CHARM: Cord-Based Haptic Augmented Reality Manipulation

Konstantin Klamka(✉), Patrick Reipschläger, and Raimund Dachselt

Interactive Media Lab Dresden, Technische Universität Dresden,
Nöthnitzer Str. 46, 01187 Dresden, Germany
{konstantin.klamka,patrick.reipschlaeger,raimund.dachselt}@tu-dresden.de

Fig. 1. Our *CHARM* system is a combination of a versatile retractable input device for radial AR menus **(A)** and 3D object manipulation **(B)**, that is fully implemented for state-of-the-art AR glasses. The smart handle **(C)** provides additional controls.

Abstract. The recent trend of emerging high-quality Augmented Reality (AR) glasses offered the possibility for visually exciting application scenarios. However, the interaction with these devices is often challenging since current input methods most of the time lack haptic feedback and are limited in their user interface controls. With this work, we introduce *CHARM*, a combination of a belt-worn interaction device, utilizing a retractable cord, and a set of interaction techniques to enhance AR input capabilities with physical controls and spatial constraints. Building on our previous research, we created a fully-functional prototype to investigate how body-worn string devices can be used to support generic AR tasks. We contribute a radial widget menu for system control as well as transformation techniques for 3D object manipulation. To validate our interaction concepts for system control, we implemented a mid-air gesture interface as a baseline and evaluated our prototype in two formative user studies. Our results show that our approach provides flexibility regarding possible interaction mappings and was preferred for manipulation tasks compared to mid-air gesture input.

Keywords: Augmented reality · Haptic feedback · Elastic input · Cord input · Radial menu · 3D interaction · 3D transformation · Wearable computing

J. Y. C. Chen and G. Fragomeni (Eds.): HCII 2019, LNCS 11574, pp. 96–114, 2019.
https://doi.org/10.1007/978-3-030-21607-8_8

1 Introduction

The dissemination of high-quality head-mounted displays with augmented reality (AR) capabilities (e.g., Microsoft HoloLens) served as the foundation for the development of new AR applications in various fields. However, while the opportunities are clearly inspiring, well-known problems in interacting with these AR applications still prevail. Those issues are, e.g., interface limitations [1] and missing tactile feedback of physical surfaces [12] and references (e.g., a desk or display). This often makes it difficult and physically demanding to select or manipulate virtual objects using hand gestures. Current system control solutions often lack the support of even simple control tasks. To address the lack of haptics in AR and VR, several approaches have been proposed. Emerging technologies, including smart textiles (e.g., [8,21]), tiny wearable devices (e.g., [25]) or specialized AR devices (e.g., [9]) have been introduced for head-mounted displays and provide different forms of tactile feedback. In addition, *string-based systems* in stationary and cave-like environments connect fingers, wrists, tangible grips or even the hole body with retractable strings in a fixed interaction frame to enable force or torque feedback. However, the problem remains that these approaches often do not fulfill important needs of personal AR interaction such as mobility, eyes-free interaction, and tactile controls.

In our work, we aim to provide an unintrusive mobile controller that enables an easy and sensory-rich on-demand access to AR system control and 3D transformation tasks. In particular, we want to support interaction with AR applications by providing a frame of spatial reference. Therefore, we present *CHARM*, *C*ord-based *H*aptic *A*ugmented *R*eality *M*anipulation[1]. We see high potential in using retractable body-worn string controllers, building on mechanical wind-up mechanisms that are able to change the string length through pulling and thereby provide continuous haptic feedback. Thus, we build on our previous work Elasticcon [14], investigating its application to AR scenarios. As system control is an important aspect of AR applications, we devised an AR menu and widgets controlled by our *CHARM* device that allow to change states, modes or values. In addition, we also support *object manipulation*, which is central to most AR applications [4], by providing interaction techniques for 3D transformation.
The contribution of our work is composed as follows:

- **An elastic input device** consisting of a belt-worn retractable multi Degree-of-Freedom (DoF) handle which provides a rich,cone-shaped interaction space and can be natively connected to the HoloLens.
- **A menu and interaction solution** that provides a flexible, radial widget for AR system control tasks and is controlled by our *CHARM* device.
- **Interaction techniques** for **3D object manipulation** including translation, rotation and uniform scaling of 3D content in AR environments.
- **A fully-functional software prototype** implementation of our menu and 3D transformation techniques for the Microsoft HoloLens.

---

[1] See our project website for additional information: http://www.imld.de/charm.

- **A formative, qualitative user study** investigating our menu solution regarding different interaction mapping schemes and comparing it to a baseline mid-air gesture interface.

The paper is structured along these contributions: First, we summarize and discuss previous work and thereby position our own approach. Then, we introduce the concept and realization of our mobile elastic controller for AR interaction. As a next step, we present our radial widget control concept and report about its formative evaluation. Finally, we propose a 3D transformation concept for object manipulation and conclude with a discussion and future work.

2 Background and Related Work

The related work for our approach is twofold, and we structured it into the two sections: *Interactive Controls and Menus in AR Environments* as well as *Body-worn Cord Controllers.*

2.1 Interactive Controls and Menus in AR Environments

A vast variety of interaction techniques was developed in the field of VR and AR (see [4] for an overview, [6] for menus in particular). For example, *hands- & glove-based* approaches have been used to attach menus to the user's hand and link items or interactions to different fingers [3,19]. Tinmith-Gloves [19] can be used to browse a display-referenced top menu and specify 3D input based on contacting fingers gestures. In contrast, TULIP [3] was designed to access three menu items at a time while using the fourth finger to switch to a new set. In addition, *mid-air interactions* focus on floating gestures in front of the user. For instance, Microsoft's HoloLens combines air tap gestures with a gaze cursor to confirm selections. Furthermore, *physical handheld surfaces* have been investigated to provide graspable 2D interaction surfaces that enable a familiar frame of reference for 3D interaction menus and tasks [5,13,23,26]. Szalavári introduced a two-handed Personal Interaction Panel [23] which enables pen interaction on a handheld tablet transferring the pen-and-tablet paradigm to AR menus. Further, Shake Menus [26], a menu displayed around a tracked cardboard, applies the metaphor of shaking a wrapped gift to explores menu options and thereby focus on more tangible interactions. Hyeongmook et al. [13] used a mobile phone as an interactive surface panel. *Physical controllers* provide advanced capabilities [10,17]. Gebhardt et al. [10] investigate pick ray, hand projection and hand rotation interaction techniques for extended pie menus with commercial fly sticks. Instead of pointing, Lee and Woo [17] developed a tangible spin cube for a 3D ring menu in space. In addition, the Cubic Mouse [9] and YoYo Device [22] are VR interaction devices that enable seamless 3D navigation and the application of cutting planes.

These solutions do not meet all requirements for an unobtrusive mobile AR controller as they are either not eyes-free, provide little to no tactile feedback or require large setups. In order to overcome these limitations and to provide an always-available system, we specifically focus on the promising class of body-worn retractable string controllers that have elastic and haptic properties.

2.2 Body-Worn Cord Controllers

A number of wearable and retractable cord controllers have been proposed in the literature. Some controllers build on mechanical wind-up mechanisms that are able to change the string length through pulling and thereby provide continuous haptic feedback [2,14,16,20,24]. Furthermore, cord controllers have been proposed for different positions of the body and accessories including the chest [16], wrist [2], finger [24] and belt [14,20] as well as at head-phone cables [18,21] and hoodies [15,18,21]. In addition, several degrees of freedom including the strings' traction, deflection, manipulation, and additional knobs as well as displays at the strings' end have been proposed for carrying out simple selection and navigation tasks. For example, Blasko et al. [2] presented a small wrist-worn dual-display that uses the string's length and angular deflection to provide access to a set of angular cells, while Koch and Witt [16] control a basic $3 \times 3 \times 3$ selection grid capturing the position of a chest-worn string in a cone-shaped interaction space. Pohl et al. [20] combined a retractable belt-worn system with a display badge to support indoor navigation. Furthermore, Schwarz et al. [21] propose an touch-enabled hoodie zipper, called CordInput. ARCord [15] extends the interactive hoodie cords with holographic visual overlays, while I/O Braid [18] enables visual feedback based on weaved optical fibers.

In our previous work Elasticcon [14], we introduced a design space for body-worn retractable controllers and proposed a generic belt-worn string controller with a set of exchangeable traction knobs focusing on mappings for essential interaction tasks. Although we previously already argued that body-worn retractable string controllers have a promising potential for wearable AR glasses, we conducted no detailed investigation of this scenario. In this work, we propose using a body-worn retractable string controller for *mobile AR system control and direct 3D manipulation*, which we will describe in more detail in the next sections.

3 The CHARM Input Device

First, we want to introduce our wearable *CHARM* input device (see Fig. 2) that we developed to address the lack of haptics in AR interaction by providing physical constraints. Based on our prior work Elasticcon [14], our system consists of a string-based control handle that can be smoothly pulled away from the body (see Fig. 2A) or deflected in mid-air (Fig. 2B) and thereby enables several body-relative degrees of freedom (DoF). In addition, a tangible handle (Fig. 2C) at the end of the string provides a thumb-joystick, three push- and one trigger-button, and vibro-tactile feedback. All DoFs work in synergy and create a cone-shaped interaction space (Fig. 2D).

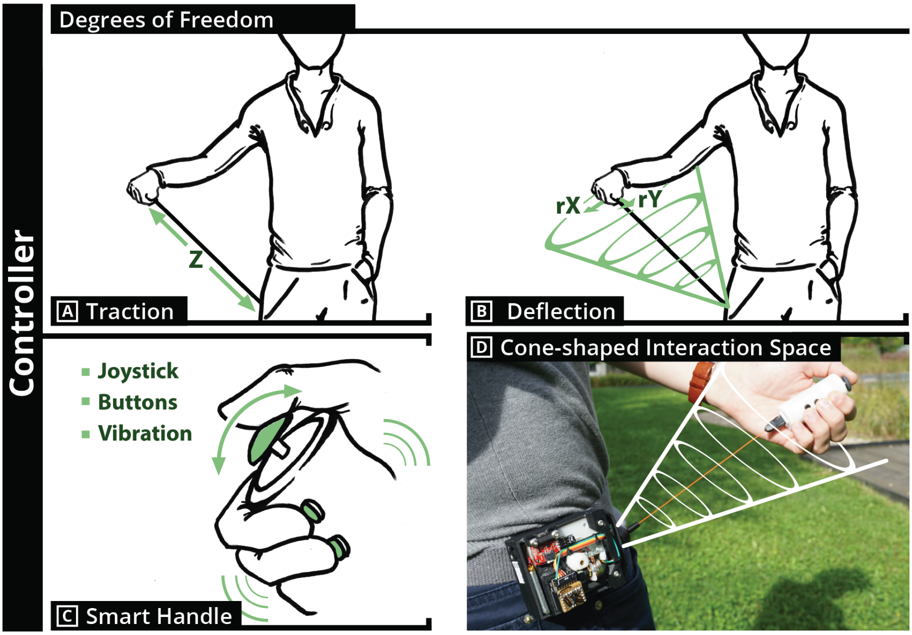

Fig. 2. Our *CHARM* controller integrates several DoF **(A–C)** including the dimensions of traction **(A)**, deflection **(B)** and a multi-DoF handle control **(C)** that creates a rich cone-shaped interaction space **(D)**.

3.1 Belt-Worn *CHARM* Controller

To detect the traction length (Z) and radial deflection (rX, rY) of the cord, our system needed the integration of a retractable winding and deflection mechanism as well as related sensing, processing, power and transmission components. The retractable winding mechanism (see Fig. 3A) was taken from a disassembled GameTrak[2] controller. A worm drive translates the axis of the spring-loaded spool to a potentiometer measuring the current traction length. The deflection of the corresponding pulling direction is tracked by a two-axis joint from a regular thumb-joystick using linear potentiometers. Our main logic board primarily consists of a Semiconductor nRF51822 micro-controller with built-in Bluetooth Low Energy (BLE) capabilities. We used custom BLE peripherals and implemented three sensor characteristics (traction: Z and deflection: rX, rY) based on the Generic Attribute Profile (GATT) to provide versatile wireless connectivity. In contrast to prior research, this allows us to *natively connect* our prototype to the HoloLens without other computers or mobile phones as intermediate devices. A power switch and LED were integrated in the 3D-printed casing. Small mounting rigs make it easy to clip the prototype to the belt for either right- or left-handed use. The device is powered by a 3.7 V lipo battery with 900 mAh and can be charged via an external Micro-USB connector.

---

[2] See http://en.wikipedia.org/wiki/Gametrak.

Fig. 3. Hardware prototype of our *CHARM*-Controller **(A)**, showing the first **(B)** and second, ergonomic **(C)** iteration of our handle, as well as the hardware inside **(D)**.

3.2 *CHARM* Handle

The *CHARM* handle went through several iterations. While at first we focused on a small and unobtrusive design (see Fig. 3B), the studies we conducted have shown that users prefer a more ergonomic handle. We addressed this issue with a refined version of the handle (see Fig. 3C). The final device uses a nRF51822 micro-controller, that handles the input of a two-axis thumb-joystick with a center button, a frontal trigger-button, two tactile push-buttons positioned left and right of the trigger, and also provides pulse-width modulated vibro-tactile feedback. The case of the *CHARM* handle consists of a custom designed, 3D-printed left and right part that hold the electronics and are held together by screws, which enables an easy access to the hardware (see Fig. 3D). All sensor values are represented in GATT characteristics and can be – depending on their type – subscribed or written via BLE.

3.3 Software Prototype

We implemented our *CHARM* Prototype for the Microsoft HoloLens[3] (as a representative of state-of-the-art AR glasses) using the Unity 3D game engine. To achieve our goal of a generic menu solution, we also designed on our software architecture to provide high flexibility and extensibility, using a modular structure and completely encapsulated the interaction functionality to offer easy support for different interaction modalities. The composition of our menus is defined by an accompanying XML-description. For 3D transformation we implemented a separate *CHARM* controlled and gesture controlled transformation widget, which can be attached to arbitrary objects and can be easily integrated into existing applications.

---

[3] See https://www.microsoft.com/en-us/hololens.

4 Our CHARM Menu-Design

One important task for most applications is system control, which we address with a configurable radial AR menu, comparable to the works of, e.g., Davis et al. [7]. Therefore one of our goals was to use our *CHARM* device (as described in the previous chapter) for controlling generic widgets, that can be easily adapted to the requirements of arbitrary applications. In the following, we introduce the design of our AR menu and describe the control scheme using our *CHARM* device and how its several DoFs are utilized (see Fig. 4A+B).

4.1 General Menu Design

To take advantage of the cone-shaped interaction space of our input-device (see Fig. 2D), we decided to use a planar, hierarchical radial menu. The menu can either be situated at a specific real world position for controlling aspects of real or virtual objects, or in case of general menus, be situated in front of the user, following her. Since the menu is planar, it always faces the user to prevent any visibility issues. The segments of the menu are always distributed equally to form a full circle (see Fig. 4A). Although our menu basically supports any number of items, we have limited the maximum number of items in our prototype to eight in order to ensure good visibility and interaction with each individual menu item. The menu is hierarchical, so that a menu item can activate a sub menu with different items, thus enabling menus of arbitrary depth and complexity (see Fig. 4A). An additional element in the menus center displays the current menu level and also acts as a trigger to return to the previous level.

4.2 Design of Menu Sections and Widget Controls

Since our goal is to provide a generic menu solution, we integrated common control widgets that are devised based on established graphical user interfaces. Each menu segment consists of a description label, an icon, and widget-specific elements like, e.g., the selected element of a list (e.g., see Fig. 4C). In the following, we describe each of the individual widget types:

Buttons are probably the most basic, but also most important type of widget for any menu. In our solution they can activate a specific action, like switching to a sub-menu or trigger an application-specific function (see Fig. 4A). Additionally, we also provide toggle buttons that can be switched either on or off and are therefore suitable for controlling boolean operations within an application, like showing or hiding specific objects (Fig. 4B).

Sliders can be used to adjust a value within a specific range. Our slider can be configured to handle continuous values, as well as discrete ones. The allowed minimum and maximum values are configurable as well, and a suffix can be defined to allow for the representation of specific units. Sliders show their current value directly in the menu segment when inactive (see Figs. 1B and Fig. 4C). When a slider is activated, a scale is shown above or beside the menu segment,

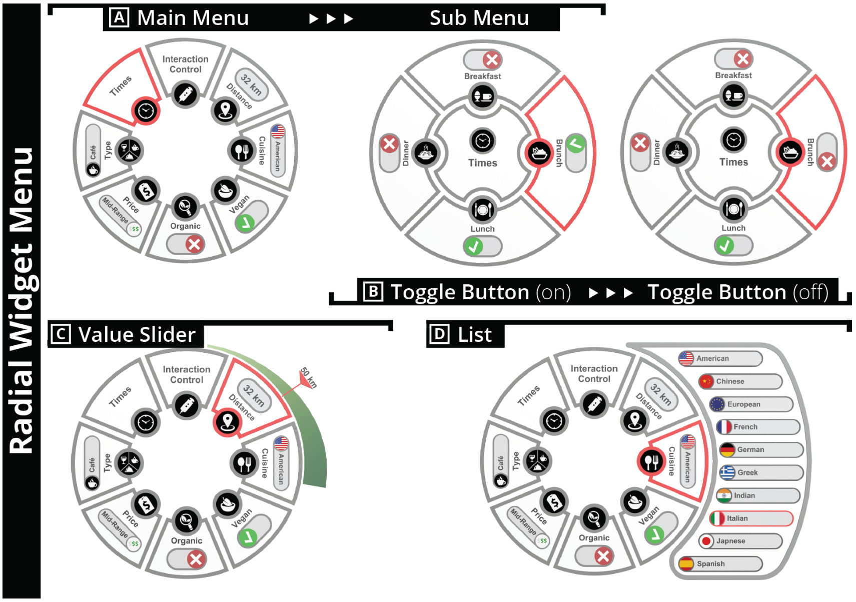

Fig. 4. Our radial AR menu consists of hierarchical sub-menus **(A)** and widget controls including toggle buttons **(B)**, sliders **(C)** and lists **(D)**.

which is rotated so that the original value is centered at the middle of the menu segment. This makes it easy for a user to determine if the selected value is higher or lower than the original one. During an adjustment a small red moving arrow and a respective label preview the currently selected value. In addition, the slider also supports range selections which are visualized with two arrow handles of both ends of the value range and a semi-transparent mask between them.

Lists support the selection of items from a larger data set. Each item consist of a description and a corresponding icon. Similar to the previous widgets, lists show their current selection directly in the menu segment. When a list widget is activated, a side menu is shown to the left or right, depending on the segments positions (see Figs. 1A and Fig. 4D). It shows the available items, centered on the currently selected one. To ensure optimal readability, only ten items are displayed at the same time and the list can be scrolled vertically as necessary, with a scrollbar indicating the current position.

4.3 Interaction Design for Menu Control

This section describes how the menu can be controlled with our elastic input device, *CHARM*. Simply pulling the *CHARM* handle makes the general menu appear at a fixed distance relative to the user. Object-specific menus are situated at the object's position and can be accesses by pressing the trigger-button. All menus always face the user.

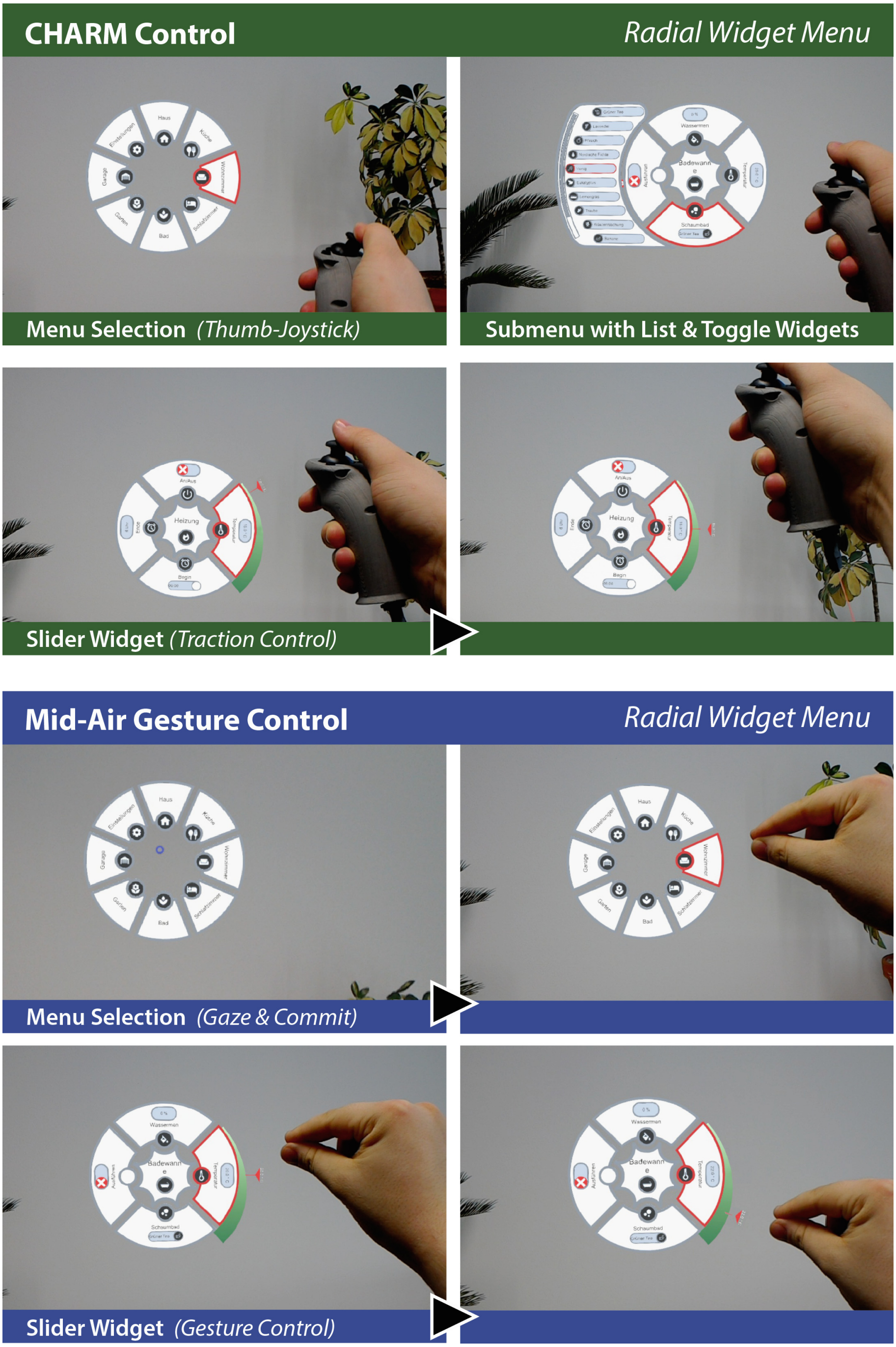

Fig. 5. Our Radial Widget Menu using our *CHARM* control and mid-air gestures.

Segment Navigation: Because of the rich input space offered by our device, the best interaction mapping for navigating the radial menu is not immediately obvious. Therefore, we propose three alternative *CHARM* interaction mappings (**C1-C3**) for the navigation: The *pulling-based browsing* mapping (**C1**) uses the string's pulling length (see Fig. 2A) to select a menu segment. When the user slightly pulls the handle and string, the menu segments are selected in clockwise order, depending on the current pulling length and symbolized by a red outline. The *deflection-based selection* (**C2**) maps the angular deflection of the string to a menu segment (see Fig. 2B). For instance, moving the handle to the left highlights the left segment in the radial AR menu. The *thumb-joystick* mapping (**C3**) uses the deflection of the thumb-joystick (see Fig. 2C) to target a specific menu segment. If the user moves the joystick to the right, the right segment in the menu is selected. While the deflection-based techniques map the finger (**C3**) or arm (**C2**) direction to a radial segment relying on two spatial dimensions, the pulling-based selection (**C1**) provides access to the radial segments by only using the one dimension of traction. Every selection change is supported by vibro-tactile feedback of the handle, which can also be deactivated if the user prefers.

Segment and Widget Control: Pressing the handle's trigger-button activates a menu-segment. Depending on the selected segment type (e.g., hierarchical submenu or a specific widget control), a corresponding action is executed: Changing the menu level replaces the current menu items with the ones of the corresponding sub-menu and alters the middle element accordingly. Activating a list or slider shows either the side menu with the list items or the slider scale. Since lists and sliders have only one dimension, they are manipulated by pulling the handle away from or to the body, as this offers the most haptic feedback to users. We integrate a relative mapping, using the current pulling length of the string as a starting position for sliders and lists. This makes it very easy for users to increase or decrease the current value or selection and also prevents the list or slider from immediately jumping to a new value when the section is activated as it would be the case with an absolute mapping. If a specific value is unreachable, e.g., due to physical constraints, our system provides a clutching method that allows users to hold down the trigger-button, return to a comfortable position, and release the button to continue. Lists automatically scroll when the second or second to last item is selected. In case of range sliders, the left push-button of the *CHARM* handle is used to switch between both values, which are then adjusted by pulling the handle.

5 User Feedback

To evaluate our menu concept for system control and prototype implementation, we conducted two small-scale qualitative user studies for hands-on feedback and insights. In the first study we were particularly interested in finding the most suitable interaction mapping and get general feedback to our system design. The second study improved upon our design based on the results of the first study and was focused on comparing our *CHARM* interaction to gesture interaction using the native air tap of the Microsoft HoloLens as a baseline. Both studies were conducted using our first generation handle (see Fig. 3B).

5.1 First Formative Study

The subjects of this study had to explore our menu (a restaurant finder use-case) using our *CHARM* device and perform simple menu tasks, that included all widgets. We recruited 6 participants (3 female, 3 male) between the age of 24 and 49 from students and post-doctoral personnel of our local university. Participants reported some experience with mixed reality and all except one participant had used some form of radial menu before.

Tasks and Procedure: Each participant started with a short training session to try out the menu and familiarize themselves with the interaction mappings. After that, we evaluated the three mapping conditions introduced in Sect. 4.3 (**C2**: *pulling-based*, **C2**: *string-deflection* and **C3**: *thumb-joystick*) in a counterbalanced within-subject design. Participants had to solve a continuous sequence of eight tasks for each condition, which varied between each condition to avoid learning effects. The tasks incorporated interacting with all our proposed widgets, as well as sub-menus, as we asked participants to find restaurants matching certain criteria (e.g., cuisine, price, distance, type). Participants had to accomplish these tasks without help from the experimenter and were encouraged to describe and comment on all their actions. Every interaction condition (**C1**-**C3**) could only be controlled with the current mapping (e.g., only thumb-joystick or deflection) with no combination of mappings. The overall duration of the study for each participant was approximately 45 min.

Measurements: We recorded the video stream from the HoloLens and a video of each participant from far away. This enabled us not only to see the user's perspective but also to reconstruct their corresponding interaction with our *CHARM* device. Of the two present investigators, one was primarily responsible for conducting the study, while the other took detailed notes of the observations of the participant. All sessions were accompanied by questionnaires after each condition that included a raw NASA-TLX [11] with seven-point scales and three open questions (general pros, cons, and comments) to get qualitative feedback.

Results: In general, our user feedback revealed that our approach has been assessed as useful and suitable for the control of AR menus. Surprisingly, no interaction mapping proved clearly superior over the others, but was instead subject to user preference. Two participants preferred the *pulling-based segment navigation* (**C1**), three participants the *string deflection* (**C2**), and one participant the *joystick-based navigation* (**C3**). Although **C3** was only rated once as the preferred interaction mapping, nearly all participants rated it as their second favorite, stating that they liked the condition in general, but did not like moving and pressing the joystick at the same time. Therefore, we have the strong assumption that this mapping would perform significantly better when the trigger button of our new handle design would be used instead of the joystick button.

Based on the feedback we also improved the mappings of all interaction styles, e.g., how much a slider value changes when the handle is pulled, and adjust a deadzone to the deflection. All participants mentioned that the tactile buttons are very helpful for interacting with Mixed Reality and much better than only getting visual feedback. Our results from the NASA-TLX test showed no conspicuous differences in the task load index (1-best; 7-worst) between our tested conditions (**C1**-**C3**). Participants rated *physical demand* ($A = 2.55, SD = 0.13$), *mental demand* ($A = 2.46, SD = 0.05$), *frustration* ($A = 2.55, SD = 0.10$), and *success* ($A = 2.44, SD = 0.26$) very similar between all conditions. However, the general workload was also low, which means that CHARM is useful for mobile system control tasks regardless of the used mapping condition. We were particularly pleased that participants assessed the elastic input very positively regarding haptic feedback and support for controlling the AR menu.

5.2 Second Formative Study

The goal of the second study was to compare our *CHARM* interaction concept with a suitable baseline to evaluate user satisfaction. Additionally we logged task completion times to gain first insights about user performance. We implemented a gesture interaction interface using the air tap provided by the Microsoft HoloLens in combination with a gaze-cursor. The cursor is used for selecting menu segments, while the air tap activates buttons, sliders and lists, and tap & hold manipulates slider and list values (see Fig. 5). Furthermore, we implemented two different manipulation techniques for the gesture interaction: The first is an position-based mapping, where moving the hand when performing a tap & hold gesture is directly mapped to moving the virtual slider or selected list item. The second approach uses a rate based system, where moving the hand up or down during the tap & hold gesture results in a continuous change of intensity the further the hand is moved away from the neutral position. For this study, we recruited seven participants (all male) aged between 20 and 28 from students of our local university, which did not participate in the first study. Six participants were right- and one was left-handed, all had experience with radial menus, but little experience with mixed reality.

Tasks and Procedure: To compare both interaction modalities each participant either started with the gesture interaction or the *CHARM* interaction. We alternated the order with every participant. For each modality, we again started with a short training session where participants would familiarize themselves with the current interaction scheme. After that we evaluated two different mappings for the specific modality, counterbalanced between participants. For the *Gesture* interaction, we evaluated the position-based (**G1**) and rate-based (**G2**) mappings and for *CHARM* interaction the thumb-joystick (**C2**) and string-deflection (**C3**) based mappings. We decided not to use mapping **C1** because results of our first study suggested that its least suitable for selecting menu segments. Participants solved two sequences of four tasks per condition, involving sub-menus and all widgets. The last task was always to activate a toggle button. Instead of the restaurant finder, we used a home automation use case were users could adjust a variety of home related values like lighting condition or temperature. Participants had to solve each tasks without help from the experimenter and were told to solve the tasks as quickly as possible (we did not encourage thinking aloud this time to not influence the time measurements). The overall duration of the study for each participant was approximately one hour.

Measurements: We again recorded the livestream from the Microsoft HoloLens and the current state of the *CHARM* device (which buttons were pressed, current deflection values, etc.), gestures recognized by the HoloLens and all events that were triggered within our menu. With this data we could accurately reproduce the actions of the users, which helped in identifying problems with the interaction mappings. Furthermore, we measured the completion times for each task sequence, taking the toggle button at the end of each sequence as completion mark. Similar to the first study, two investigators were always present, which one conducting the study and the other taking detailed notes of each participant. All sessions were accompanied by questionnaires of seven-point scales including a raw NASA-TLX [11], and questions on how easy it was to select menu items and manipulate sliders and lists with a particular interaction mapping.

Results: Somewhat surprisingly, task completion times for both gestures ($A_{\mathbf{G1}} = 49.71\,\mathrm{s}$, $A_{\mathbf{G2}} = 47.08\,\mathrm{s}$) and *CHARM* ($A_{\mathbf{C2}} = 44.52\,\mathrm{s}$, $A_{\mathbf{C3}} = 49.64\,\mathrm{s}$) were mostly the same, with *CHARM* being slightly faster overall. Due to the small number of participants we did not test for statistic significance. However, as a first indication of user performance, the results where nonetheless very interesting to us, as we did not expect this outcome. We observed that the selection of segments using the gaze cursor was a lot faster than using either **C2** or **C3**, while *CHARM* was faster manipulating sliders and lists. This is also supported by our questionnaire, were four participants stated that gestures supported them more for selection tasks, compared to two preferring *CHARM* and one undecided. For manipulation, *CHARM* was preferred by four participants and gestures by two, with one undecided. The results of the NASA-TLX (seven point scale, 1 being best and 7 being worst) showed distinctively less physical demand ($A_{\mathbf{G1}} = 3.71, SD_{\mathbf{G1}} = 1.16$, $A_{\mathbf{G2}} = 3.71, SD_{\mathbf{G2}} = 1.27$, $A_{\mathbf{C2}} = 1.57, SD_{\mathbf{C2}} = 0.49$, $A_{\mathbf{C3}} = 2.00, SD_{\mathbf{C3}} = 0.76$) for *CHARM* compared to gestures, and also reduced stress level ($A_{\mathbf{G1}} = 2.86, SD_{\mathbf{G1}} = 0.99$, $A_{\mathbf{G2}} = 3.14, SD_{\mathbf{G2}} = 1.64$, $A_{\mathbf{C2}} = 1.71, SD_{\mathbf{C2}} = 0.70$, $A_{\mathbf{C3}} = 2.14, SD_{\mathbf{C3}} = 0.63$). The other categories were very similar between both modalities, although *CHARM* scored slightly better in all categories in comparison to free hand gestures.

5.3 Discussion

Building of the insights of this study, we propose a hybrid approach of our *CHARM* concept, using the gaze cursor from our gesture interaction baseline for selecting menu items and using our *CHARM* device to manipulate them. Although we have not yet evaluated this approach, we are confident that this leads to a faster and more satisfying solution. Furthermore, we learned that users have very different preferences regarding their favored interaction mappings. Therefore, an important point for future developments is to provide customizable mappings or even their conjunction. For example, the deflection-based selection could be the default mapping, while using the thumb-joystick overrides the selection. Such a combined mapping scheme could enable a highly adaptive input device with synergetic interaction mappings that work seamlessly together and provide alternatives. Our own observations during both studies and discussions with our participants lead to the assumption that our *CHARM* device possesses a flexibility and adaptability which not only makes it suitable for system control tasks, but also as a generic input device for AR applications. This inspired us to extend our system to also incorporate an interaction scheme for the 3D transformation of objects, described in the next section.

6 3D Transformation

3D transformation is also an important aspect for AR applications and serves as an example of how our *CHARM* device can be used for the direct manipulation of objects within an application. Our concept and prototype incorporates seven degrees of freedom: Objects can be translated and rotated freely on all three axis and uniformly scaled. We decided for a uniform scale over a free scale on all three axis, because it does no distort the transformed object. Every transformation is always performed in relation to the users current position and orientation so that, e.g., moving the handle to the left always moves the object to the left as well from the perspective of the user.

Translation is activated by pressing and holding the trigger button and moving the handle in the desired direction (see upper Fig. 6). Releasing the trigger button stops the translation. We use a direct mapping between handle and virtual object, e.g., moving the handle one meter to the left also moves the transformed object one meter to the left. We also experimented with a rate-based mapping, where the rate of deflection of the handle in a certain direction determines the movement speed of the object in this direction. However, early tests with users showed that the direct mapping was preferred by users over the rate-based approach, as it was more precise and easier to understand.

Rotation is controlled by the thumb-joystick, using a rate-based quadratic transfer function to determine the speed with which the object is rotated. Since the joystick provides only two degrees of freedom, a mode switch is used to iterate through the different rotation axes. Only one rotation axis is active at a time and symbolized by a green circle around the object (see Fig. 6). The axis can be switched by pressing the joysticks push-button. Both deflection directions of the joystick result in a rotation of the object around the currently active axis. While it would be possible to map two axes at once, we found that this confuses the user more than it helps to reduce the required mode switches.

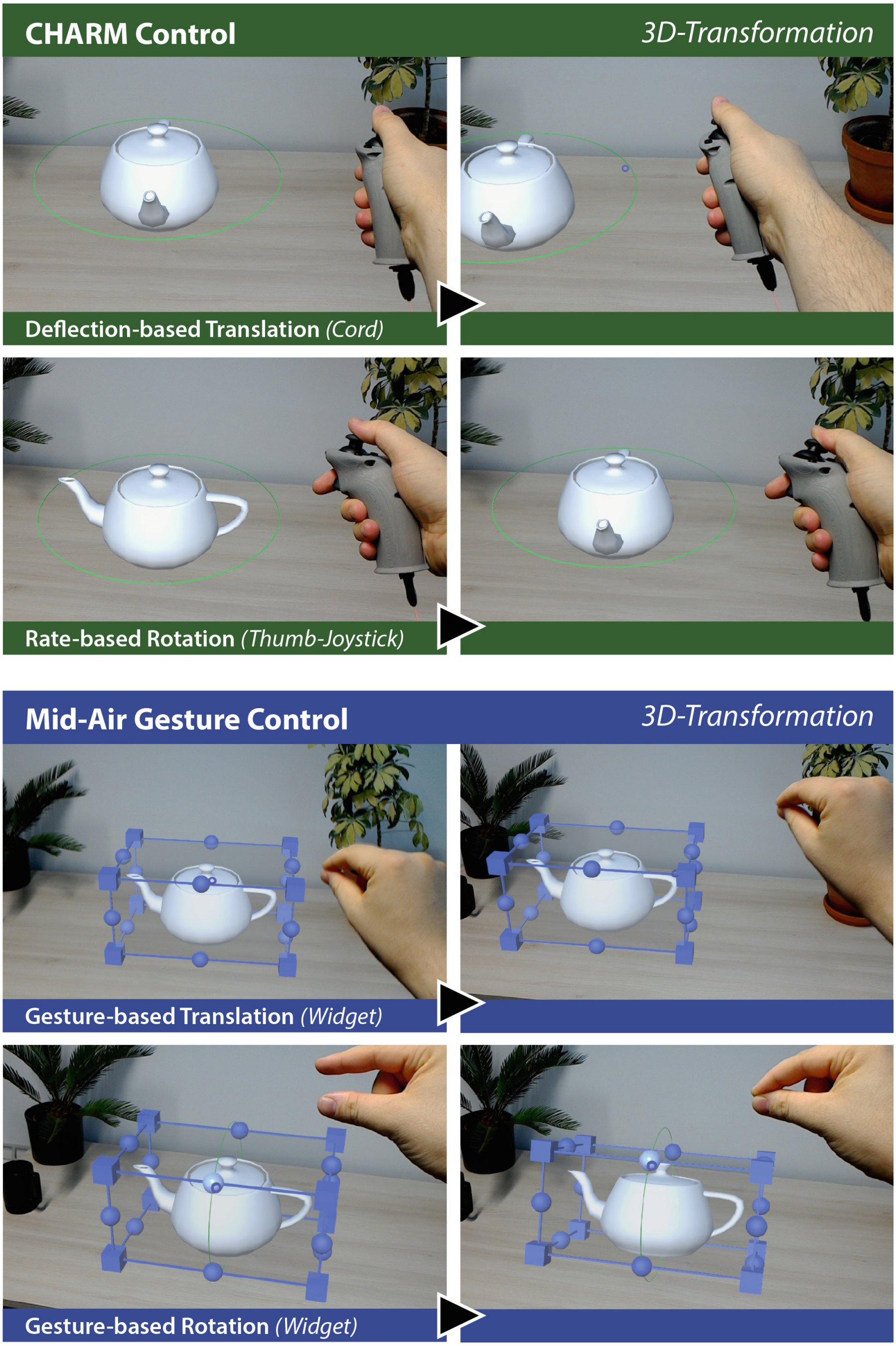

Fig. 6. 3D transformation using our cord-based techniques and mid-air gestures. (Color figure online)

Scale is activated by pressing and holding the left push-button and pulling the handle away from the body to enlarge the object and pulling it to the body to shrink it. Releasing the button stops the scale. The scale uses an linear mapping, where pulling the string 50 cm in one direction results in the object getting 50% smaller or larger. We again experimented with a rate-based mapping, but are convinced that a linear mapping works better and is more precise.

Our tests have also shown that our translation mapping is not suitable for moving objects over large distances like several meters, as this requires a lot of arm movement and is exhausting. We propose two solutions for this issue: The first is an alternative long range mode, toggled by the right push-button of the *CHARM*-device, which activates a non-uniform mapping of the deflection to the object's position. This means that moving the handle a certain distance translates into the object moving several times that amount. We made good experiences with a factor of five, but of course this can be freely configured according to the use-case. This provides our *CHARM* device with an imprecise long range mode on the direct mapping for the exact positioning of objects. The second solution is to harness the movement of the user itself by picking an object up, which results in the object moving in accordance to the user, and putting it down again, after which it can be positioned with the *CHARM* device as normal. We found the second solution to be preferable, as it enables users to pick up several objects at the same time and is also more intuitive than using a non-uniform mapping.

In addition to our *CHARM* transformation techniques, we also implemented a gesture interaction interface. The air tap supported by the HoloLens only provides three degrees of freedom. To compensate for this, our gesture interface therefore uses widgets (see lower Fig. 6). This is a contrast to the *CHARM* transformation, which makes nearly no use of widgets or additional visual feedback, with the sole exception being the circle indicating the current rotation axis. We based the widgets for the gesture interface on the ones the HoloLens itself uses, but have extended them to provide the same seven degrees of freedom *CHARM* provides. The widget can be thought of as a cube around the transformed object. It can be moved by executing a tap & hold on one of the cube's sides and moving the hand. The object will perform the exact same movement as the hand. To rotate and object on a specific axis, the corresponding handles on the edges of the cube are used. However, the object is always rotated around its center and not around the edge itself, with a green circle indicating the rotation axis. Scaling is performed by handles on the corner of the cube. Pulling the handle away from the object enlarges it, and polling the handle to the center shrinks the object. All transformations are done in relation to the users current position, e.g., moving the hand to the left always rotates the object to the left. This gesture interface acts as a baseline for a future evaluation of our 3D transformation concepts.

7 Conclusion and Future Work

In this paper, we investigated the potential of string-based elastic interaction for AR applications. Therefore, we presented *CHARM*, a retractable string-based device with a multi-DoF handle for generic tasks in AR. To demonstrate the suitability of our *CHARM* approach, we introduced a set of interaction concepts for system control (by means of AR menus) and 3D transformation. In order evaluate the feasibility of our body-centric elastic interaction approach, we built a fully-functional prototype. It can be seamlessly connected to state-of-the-art AR glasses, which we demonstrated on the example of the HoloLens. Based on mobile real-world interaction tasks, we evaluated our AR menu for system control within two small-scale user studies. Our results suggest that *CHARM* sufficiently supports generic AR interaction in a casual and easy to use way, while being useful for precise input. However, as gaze selection was rated as a promising input for rough selections, we proposed a hybrid input method of using *CHARM* in conjunction with a gaze cursor. In addition, we introduced a 3D transformation concept that allow users to translate, rotate or uniformly scale objects directly using our *CHARM* device.

For future work, our *CHARM* system needs to be miniaturized to enhance the degree of wearable integration. Furthermore, we plan to evaluate the differences between our *CHARM* and gesture interaction for the menu, as well as the 3D transformation in a future comparative quantitative user study We are confident, that *CHARM* provides a promising modality for interacting with three-dimensional content in Augmented Reality.

Acknowledgments. We would like to thank Andreas Peetz for helping us to improve the CHARM handle and Thomas Schwab as well as Paul Riedel for working on the AR menu and gesture control. This work was partly funded by the German Research Foundation (DFG, Deutsche Forschungsgemeinschaft) as part of Germany's Excellence Strategy – EXC 2050/1 – Project ID 390696704 – Cluster of Excellence "Centre for Tactile Internet with Human-in-the-Loop" (CeTI) of Technische Universität Dresden.

References

1. Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., MacIntyre, B.: Recent advances in augmented reality. IEEE Comput. Graph. Appl. **21**(6), 34–47 (2001). https://doi.org/10.1109/38.963459
2. Blasko, G., Narayanaswami, C., Feiner, S.: Prototyping retractable string-based interaction techniques for dual-display mobile devices. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2006, pp. 369–372. ACM, New York (2006). https://doi.org/10.1145/1124772.1124827
3. Bowman, D., Wingrave, C.: Design and evaluation of menu systems for immersive virtual environments. In: Proceedings IEEE Virtual Reality 2001, pp. 149–156 (2001). https://doi.org/10.1109/VR.2001.913781
4. Bowman, D.A., Kruijff, E., LaViola, J.J., Poupyrev, I.: 3D User Interfaces: Theory and Practice. Addison Wesley Longman Publishing Co., Inc., Redwood City (2004)

5. Coquillart, S., Wesche, G.: The virtual palette and the virtual remote control panel: a device and an interaction paradigm for the responsive workbench(tm). In: Proceedings IEEE Virtual Reality (Cat. No. 99CB36316), pp. 213–216, March 1999. https://doi.org/10.1109/VR.1999.756953
6. Dachselt, R., Hübner, A.: A survey and taxonomy of 3D menu techniques. In: Eurographics Symposium on Virtual Environments. The Eurographics Association (2006). https://doi.org/10.2312/EGVE/EGVE06/089-099
7. Davis, M.M., Gabbard, J.L., Bowman, D.A., Gracanin, D.: Depth-based 3D gesture multi-level radial menu for virtual object manipulation. In: 2016 IEEE Virtual Reality (VR), pp. 169–170, March 2016. https://doi.org/10.1109/VR.2016.7504707
8. Dobbelstein, D., Winkler, C., Haas, G., Rukzio, E.: PocketThumb: a wearable dual-sided touch interface for cursor-based control of smart-eyewear. Proc. ACM Interact. Mob. Wearable Ubiquit. Technol. **1**(2), 9:1–9:17 (2017). https://doi.org/10.1145/3090055
9. Fröhlich, B., Plate, J.: The cubic mouse: a new device for three-dimensional input. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2000, pp. 526–531. ACM, New York (2000). https://doi.org/10.1145/332040.332491
10. Gebhardt, S., Pick, S., Leithold, F., Hentschel, B., Kuhlen, T.: Extended pie menus for immersive virtual environments. IEEE Trans. Visual. Comput. Graph. **19**(4), 644–651 (2013). https://doi.org/10.1109/TVCG.2013.31
11. Hart, S.G.: NASA-task load index (NASA-TLX); 20 years later. Proc. Hum. Factors Ergonomics Soc. Annu. Meet. **50**(9), 904–908 (2006). https://doi.org/10.1177/154193120605000909
12. Hinckley, K., Pausch, R., Goble, J.C., Kassell, N.F.: A survey of design issues in spatial input. In: Proceedings of the 7th Annual ACM Symposium on User Interface Software and Technology, UIST 1994, pp. 213–222. ACM, New York (1994). https://doi.org/10.1145/192426.192501
13. Hyeongmook, L., Dongchul, K., Woontack, W.: Graphical menus using a mobile phone for wearable AR systems. In: 2011 International Symposium on Ubiquitous Virtual Reality, pp. 55–58, July 2011. https://doi.org/10.1109/ISUVR.2011.23
14. Klamka, K., Dachselt, R.: Elasticcon: elastic controllers for casual interaction. In: Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services, MobileHCI 2015, pp. 410–419. ACM, New York (2015). https://doi.org/10.1145/2785830.2785849
15. Klamka, K., Dachselt, R.: ARCord: visually augmented interactive cords for mobile interaction. In: Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems, CHI EA 2018, pp. LBW623:1–LBW623:6. ACM, New York (2018). https://doi.org/10.1145/3170427.3188456
16. Koch, E., Witt, H.: Prototyping a chest-worn string-based wearable input device. In: 2008 International Symposium on a World of Wireless, Mobile and Multimedia Networks, pp. 1–6, June 2008. https://doi.org/10.1109/WOWMOM.2008.4594882
17. Lee, H., Woo, W.: Tangible spin cube for 3D ring menu in real space. In: CHI 2010 Extended Abstracts on Human Factors in Computing Systems, CHI EA 2010, pp. 4147–4152. ACM, New York (2010). https://doi.org/10.1145/1753846.1754117
18. Olwal, A., Moeller, J., Priest-Dorman, G., Starner, T., Carroll, B.: I/O braid: scalable touch-sensitive lighted cords using spiraling, repeating sensing textiles and fiber optics. In: Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology, UIST 2018, pp. 485–497. ACM, New York (2018). https://doi.org/10.1145/3242587.3242638

19. Piekarski, W., Thomas, B.H.: The tinmith system: demonstrating new techniques for mobile augmented reality modelling. In: Proceedings of the Third Australasian Conference on User Interfaces, AUIC 2002, vol. 7, pp. 61–70. Australian Computer Society Inc., Darlinghurst (2002). http://dl.acm.org/citation.cfm?id=563985.563994
20. Pohl, N., Hodges, S., Helmes, J., Villar, N., Paek, T.: An interactive belt-worn badge with a retractable string-based input mechanism. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2013, pp. 1465–1468. ACM, New York (2013). https://doi.org/10.1145/2470654.2466194
21. Schwarz, J., Harrison, C., Hudson, S., Mankoff, J.: Cord input: an intuitive, high-accuracy, multi-degree-of-freedom input method for mobile devices. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2010, pp. 1657–1660. ACM, New York (2010). https://doi.org/10.1145/1753326.1753573
22. Simon, A., Froehlich, B.: The YoYo: a handheld device combining elastic and isotonic input. In: Human-Computer Interaction INTERACT 2003: IFIP TC13 International Conference on Human-Computer Interaction, 1st–5th September 2003, Zurich, Switzerland (2003)
23. Szalavári, Z.: The personal interaction panel - a two-handed interface for augmented reality. Ph.D. thesis, Institute of Computer Graphics and Algorithms, Vienna University of Technology, Vienna, Austria (1999). https://www.cg.tuwien.ac.at/research/publications/1999/Szalavari-thesis/
24. Tsai, H.R., Rekimoto, J.: ElasticVR: providing multi-level active and passive force feedback in virtual reality using elasticity. In: Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems, CHI EA 2018, pp. D300:1–D300:4. ACM, New York (2018). https://doi.org/10.1145/3170427.3186540
25. Weigel, M., Steimle, J.: Deformwear: deformation input on tiny wearable devices. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. **1**(2), 28:1–28:23 (2017). https://doi.org/10.1145/3090093
26. White, S., Feng, D., Feiner, S.: Interaction and presentation techniques for shake menus in tangible augmented reality. In: 2009 8th IEEE International Symposium on Mixed and Augmented Reality, pp. 39–48, October 2009. https://doi.org/10.1109/ISMAR.2009.5336500