{p}_1 \) and \( \mathbf{p}_2 \). The Falcon is originally designed as an impedance-type haptic device; sensing position and exerting force. To control the position, I employ a PID controller. In every step, the PID controller calculates desired force to make the device track the desired position based on feedback principle. The parameters of the PID controller are tuned using Zeigler-Nichols Method [27].

5.2 Collision State

The next part of the algorithm is for the collision state. Two interaction modes are considered. First mode assumes that the object is static, and the user intends to stroke to feel the contour of the object. This interaction scenario can be achieved by keeping the configuration of the plate constant, allowing the user to feel a solid contour. Thus, until the contact is released, \( \mathbf{p}_1 \) and \( \mathbf{p}_2 \) are not updated and pinned to the values calculated at the moment of contact. In this scenario, it is also assumed that the stroking occurs in relatively small area (about 20 mm) around the contact location. Even though the generated curvature is not uniform within this small area, I speculate that curvature change in such small region is perceptually negligible compared with low human curvature discriminability. I confirmed this using the collected curvature data. For each curve in Fig. 2, I cut the 20 mm region around the peak and found minimum and maximum curvature values within the region. Then, the difference between them was compared with the difference threshold of human curvature perception. The threshold was calculated based on the Weber fraction on curvature perception reported in [21] (= 6.2-8.5\%). For all curves, the differences were smaller than the difference threshold, which indicates that the curvature variation in 20 mm region is perceptually insignificant.

Second interaction mode aims at providing a grabbing interaction. The user can freely grab, lift, and move the virtual object with multiple fingers. This interaction is done by keeping the desired curvature constant, while letting the direction and position of curvature freely follow the user’s movement. This mode, in fact, does not provide accurate curvature to all fingers. The target curvature is correctly rendered only on a finger that is tracked. Nevertheless, this mode can demonstrate the potential of the approach and envision my future research direction.

In Sect. 7, the two interaction modes are examined with a simple virtual object.

6. Analysis of Workspace and Curvature Range

6.1 Achievable Workspace

This section analyzes achievable workspace in 3D Cartesian space. Martin et al. have quantified the workspace volume of a sole Falcon [28] based on its kinematics. I utilize this data to estimate the achievable workspace of the plate center as follows. First, the workspace of a sole Falcon in [28] is extended to my two-Falcon configuration (red volume in Fig. 6). Then, I densely and evenly sampled (every 2.8 mm)

![Fig. 5 Peak height with respect to the distance.](image-url)
points in this volume and constructed a set of 3D point pairs representing all possible combination of a point from one Falcone’s workspace to a point in the other. From this set, every point pair whose distance is larger than 120 mm were removed. This set now contains dense samples from every possible combination between \( p_1 \) and \( p_2 \).

For each element in the set, \( p_t \) can be calculated using Eqs. (1) and (2). This forms a set of all possible \( p_t \)'s, which approximates the achievable workspace. Green volume in Fig. 6 shows the enclosing 3D contour of \( p_t \). The volume is approximately 537,230 mm\(^3\).

### 6.2 Achievable Curvature Range

According to Fig. 2, we can achieve the maximum curvature 0.035 mm\(^{-1}\) at 50 mm distance and 0 mm\(^{-1}\) at the flat configuration. However, the full range of curvature cannot be rendered throughout the whole workspace. There are areas where fully flat or full bent configuration is not possible, especially near to the boundary of the workspace. In order to systematically analyze this, the workspace (green volume) was divided into dense voxels. Each voxel possesses one or more number of \( p_t \) sampled in Sect. 6.1 For each voxel, \( \kappa \) of \( p_t \) belonging to the voxel were calculated based on the corresponding point pair, and the maximum curvature value was taken among them, resulting in a voxelized workspace representing achievable curvature.

To visualize this, contour lines are extracted from the voxel model. The contour lines begin with 0.005 mm\(^{-1}\) curvature, increased up to 0.035 mm\(^{-1}\). Figure 7 shows the contour lines projected onto two axial planes. Note that full range of curvature can be rendered in most workspace, whereas limited coverage is observed at the boundary.

As a final note, achievable curvature generally depends on the bending characteristic of the plate, which is determined further by physical properties such as the width, stiffness, and thickness of the plate. For example, narrower plate may result in wider curvature range, but too small plate needs very fine position control, which may limit the accuracy of rendering. On the other hand, less stiff or thinner plate would yield wider range, but too flexible plate will not be able to withstand user’s pushing force, resulting in distorted curvature. Thus, the plate should be chosen appropriately considering the target application.

### 7. Application to Visuo-Haptic Setup and Physical Performance

The feasibility of the system was demonstrated by a visuo-haptic interaction scenario, where a user grasps and interacts with a virtual aluminum can while feeling its curvature. Visual rendering of the can is also provided through augmented reality (AR) hardware (video see-through head mounted display, Vuzix Corporation; WRAP1200 AR model), resulting in a collocated visuo-haptic scene. The head mounted display is tracked using an inertia tracker embedded in it, which is used for real-virtual registration in the visual scene. The user can stroke the aluminum can (upper row in Fig. 8, the first interaction mode described in Sect. 5), grab, and freely move it, e.g., leaning it for pouring water (lower row in Fig. 8, the second mode in Sect. 5).

The mesh model of the can consisted of 6562 polygons. I were able to keep 1 kHz update rate for haptic rendering, which demonstrates the efficiency of the algorithm. In the collision state, the rendering was stable until the pushing force of the stroking is over 3 N. Pushing force was measured using NANO17 miniaturized force sensor. After 3 N, the plate starts being significantly deformed, so that the curvature was distorted. Even over 3 N, no oscillation or in-
stability was observed. During no-collision state, the plate stably follows the user’s finger movement until the velocity of the movement exceeds 120 mm/s. Very fast movement (faster than 120 mm/s) resulted in unstable response of the device, which is mainly due to relatively slow update rate of position tracking data.

8. Perceptual Performance

Through a psychophysical study, I assessed the effectiveness of the system on conveying virtual object’s curvature information. Six participants (1 woman, average age of 26) conducted the experiments. Using the haptic setup introduced in Sect. 7, the participants were presented with a pair of cylindrical virtual objects having different curvatures, which were placed side by side. The task of the participants was to tell which object has higher curvature. Five different curvature values were used since this number of stimuli shows reasonable experimental resolution without significantly long experiment. They were 0.015, 0.017, 0.019, 0.021, and 0.023 mm⁻¹, and ten combinations of curvature pair were constructed from the 5 curvature values. Each combination was repeated five times, yielding total 50 trial pairs per a participant. After freely exploring the cylinders with the tracked finger (multifinger interaction was not allowed), participants answered either “Object 1 has higher curvature” or “Object 2 has higher curvature”. Presentation order of the curvature pair was randomized to minimize ordering effect. Before the main session, participants went through a training session where they were trained to explore the surface with moderate pushing force with one finger. One comparison took around 20 seconds, and the whole experiment took 20 minutes including training session.

Full 3D visual graphics of the cylinder was not provided in order to remove any visual cues for curvature judgment. Instead, only two center points of the virtual cylinders and the position of user’s hand were displayed in the HMD for the purpose of movement guidance. Background video stream displaying two haptic devices was also eliminated to ensure that the curvature is perceived only through haptic cues.

Figure 9 shows the correct discrimination rates for different comparisons. In order to evaluate the perceptual soundness of the system, these correct answer rates can be compared with difference threshold for human curvature perception. Weber fraction on curvature perception is roughly reported as 8.5% (e.g., in [21]). From this value, difference threshold at each curvature values can be calculated as shown in the figure. From the definition of difference threshold, if curvature difference of two arbitrary surfaces in real world exceeds this threshold, the difference may be perceivable to human.

This real world discrimination law is well reflected in the experimental result. For instance, in the comparison between 0.023 and 0.021 mm⁻¹ (see the fourth value of the last row in Fig. 9), the curvature difference is 0.002 mm⁻¹, which is nearly as same as the corresponding threshold value (0.001955 mm⁻¹). The correct answer ratio for that comparison was 0.51 (nearly random choice), which means that participants could not discriminate the two curvatures since the difference was too small (not much different from the threshold value). However, as curvature of object 1 decreases, threshold value decreases as well while the difference between adjacent curvature remain constant (0.001955 mm⁻¹), and thus the correct answer ratio increases (see the diagonal values). Comparisons of non-adjacent pairs (non-diagonal values) show nearly 90% correct ratio, which is expected since the curvature differences of these pairs far exceeds the corresponding threshold values. Overall, the discrimination characteristics of virtual curvatures generated by the system coincides well with those of real situation. This indirectly prove that my system can generate perceptually acceptable virtual curvature.

9. Limitations and Future Work

Although the present system showed competent physical and perceptual performance, there still exist many limitations that require further improvements and investigation in order for the present system to be practically employed in a virtual reality framework.

First, the current rendering algorithm only provides a limited range of shape (cylindrical objects) and limited interaction; 1. stroking a static object along a fixed direction within a small patch, and 2. grasping and lifting an object without stroking. These limitations can be effectively resolved by introducing more sophisticated rendering algo-

![Fig. 8](image1.png)

Scenes captured from the haptic AR system. The virtual can is rendered transparently due to virtual-real objects occlusion.

![Fig. 9](image2.png)

Table 1: Perceptile of the correct answer “Cylinder 1 has higher curvature”.

| Curvature of cylinder 1 (mm⁻¹) | Curvature of cylinder 2 (mm⁻¹) | Correct answer ratio | Difference threshold (mm⁻¹) |
|--------------------------------|--------------------------------|----------------------|-----------------------------|
| 0.015                          | 0.017                          | 0.67                 | 0.001275                    |
| 0.017                          | 0.019                          | 0.69                 | 0.001445                    |
| 0.019                          | 0.021                          | 0.63                 | 0.001615                    |
| 0.021                          | 0.023                          | 0.59                 | 0.001785                    |
| 0.023                          | 0.025                          | 0.51                 | 0.001955                    |
rithm. Since the user’s finger (or fingers) is tracked in real-time, it is possible to track and even predict the direction of user’s stroking for very short period. The system can utilize these information together with the virtual object’s geometry around the contact point for continuously adjusting the position and orientation of the rendering patch in real-time in a way that the patch is smoothly coincide with the orientation of the surface at the contact point. In addition, using the predicted direction of user’s stroke, the plate’s bending direction, i.e., the direction of maximum curvature, can be also continuously controlled in a way that the curvature along the user’s stroking direction smoothly corresponds to the target curvature (curvature of a surface directly under the contact point). Nevertheless, current implementation of the rendering algorithm does not provide such functionalities, which will be my tangible future work.

Second, the present system only considers rigid objects: the simulation of contact dynamics due to stiffness of the objects is not included. Nevertheless, the holding devices (Novint Falcons) in the system are basically impedance-type haptic interfaces, which are designed to render stiffness, and potentially, they can be also easily used for stiffness rendering. Merging conventional stiffness simulation algorithms into the plate controlling algorithm will not be a challenging task, which will be also out next target for this project.

Third, the current hardware implementation has quite limited workspace, range of plate direction, and achievable curvature. As analyzed in Sect. 6.1, this is mainly due to the small workspace and parallel-linkage nature of the Flacon devices. Since the core part of the algorithm was tried to be as general as possible, the algorithm is mostly independent from the hardware setup and can be easily adapted to larger, more sophisticated robot arms setup, preferably serial linkage type that usually has larger degree of freedom/workspace and less interferences between the arms. Ideally, if the holding arms are large enough and the interference between the devices can be minimized, the workspace as well as the range of direction and achievable curvature can be significantly extended. Currently, this extension is on-going—two PHANToM Premium haptic interfaces replaced the Falcons—and so far the result was promising.

When the improved system is ready, I plan to apply it to touch-enabled applications that need the bare-hand rendering of virtual object’s shape and stiffness. One of the first target will be a virtual prototyping or virtual show-window system where a digital copy of a product or an environment, e.g., a mobile phone, a computer mouse, and a control panel in a driver’s seat, is haptically and visually stored and displayed through the system. Users can remotely inspect the contour and shape of the digital copy by touching and seeing, in order to assess the affectability and usability of it. The system can be also applied to a tele-operation scenario, where a user controls a remotely located robot (equipped with a 3D camera and probing tool) so as to explore remote environment. The user gets feedback on what the robot touches through a bare-hand and contour-enabled manner, e.g., inspecting the bent hull of a shipwreck under water and contour-following the face of a friend living far.

10. Conclusion

This paper presents an effective method for haptically rendering variable curvature using a flexible plate. Complete modules including hardware as well as modeling, control, and rendering algorithms are implemented. The performance including the achievable workspace and curvature range are analyzed, and the limitations of the system are also discussed. In addition, the system is combined with a visual AR, demonstrating the potential of the system. Finally, perceptual performance of the system was assessed through a psychophysical experiment.

This implementation is one of my building blocks towards the “Universal End effector” for the encountered-type haptic concept. Many short-term future research plans are also discussed in the paper. Beside the bending plate method, I will also approach the goal using different technologies, e.g., manufacturing the end effector with smart material such as MR fluid to systematically control the shape and viscosity of the handle.

Acknowledgments

This work was supported by a grant from the Kyung Hee University in 2012 (KHU-20121638), by the Basic Science Research Program through the NRF of Korea (NRF-2014R1A1A057100), by the ITRC program through IITP of Korea (IITP-2016-H8501-16-1015), and by ERC program through NRF of Korea (2011-0030075).

References

[1] K. Salisbury, F. Conti, and F. Barbagli, “Haptic rendering: Introductory concepts,” IEEE Computer Graphics and Applications, vol.24, pp.24–32, 2004.
[2] S.J. Lederman and R.L. Klatzky, “Extracting object properties through haptic exploration,” Acta psychologica, vol.84, no.1, pp.29–40, 1993.
[3] L.A. Jones and S.J. Lederman, Human Hand Function, ch. Active Haptic Sensing, pp.75–99, Oxford University Press, 2006.
[4] L.A. Jones and S.J. Lederman, Human Hand Function, ch. Tactile Sensing, pp.44–74, Oxford University Press, 2006.
[5] V. Hayward, Human Haptic Perception: Basics and Applications, ch. Haptic shape cues, invariants, priors and interface design, pp.381–392, Birkhäuser Basel, Basel, 2008.
[6] W.A. McNeely, “Robotic graphics: A new approach to force feedback for virtual reality,” Proceedings of Virtual Reality Annual International Symposium, pp.336–341, 1993.
[7] Y. Yokokohji, R.L. Hollis, and T. Kanade, “WYSIWYF display: A visual/haptic interface to virtual environment,” Presence: Teleoperators and Virtual Environment, vol.8, no.4, pp.412–434, 1999.
[8] G. Ye, J.J. Corso, G.D. Hager, and A.M. Okamura, “VisHap: Augmented reality combining haptics and vision,” Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, pp.3425–3431, 2003.
[9] E. Ruffaldi, P. Tripicchio, C.A. Avizzano, and M. Bergamasco, “Haptic rendering of juggling with encountered type interfaces,” Presence: Teleoperators and Virtual Environment, vol.20, no.5,
[10] Y. Yokokohji, N. Muramori, Y. Sato, and T. Yoshikawa, “Designing an encountered-type haptic display for multiple fingertip contacts based on the observation of human grasping behaviors,” The International Journal of Robotics Research, vol.24, no.9, pp.717–729, 2005.

[11] S. Nakagawara, H. Kajimoto, N. Kawakami, S. Tachi, and I. Kawabuchi, “An encounter-type multi-fingered master hand using circuitous joints,” Proceedings of IEEE International Conference on Robotics and Automation, pp.2667–2672, 2005.

[12] M. Anabuki and H. Ishii, “Ar-Jig: A handheld tangible user interface for modification of 3D digital form via 2D physical curve,” Proceedings of IEEE and ACM International Symposium on Mixed and Augmented Reality, pp.55–66, 2007.

[13] H. Zhu and W.J. Book, “Practical structure design and control for digital clay,” Proceedings of ASME International Mechanical Engineering Congress and Exhibition, pp.1051–1058, 2004.

[14] M. Anabuki and H. Ishii, “Ar-Jig: A handheld tangible user interface for modification of 3D digital form via 2D physical curve,” Proceedings of IEEE International Conference on Robotics and Automation, pp.2667–2672, 2005.

[15] M. Nakatani, H. Kajimoto, D. Sekiguchi, N. Kawakami, and S. Tachi, “3D form display with shape memory alloy,” Proceedings of Virtual Reality Society of Japan Annual Conference, pp.247–248, 2003.

[16] M. Blackshaw, A. DeVincenzi, D. Lakatos, D. Leithinger, and H. Ishii, “Recompose: Direct and gestural interaction with an actuated surface,” Proceedings of ACM CHI Extended Abstract on Human Factors in Computing Systems, pp.1237–1242, 2011.

[17] D. Leithinger and H. Ishii, “Relief: A scalable actuated shape display,” Proceedings of International Conference on Tangible, Embedded, and Embodied Interaction, pp.221–222, 2010.

[18] H. Iwata, H. Yano, F. Nakaizumi, and R. Kawamura, “Project FEELEX: Adding haptic surface to graphics,” Proceedings of Annual Conference on Computer Graphics and Interactive Techniques, pp.469–476, 2001.

[19] M. Coelho and J. Zigelbaum, “Shape-changing interfaces,” Personal and Ubiquitous Computing, vol.15, no.2, pp.161–173, 2011.

[20] W.R. Provancher, M.R. Cutkosky, K.J. Kuchenbecker, and G. Niemeyer, “Contact location display for haptic perception of curvature and object motion,” The International Journal of Robotics Research, vol.24, no.9, pp.691–702, 2005.

[21] M.W.A. Wijntjes, A. Sato, V. Hayward, and A.M.L. Kappers, “Local surface orientation dominates haptic curvature discrimination,” IEEE Transactions on Haptics, vol.2, no.2, pp.94–102, 2009.

[22] T. Zeng, B. Lemaire-Semail, F. Giraud, and M. Amberg, “Contribution of slip cue to curvature perception through active and dynamic touch,” IEEE Transactions on Haptics, vol.6, no.4, pp.408–416, 2013.

[23] T. Furukawa, K. Inoue, T. Takubo, and T. Arai, “Encountered-type visual haptic display using flexible sheet,” Proceedings of IEEE International Conference on Robotics and Automation, pp.479–484, 2007.

[24] K.S. Arun, T.S. Huang, and S.D. Blostein, “Least-Squares Fitting of Two 3-D Point Sets,” Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol.9, no.5, pp.698–700, 1987.

[25] E. Kreyszig, Differential Geometry, ch. Principal Normal, Curvature, Osculating Circle, pp.34–36, DOVER PUBLN Incorporated, 1991.

[26] A. Gregory, M.C. Lin, S. Gottschalk, and R. Taylor, “A framework for fast and accurate collision detection for haptic interaction,” Proceedings of Virtual Reality Conference, pp.38–45, 1999.

[27] J.G. Zeigler and N.B. Nichols, “Optimum settings for automatic controllers,” Journal of Dynamic Systems, Measurement, and Control, vol.115, no.2B, pp.220–222, 1993.

[28] S. Martin and N. Hillier, “Characterisation of the novint falcon haptic device for application as a robot manipulator,” Proceedings of Australasian Conference on Robotics and Automation, pp.1–9, 2009.

Seokhee Jeon received the BS and PhD degrees in computer science and engineering at the Pohang University of Science and Technology (POSTECH) in 2003 and 2010, respectively. From 2010 to 2012, he was a postdoctoral researcher in the Computer Vision Laboratory at ETH Zurich. In 2012, he joined as an assistant professor in the Department of Computer Engineering, Kyung Hee University, where he is currently directing the Haptics Laboratory. His research focuses on haptic rendering in an augmented reality environment, applications of haptics technology in medical training, and usability of augmented reality applications.