doi:10.1093/iwcomp/xxxxxx

# TouchRoller: A Touch-sensitive Cylindrical Input Device for GUI Manipulation of Interactive TVs

(Removed for Blind Review)

*(Removed for Blind Review)*

**The two main tasks of a smart TV GUI are menu navigation and free pointing. Traditional remotes with directional keys are appropriate for menu navigation but not for free pointing. More recent remotes with a two-dimensional (2D) pointing device are appropriate for free pointing but not for menu navigation. To support both types of tasks well, we devised a new input device called TouchRoller. We expect that it can support both types of tasks well because it has a separable control structure and a continuous input property. A comparative user study showed that the performance of TouchRoller is comparable to that of directional keys for menu navigation and 2D pointing devices for free pointing. In addition, it was most favored by the participants, and NASA TLX test results showed that TouchRoller demands the lowest task load.**

*Categories and subject descriptors: H.5.2 User Interfaces: Input devices and strategies (e.g., mouse, touchscreen)*

*Keywords: TouchRoller, input device, interactive TV, separable control structure, continuous input property*

*Responsible Editorial Board Member: Name*

## 1. INTRODUCTION

Since the introduction of smart TVs, a TV is no longer a one-way terminal for watching broadcast content but is now an interactive interface for diverse multimedia sources, such as broadcasting stations, video-on-demand services, and the Web. Due to this increased interactivity, a smart TV requires a more efficient input device than a traditional TV remote that primarily consists of dedicated buttons for different TV functions. In particular, a traditional remote with directional keys is not suitable for controlling a smart TV GUI that requires free pointing. Therefore, many new remote controls are currently being introduced to the market. Some examples include a remote control equipped with a gyro-sensor (MacKenzie and Jusoh, 2001), a touchpad (Enns and MacKenzie, 1998), and a vision-based gesture interface (Freeman and Weissman, 1995). One of the concerns about the new remote controls is that, although they are better than directional keys for a pointing task, they are not as fast and accurate as directional keys in a menu navigation task. In other words, existing remotes are suitable only for one of the two main tasks of a smart TV GUI: menu navigation and free pointing. This issue is worth noting because the main components of a smart TV GUI are menu interfaces that show various types of content.

In order to devise an input device that can handle both menu navigation and free pointing, we considered the compatibility between the characteristics of a GUI task and the characteristics of an input device. First, we noted that they should be compatible with respect to the integrality and separability aspects of the input dimensions (Jacob et al., 1994). The menu navigation task of a smart TV usually has a separable perceptual structure while a free pointing task such as Web browsing has an integral perceptual structure. It is expected that directional keys such as a d-pad and arrow keys that have a separable control structure may be better for menu navigation, whereas two-dimensional (2D) pointing devices[1] such as a touchpad or an isometric joystick that have an integral control structure may be better for free pointing. Next, we noted that the continuity property of a GUI task and an input device should also be compatible. In menu navigation, the highlight moves icon by icon in a discrete manner, while the cursor in free pointing moves continuously to select a target at an arbitrary position. With respect to continuity, it is expected that directional keys are appropriate for menu navigation, whereas a 2D pointing device is better for free pointing.

The question that arose at this point was whether there could be an input device that had a separable control structure similar to directional keys but continuous input dimensions similar to a pointing device. Furthermore, if such a device existed, whether it could be as good as directional keys for menu navigation and pointing devices for free pointing. Regarding the first question, we introduce TouchRoller, a new input device for smart TV remote control that has such properties. Given an answer to the first question, we investigated whether the TouchRoller could be as good as directional keys for menu navigation and as good as a pointing device for free pointing. To answer this question, we conducted a user study to compare the performance of TouchRoller with directional keys and a pointing device for menu navigation and free pointing tasks, respectively.

In Section 2, we review the background work to the current research, including the usability issues of interactive TVs and their remote control options. In Sections 3, 4, and 5, we introduce the TouchRoller concept and the implementation details of the TouchRoller prototype. In Section 6, we present the user experiment design and results that verify the properties of the TouchRoller set forth above. In Sections 7 and 8, we discuss a few issues that need further study and conclude with a summary of the current study's contributions.

## 2. BACKGROUND

### 2.1. Characteristics of interactive TV

As TV transformed from a passive display of broadcast content to an interactive platform for on-demand content, researchers started to study the characteristics of interactive TV environments and presented guidelines for designing user interfaces for interactive TV.

Ju et al. (1994) present a list of special characteristics for an interactive TV environment that they consider should motivate the design of a TV pointing device. For instance, users are commonly at some distance from the display and may not have an accessible working surface for using a mouse. Moreover, the room may be dark and most TV screens cannot display small text and images well. Finally, there may be children, pets, food, noise, or multiple users in the room. Thus, Ju et al. present a list of guidelines for TV pointing devices. For example, they claim that it is necessary for the pointing device to move the cursor smoothly across the screen, and they recommend that remote controls be operable with one hand.

Lekakos et al. (2001) argue that there are several challenges to designing interactive TV applications owing to the differences between the interactive TV medium and traditional information systems, in terms of input and output devices, the viewing environment, and so on. For example, they point out that users should be able to perform all tasks for interactive TV using a single remote controller, including controlling video, entering personal codes, and moving the pointer and cursor. In particular, they point out the need for an alternative text-entry method to replace the need for a physical keyboard.

Lu (2005) suggests principles for designing an interactive TV-application interface. Lu shows that the design of such a user interface should make it easy to navigate a large amount of content because interactive TV provides many services with various content. In particular, Lu recommends a grid layout because of its repeatability and the ease with which users may be guided to important elements. In fact, a grid layout is a common feature in recent smart-TV products.

Bernhaupt et al. (2007) investigated usability issues related to interactive TV by performing user tests in a simulated domestic living room. One of the recommendations based on their observations is that only the directional keys and the OK key of a remote should be used for navigation within the information space. They also address learnability issues, such as learning the colored keys of a remote, and suggest the need for supporting a learning process. They also point out the difficulty involved in entering text with a remote and suggest the need to adopt a method familiar to users, such as the multi-tap input method.

These studies, along with current trends in smart-TV products on the market, attest to the fact that TV now requires a rich user interface for diverse interactive TV tasks, and that such a user interface now requires a remote-input option − beyond a traditional remote controller − that is designed keeping the characteristics of interactive TV in mind.

[1]By a 2D pointing device, we refer to a 2D pointing device with an integral control structure such as a mouse, touchpad, trackball, or an isometric joystick. In this paper, we refer to them simply as pointing devices.

## 2.2. Options for remotely controlling interactive TV

Many different methods have been proposed for remotely controlling interactive TV, and their advantages and disadvantages have been studied in the literature. The first and most common option is to use a dedicated remote controller, such as traditional button-based remotes and newer ones that include a pointing device. The second option is to use in-air gestures, making it unnecessary to use a hand-held device. The third option, which is becoming popular alongside the increasing ubiquity of smartphones and tablet computers, is to use a second screen. Other options that have been proposed include a speech-based interface. The applicability of a speech interface for a TV environment is questionable, however, considering that TV viewing is often shared among family members, and the current speech-interface technology is not yet sufficiently mature to understand the context of a family conversation. Hence, in this section, we focus on the first three options only.

### *2.2.1. Remotes with a pointing device*

In order to cope with the increasing complexity of the TV user interface, remote controllers have been manufactured with an embedded pointing device, such as a touchpad or a gyro-mouse. One of the most common types, both in the literature and on the market, is a remote control with a touchpad. One of the earliest studies on the touchpad remote was conducted by Enns and MacKenzie (1998). They performed a preliminary study to examine the feasibility of using a touchpad for remotely controlling a TV. One of the earliest examples of such a device in the industry was Panasonic's EZ Touch Remote (Panasonic, 2008) – the first touchpad-based remote that attempted to use an absolute mapping from a touchpad to the TV screen. More recently, Choi et al. (2011) introduced a remote with a hover-tracking touchpad to address the problem of divided visual attention between the TV screen and the remote. Beyond pointing, Aoki et al. (2011) proposed a set of thumb gestures to expand the number of possible commands that can be issued from a TV remote with a touchpad. Many smart-TV products currently available on the market manufactured by companies such as Samsung, Sony, and Panasonic include a touchpad remote.

Another popular option is a remote with air-mouse capabilities. MacKenzie and Jusoh (2001) were among the first to research the feasibility of such a device. They compared different remote pointing devices for interactive TV, including an air mouse and a thumb-controlled isometric joystick. Their experiment showed that the air mouse was faster, though less accurate, than the isometric joystick. Sohn and Lee (2004) compared the throughputs of pointing-device candidates for interactive TV. Two types of air mice and two types of joysticks were compared. They were all comparable in terms of performance, yet they were all inferior to a trackball or a touchpad. Lee et al. (2011) compared the throughputs of three types of air mice in a TV environment: a gyro-mouse, a Wii-mote, and an optical air mouse. Their experimental results showed that Nintendo's Wii-mote was the fastest among the three. The performance of an air mouse in a TV context appears to depend on many design parameters, such as the coordinate-mapping scheme (e.g., whether it is absolute or relative), the control-display gains, and the cursor-stabilization method. The effect of such design parameters on the performance of an air mouse in a TV environment is still a subject for further investigation.

In addition to the pointing performance, the design of possible gestures for an air-mouse has been a popular research subject. One early study by Kela et al. (2006) discusses examples of air-mouse gestures for applications such as TV and VCR, and studies the role of spatial gestures in a multimodal interface. More recently, Bailly et al. (2011) compiled guidelines for designing a gestural vocabulary and presented interaction techniques with a Nintendo Wii-mote. They designed a rotational gesture set suitable for controlling TV content and evaluated its user-preference in different contexts. The design of air-mouse gestures for text entry has been also a popular research subject. Jones et al. (2010) developed key layouts and hand gestures to enable users to enter text using an air-mouse. A more recent example was offered by Vatavu (2013). Exploring the concept of an augmented-TV environment utilizing the space surrounding a TV, Vatavu conducted a participatory design study to find a suitable set of gestures with Nintendo's Wii-mote for augmented-TV interaction.

Magic Remote by LG is a representative example of an air-mouse on the market that comes with a smart TV. Although it is not designed for TV viewing, Nintendo's Wii-mote is another representative air-mouse that is widely used. Despite their popularity, these air-mice have been known to have problems related to fatigue and instability (i.e., jittering) because they must be operated in mid-air (Olsen and Nielsen, 2001; Myers et al., 2002).

### *2.2.2. Freehand gestures*

Using freehand gestures to control an interactive TV has had a long history, and recently has witnessed a renewed interest. Freeman and Weissman (1995) provide an early example of controlling a TV remotely using hand gestures. With their prototype, a single gesture of an open hand was used to control a cursor on the screen. More recently, Stenger et al. (2009) presented a similar gesture interface for a TV, with which users can control a TV using hand movements and an activation mechanism. Chen et al.

(2010) demonstrated a hand-gesture recognition system for controlling a TV. The system does not rely on a cursor, but rather recognizes various hand gestures to, for example, control the audio volume and turn the TV on/off. Vatavu (2012) investigated users' preferences for freehand gestures when controlling a TV, and proposed a set of gesture commands for basic TV-control tasks. Dias et al. (2013) proposed a natural gesture-based user interface to interact with a Flickr client application on a smart TV. The interface was based on a depth sensor (Microsoft Kinect), and users were able to search and browse pictures merely with gestures. Wu and Wang (2012) conducted a guessability study for hand-gesture inputs in a TV-viewing context. As a result, they presented a hand-gesture taxonomy and a user-defined gesture set for basic TV commands – e.g., for playing, pausing, and stopping a video.

Freehand gestures allow users to manipulate a TV without an additional handheld device, but performing numerous consecutive gestures can lead to fatigue for users. A possible solution to this problem is to permit smaller fine-grained finger gestures that require less physical effort than gestures involving the entire hand. One such model is provided by Vatavu and Zaiti (2014), who conducted a gesture-elicitation study to derive a set of guidelines for designing *Leap Motion* gestures in the context of interactive TV. They also presented a set of finger-gesture examples for controlling various functions of a TV set. Another study of small gestures is given by Dezfuli et al. (2012), who investigated the concept of a PalmRC: an eyes-free palm-surface-based TV remote control. Leveraging the non-dominant hand as an interactive input surface, users could operate a TV by interacting with the other hand's index finger.

#### *2.2.3. Using a secondary screen*

Using a secondary screen is an increasingly viable option for remotely controlling a TV, as smart devices such as smartphones and tablet computers are now becoming more available. Robertson et al. (1996) introduced an early example of a multi-device application consisting of a PDA that operates in conjunction with an interactive TV. More recently, Cruickshank et al. (2007) conducted interviews to investigate the possibility of using a PDA as a remote control. User comments from the interviews were mostly positive. One of the negative comments noted the difficulty in selecting small icons in the secondary screen with the bare hand. After reviewing previous research, Cesar et al. (2008) identified the four primary uses of a secondary screen in an interactive TV environment: controlling, enriching, sharing, and transferring television content. They also shared their experiences in developing relevant scenarios as well as an initial evaluation of them. Lin et al. (2012) presented a prototype system that can recognize the contextual situation of each TV program. The system then provides a different visual user interface on the viewer's handheld device. Their prototype is an example of technology that benefits from the advanced functionality of a smart device – in this case, context awareness.

According to a study by Fleury et al. (2012), the use of a secondary device leads to a division in the visual attention of the user between the TV content and the secondary device. When using the secondary device, a user can follow the TV content by listening to it. However, the user cannot follow TV content that require a higher level of engagement whilst using the secondary device. A related study was conducted by Rashid et al. (2012). They compared the following three combinations of a large display and a mobile device: 1) a mobile device is used as a touchpad to control a large display, 2) both a large display and a mobile device are used for output, and 3) only a mobile device is used. Their results showed that case 2, where visual content is distributed across devices, is the inferior option.

A brief review of the three options for remote control has thus far revealed that each option has its strengths and weaknesses. A remote with a pointing device has enjoyed a long history and is currently the most popular option. Yet, it is criticized for requiring a separate and dedicated device. The freehand gesture interface is actively researched, both in academia and in the industry, and is expected to evolve rapidly into a viable option. However, the technology is not yet sufficiently mature to be accepted as a practical option on the market (Trusted Reviews, 2013). In addition, the freehand gesture interface in a TV-viewing context still suffers from unresolved issues, such as the fatigue problem and problems related to social acceptance. Using a second screen is quickly becoming a practical option, owing to the spread of smartphones. In fact, almost every smart-TV manufacturer is currently providing smartphone applications for this purpose. However, its adoption by consumers is slow, possibly because of the clash between the personal attributes of a smartphone and the shared experience of the TV-viewing environment.

It is still too early to predict which of the three options will dominate in the future. It seems that we are in a phase of parallel exploration of multiple possibilities. Of the three options, the TouchRoller is an example of the first: a TV remote with a pointing device. However, the TouchRoller is distinct from other examples in this category, such as the touchpad remote and the air mouse, because of its separable control structure and continuous input property, which will make it unique as an input device that is suitable for both 2D pointing and menu-selection tasks.

At the time of writing it was brought to our attention that there exist previous examples of an input device with a roller. A TV remote by Morito (1995) has a roller, and a user can rotate or exert force on it. Whereas TouchRoller and Morito's device look similar, they are fundamentally different; TouchRoller utilizes the horizontal position of the thumb on the roller while Morito's device does not. Another example is RollerMouse (Bohan et al., 2003), which has a roller and a user can slide and rotate it with a hand. RollerMouse, however, is designed for use in a desktop computer environment aiming to replace a mouse. In addition, its form factor and usage are clearly distinct from that of TouchRoller.

## 2.3. Integrality and separability

The perceptual structure and the control structure influence the performance of a device (Jacob et al., 1994). The perceptual structure is either separable or integral, depending on the users' perception of the attributes that are controlled in the task. An integral perceptual structure is found where there is a relation between attributes that are perceived as connected. For example, according to the previous study, the lightness and saturation of an object's color are perceived as changing together, having an integral perceptual structure. On the other hand, attributes with a separable perceptual structure are perceived by users to change independently. For example, the shape and color of an object are attributes with a separable perceptual structure (Handel and Imai, 1972).

The control structure refers to the way a device is controlled. When a device has more than one degree of freedom (DOF), the control structure is separable when each dimension is controlled independently. An example of a device with the separable control structure is a d-pad, which includes a separate key for each direction. On the other hand, the control structure is said to be integral when a device allows the user to control more than one dimension at the same time. A touchpad is an example of a device with an integral control structure. Users can move the pointer on a touchpad in a horizontal, vertical, or diagonal direction with a touchpad, changing the x and y coordinates at the same time. According to a previous study (Jacob et al., 1994), better performance is assured when there is agreement between the control structure and the perceptual structure.

There are devices and input methods that are designed based on the perceptual and control structures. Multitouch gestures were designed to have a control structure suitable to 3D manipulation tasks (Martinet et al., 2012). Another study shows that the control structure has an effect on the performance of navigation tasks in a 3D virtual environment (Casiez and Chaillou, 2005). Based on the design theory and the examples cited,

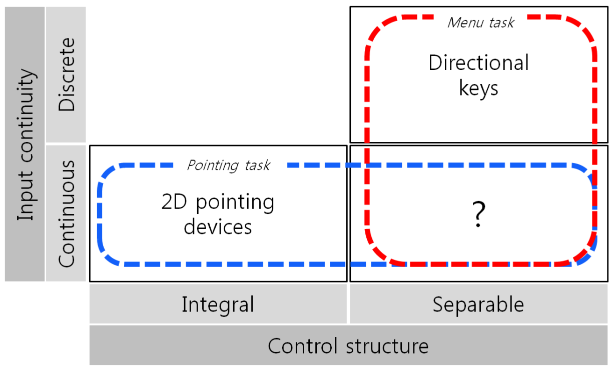

**Figure 1.** Properties of the devices and GUI tasks.

TouchRoller is designed to have a control structure suited for manipulating the content of interactive TV.

## 3. TOUCHROLLER

Directional keys are effective for a menu navigation task, but are not suitable for a pointing task. On the other hand, a pointing device is effective for a free pointing task, but does not perform very well during a menu navigation task. We conjectured that their performance differences with respect to the two GUI tasks are caused by differences in their control structures and input continuity properties. Figure 1 shows an input device space defined by two dimensions: control structure and input continuity. In this space, directional keys belong to the separable-discrete quadrant while a pointing device belongs to the integral-continuous quadrant. In the same space, we may map different GUI tasks, considering their perceptual structures and input continuity properties. A menu navigation task has a separable perceptual structure; hence, the red dashed rectangle in the device space. On the other hand, a free pointing task requires a cursor to be moved to an arbitrary point on the screen; hence the blue dashed rectangle in the device space. The resulting diagram is one model that explains the performance differences of the two devices for the two GUI tasks.

Figure 1 suggests that if there was an input device that belonged to the separable-continuous quadrant, it may support both tasks well. This was the primary motivation of the current research. Hence, we developed TouchRoller, an input device in the form of a roller, as shown in Figure 2(a). It can detect finger position along its length and map it to the horizontal displacement on the screen. It also rolls around its axis and maps its rotation angle to the vertical displacement on the screen. A notable property of TouchRoller is that its control structure is

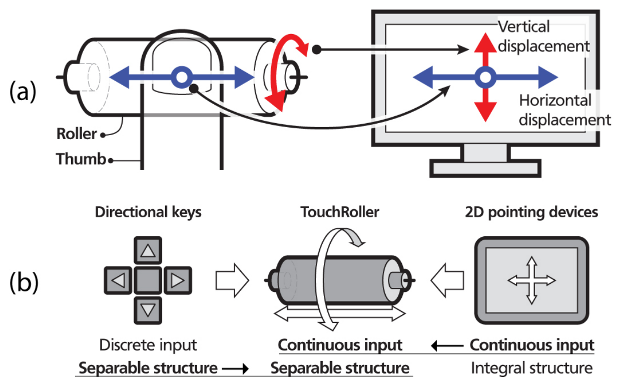

**Figure 2.** TouchRoller's (a) two input variables and (b) position relative to directional keys and pointing devices.

largely separate. It is possible to move the finger along the roller while rotating the roller, but the two operations are generally performed separately. Another property of TouchRoller is that both the movement along the roller and the rotation of the roller are continuous, meaning that its input is continuous. In short, the new device belongs to the separate-continuous quadrant of Figure 1.

The position of TouchRoller in relation to directional keys and pointing devices is depicted in Figure 2(b). TouchRoller is between the two devices as it shares the separable control property with directional keys and the continuous input property with pointing devices. As it shares a separate control property, it should perform as well as directional keys for menu navigation tasks. Because it shares a continuous input property, it should also perform as well as a pointing device for free pointing tasks. Thus, we expect that the new device will appropriately support both types of tasks. The goal of the current research is to verify this expectation through experiments.

## 4. PROTOTYPE IMPLEMENTATION

### 4.1. Hardware

Figure 3(a) shows the final TouchRoller prototype after several iterations of design and evaluation. The prototype is shaped as an inverted triangle. This design allows the thumb to be perpendicular to the roller when the prototype is held in the hand. As it is laterally symmetric, the design is equally effective for both right- and left-handed users. The length of the roller is approximately 36 mm, and the diameter of the roller is approximately 12 mm. The length of the roller is a compromise between a comfortable movement range for the thumb and the need to minimize clutching actions on the roller.

Figure 3(b) shows the sensors around the roller to detect roller thumb operations. All these sensors are common off-the-shelf electronic components. An optical mouse displacement sensor under the roller detects its rotation. We chose the ADNB3532 (Avago) because of its small and thin form. The prototype uses its default resolution of 500 cpi (counts per inch). As the roller surface moves as the thumb moves vertically, this resolution corresponds to the resolution of the prototype's vertical movement sensing ability. By the way the roller in this prototype is not spin-able. The roller, made of brass, has enough inertia to be spin-able, but its spin-ability seemed to cause more stability problems than efficiency benefits. Therefore, we added some amount of intentional friction to the roller mechanism to make the roller not spin-able in the current prototype.

To detect the horizontal position of the thumb on the roller, we chose to construct a linear optical sensor based on the optical imaging method introduced by Choi et al. (2011). It consists of an array of LEDs and an array of phototransistors. Phototransistors are connected in parallel and act as a single photosensor. The LEDs in the array are turned on and off sequentially, one at a time, from left to right. Light reflected from the thumb is then measured by the photo-transistor array. The output of the photo-sensor for a single scan of the LED light sequence is a proximity image of the thumb on the roller, and its centroid is used to estimate the position of the thumb. The estimate of the thumb position is not considerably accurate because the LEDs are not placed exactly on the roller. This, however, was not a problem in practice because we do not need the absolute position of the thumb but the relative displacement of the thumb on the roller.

To enable a clutching action on the roller, it is necessary to sense the contact state of the thumb on the roller. To this end, we added a touch sensor to the roller. As the touch sensor is capacitive, the roller, as well as the path to it, has to be made electrically conductive. The roller and the supporting structure are made of brass, and the roller is covered with an insulating layer.

In addition to the roller, the TouchRoller prototype has a button under the roller, as shown in Figure 3(b). It is made of a transparent material in order not to interfere with the optical sensor. It feels similar to a mouse button and is used for target selection, similar to a left mouse button.

### 4.2. Signal processing

Figure 4 illustrates the signal paths from the sensors to the TouchRoller events that are sent to the GUI system. All sensors are sampled at a rate of 56 Hz. There are four different events, dX, dY, Up, and Down, where dX and dY are the horizontal and vertical movement events, and Up and Down are button events.

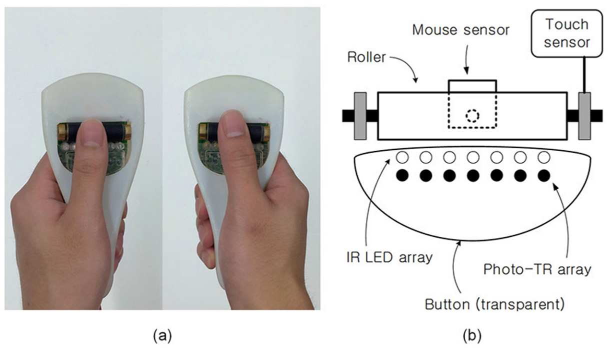

**Figure 3.** TouchRoller hardware: (a) the ergonomic shape of the prototype and (b) the sensors around the roller.

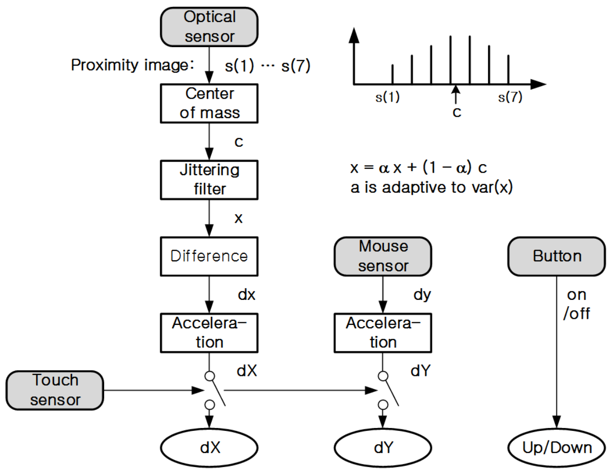

**Figure 4.** Signal processing from the sensors to the GUI events.

The touch sensor output is a binary value indicating the touch state of the finger on the roller. This touch variable only gates displacement events from the optical and mouse sensors and is not sent to the GUI system. The optical sensor outputs a linear proximity image of the thumb on the roller. The centroid of the proximity image is calculated to estimate the thumb position on the roller. The next step is a low-pass filter to reduce jitters caused by ambient optical noise. Finally, the change in thumb position x is calculated, and therefore, if it is larger than a certain threshold, a dx-event is generated. Displacement dx is scaled dynamically to calculate dX using an acceleration algorithm that is further described in the Appendix.

The mouse sensor outputs integer values proportional to the vertical thumb displacement dy on the roller since the last output. The signal processing module generates a dy-event whenever dy is not zero. The displacement dy is scaled dynamically to calculate dY with an acceleration algorithm that is also described in the Appendix.

### 4.3. Redesign iterations

Before we arrived at the final design, we went through several redesign iterations, during which several different designs of TouchRoller were implemented and evaluated. We summarize here a few major changes that were implemented during these iterations.

The first TouchRoller prototype had a 25 mm roller length that was shorter than the final design. The short length of roller led to a high Control-Display (CD) gain in order to reduce clutching when a user needed to move the cursor from the left to the right end, and vice versa. However, the high CD gain also triggered overshooting and reduced performance. Therefore, a longer roller and lower CD gain were used in the final prototype.

The first prototype also used a different target selection method. A force sensor was placed under the roller to measure the force applied to the roller, and a user pressed the roller to select a target. The problem with this design was inadvertent selections caused by the force exerted on the roller when the user rotated it. To resolve this problem, we changed the target selection method to a tap on the surface of the roller in the next prototype. However, even this method caused false activations while manipulating the roller. Therefore, the final design of TouchRoller has a separate button for selection.

The shape of the first prototype was a rectangular bar similar to a common TV remote. With this shape, the angle between the roller and the thumb was not perpendicular. Such a posture caused a horizontal shift in the thumb position when a user rotated the roller. Therefore, the next prototype was given an inverted triangular shape. It has a narrow handle part and wide area around the roller to allow the angle between the roller and the thumb to be perpendicular when the user holds the prototype. Figure 3 shows the final design of the TouchRoller prototype.

## 5. SMART TV GUI

To evaluate the performance of TouchRoller as a TV remote control, we developed prototype GUIs to represent the ones used in smart TVs. The GUIs provided in smart TV can be divided into two categories: menus and pointing interfaces. Menu interfaces can be classified further into three interfaces based on the arrangement of the contents: horizontal menu, vertical menu, and grid interfaces. Figure 5 shows examples of the three menu interfaces as well as a pointing interface commonly used by smart TVs. Figure 6 shows the four prototype interfaces that we implemented (horizontal menu, vertical menu,

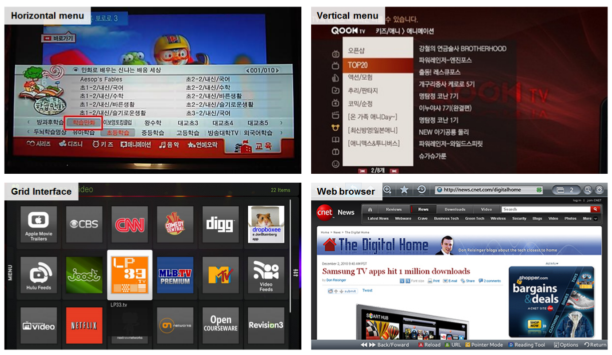

**Figure 5.** GUI examples from a real smart TV.

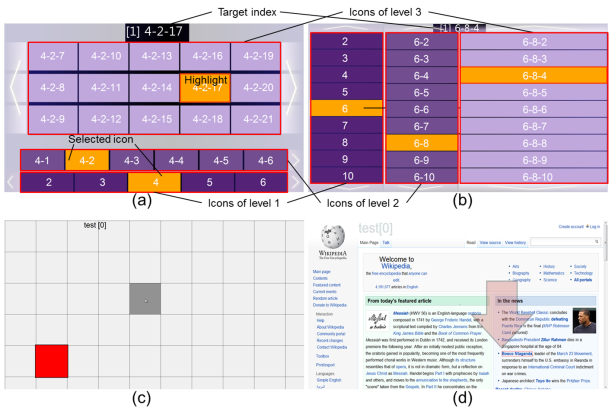

**Figure 6.** Prototype interfaces: (a) horizontal menu, (b) vertical menu, (c) grid, and (d) Web browser.

grid, and Web browser interfaces) corresponding to the four examples in Figure 5. The screen resolution of the three menu interfaces was 1920 × 1080 pixels and that of the Web browser was 1280 × 720 pixels.

### 5.1. Horizontal and vertical menu interfaces

Figure 6(a) and 6(b) show the horizontal and vertical menu interfaces that we implemented for the user study. Both interfaces have ten icons at the first level and ten icons at the second level. At the third level, there are thirty icons. The horizontal menu interface has 336 × 130 pixel icons at the first level, 277 × 136 pixel icons at the second level, and 318 × 180 pixel icons at the third level. The vertical menu interface has 47 × 104 pixel icons at the first and second levels and 950 × 104 pixel icons at the third level. We determined the numbers and sizes of icons at different levels based on common smart TV GUI designs on the market.

When a pointing device such as a touchpad is used, a visual cursor is displayed on the screen and a menu icon under the cursor is highlighted. When directional keys are used, no visual cursor is shown on the screen. In the case of TouchRoller, whether to show a cursor or not was an important design issue, as the TouchRoller produces continuous input but has a separable control structure. Based on a preliminary pilot study in the early design iterations, we decided not to display the cursor. Users could easily control the highlight in the menu without a cursor, possibly because of its separable control structure. In addition, a cursor in practice caused more interference than guidance. However, the decision to not display a cursor created a mapping issue from the continuous position input to a discrete highlight change. Simple discretization of the continuous input often resulted in an error when the continuous input was near the boundary of a menu item. Therefore, we added a simple hysteresis mechanism to the continuous-to-discrete TouchRoller mapping.

In the user study described in Section 6, participants selected a target in the menu from a multi-level target index. For instance, when they were instructed to select target 4-2-17, they selected the fourth category in the main menu, the second category in the sub-menu, and finally the seventeenth item in the sub-sub-menu. The menu items were labeled with multi-level target indices as shown in Figure 6. When TouchRoller or a pointing device was used, the task completion time was measured from when a cursor started to move to when the target was selected. When directional keys were used, the task completion time was the time between the first push of a key to target selection.

### 5.2. Grid interface

Figure 6(c) shows the grid interface. The gray cell is the currently selected cell, and the red cell is a target cell. The cell size is 200 × 200 pixels and is same as the icon size of a grid interface found in a smart TV (e.g., Samsung). The same issues that we considered in the case of the menu interfaces, such as the visibility of the cursor and the mapping from a continuous position input to a discrete highlight change, were similarly applied to the current case.

In the user study, participants moved the highlighted cell that was initially in the center to a randomly determined target cell. The method to measure task completion time was same as that of the menu interfaces.

### 5.3. Web browser interface

For all devices, there is a cursor on the screen and users control the cursor to select a target. The method to control the cursor by pointing devices is the same as for the horizontal menu, vertical menu, and grid interfaces. Users can use TouchRoller in the same manner as in the menu interfaces, but the cursor is visible in this case. In the case of a d-pad, the cursor moves in one of the four directions while a corresponding direction button is pressed.

The Web browser interface shows a Web page with a target link. A target is surrounded by a rectangle and an arrow is used to identify it, as shown in the Figure 6(d). In the user study described below, the Web browser showed Wikipedia pages and a participant selected a target link such as a picture or a word at an arbitrary position. The smallest target link was $42 \times 17$ pixels and the largest was $220 \times 311$ pixels.

## 6. USER STUDY

The goal of the user study was to experimentally verify the following two hypotheses:

- TouchRoller performs better than a pointing device for menu navigation tasks.
- TouchRoller performs better than directional keys for free pointing tasks.

To this end, a controlled experiment was conducted with three device conditions (including TouchRoller) and four types of GUI conditions (using the four prototype interfaces described in the previous section).

### 6.1. Devices

We had to choose a representative device for 2D pointing devices with an integral control structure and a continuous input property. Possible candidates were a mouse, touchpad, trackball, gyro-mouse, or an isometric joystick. A mouse was excluded because it is not suitable for a TV environment, and an isometric joystick was excluded because it was shown to be inferior to a gyro-mouse in an earlier study (MacKenzie and Jusoh, 2001). Further, a gyro-mouse was excluded because it is known to have problems such as fatigue, jittering, and response delay introduced by cursor stabilization, similarly to other laser-pointer style direct-pointing devices (Myers et al., 2002; Olsen and Nielsen, 2001). Between the remaining two choices, we selected a touchpad in preference to a trackball because more touchpad products are commercially available from major TV makers (e.g., Panasonic, Sony, and Samsung). In fact, remotes with touchpads have existed for a while (Enns and MacKenzie, 1998). Figure 7(a) shows the prototype touchpad that we built for the current user study. The size of the prototype touchpad device is $147 \times 50 \times 22$ mm. It has a touchpad with a touch area of $49 \times 65$ mm and a mouse button under the touchpad for target selection. The touchpad sends data to a PC via a serial port.

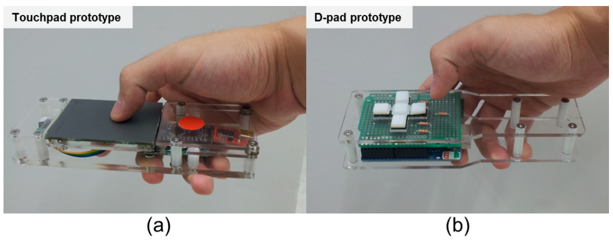

**Figure 7.** Devices for comparison: (a) touchpad prototype and (b) d-pad prototype.

Next, we needed to choose a representative device for the class of devices with a separable control structure and a discrete input property, so far referred to as directional keys. In fact, there are few different designs for directional keys. Some examples are the d-pad of a traditional remote, the directional buttons of a game controller, and the arrow keys of a computer keyboard. As we are interested in a TV remote control, we chose the d-pad of a TV remote. Figure 7(b) shows the prototype d-pad that we built for the current user study. It consists of four directional buttons and a selection button in the center. The button spacing (distance between the centers of adjacent buttons) is about 13 mm and they feel similar to the buttons of a common TV remote. The size of each button is $9 \times 9$ mm. The length and height of the device are 158 and 26 mm, respectively. The width of the upper part where the buttons are placed is 53 mm, and the width of the lower part that is used as a handle for the device is 42 mm, as shown in the Figure 7(b). The prototype sends data to the computer via a serial port (USB VCD), and therefore, there is no time delay, as is present in the case of a remote control that uses an infrared (IR) data link.

Unlike TouchRoller the touchpad prototype and the d-pad prototype have a rectangular shape instead of a triangular shape. The inverted triangular shape of TouchRoller was to make a user's thumb perpendicular to the roller. In the case of the touchpad and the d-pad, however, users do not need to maintain the thumb upright as much as in the case of rolling a roller. Instead, we thought that it would be more important to respect the current design that users are familiar with. Therefore, we selected a rectangular shape for both the touchpad and the d-pad because it is the most common form that we could see in the market.

### 6.2. Experiment

#### *6.2.1. Participants*

A total of 12 university students (6 males and 6 females, average age: 24.7 years) were recruited for this experiment. All students had more than one year of prior experience with a smartphone, which implies that the participants had sufficient prior experience using touch interfaces with their thumbs.

#### 6.2.2. *Tasks*

The horizontal menu, vertical menu, grid, and Web browser interfaces described in the previous sections correspond to the four GUI conditions. Participants were asked to select a target in each trial while using one of the four GUI conditions. The interfaces initially showed a pop-up dialog, and the trial started when the pop-up was dismissed. The trial time was measured from trial start to the moment when a target was finally selected. In the Web browser, the pop-up disappeared after the Web page was completely loaded and the trial time was measured from the moment when the cursor started to move.

#### 6.2.3. *Procedure*

The experiment was conducted in a controlled room that imitated television-watching environments using PCs and 22 inch LCD monitors (LG FLATRON W22671VZ). The aspect ratio was 16:9, and the resolution was 1920 × 1080 pixels. The size of a typical smart TV in homes is around 44 inch and the viewing distance is around 3 m. To create an equivalent viewing angle, we set the distance between the monitor and participant in our experiment to around 1.5 m.

Each participant performed tasks under the four GUI conditions with the three devices. To mitigate the carryover effect among devices, the order of the devices was counterbalanced among participants. However, the GUI conditions were always in the following interface order: horizontal menu, vertical menu, grid, and Web browser. The overall 3 × 4 factorial conditions were conducted in device-major order. For example, if a participant's device order was TouchRoller, d-pad, and then touchpad, he or she conducted the four GUI conditions with TouchRoller in the GUI order given above, then conducted the same four GUI conditions with a d-pad, and so on.

Three sessions were conducted for each condition such that each participant conducted a total of 36 sessions ($3\ sessions \times 3\ devices \times 4\ GUI\ conditions$), and the 36 sessions were distributed over three days to avoid fatigue. Each session consisted of a two-minute practice time and 30 task trials. On the last day, a NASA TLX survey was conducted to measure the participants' task load for each device and GUI condition combination.

### 6.3. Results

#### 6.3.1. *Task completion time*

The task completion time for each trial was measured during the entire sessions. The mean task completion time for each session was calculated, where the time wasted by failed trials was counted as penalty time, as shown in the following formula:

$$MeanTime = (T_s + T_f)/(N - n) \qquad (1)$$

where $N$ is the number of all trials in a session (360 in this experiment), $n$ is the number of failed trials, and $T_s$ and $T_f$ are the total task completion times of all succeeded and failed trials, respectively. Imposing a penalty in this manner does not reflect the additional time needed to recover from an erroneous result caused by an error in a real-world TV-watching situation, and therefore is a minimum penalty. Table 1 shows the number of errors and the mean task times for each session, and a statistical comparison of the devices is shown in Figure 8. As the mean task times decrease from the first to third session in all cases, the results of the final session alone were used for performance comparison among the three devices. Data are analyzed by non-parametric methods because Shapiro-Wilk test showed that the data from all conditions don't have normal distribution. Device condition had a significant main effect on the mean task times in all of the four GUI conditions (Friedman's ANOVA, $p < .0001$).

In the horizontal, vertical, and grid menu interfaces, the performances of TouchRoller were significantly better than the performances of the touchpad (Wilcoxon signed-rank test, $Z = -8.183$, $p < .0001$; $Z = -6.888$, $p < .0001$; $Z = -5.752$, $p < .0001$, respectively). In comparison to the d-pad, the performances of TouchRoller were significantly worse for the horizontal menu and grid interfaces ($Z = -5.312$, $p < .0001$; $Z = -6.360$, $p < .0001$, respectively). In the vertical menu interface, however, TouchRoller was significantly faster than d-pad ($Z = -3.095$, $p = .002$). In the Web browser interface, the touchpad had the best performance statistically. The performance of TouchRoller was the second best, and the performance of the d-pad was the worst. There were significant differences among the three devices. Figure 8 also shows the Wilcoxon signed-rank test results between pairs of the three devices.

#### 6.3.2. *NASA TLX*

To understand how much the devices induce user fatigue in the four GUI conditions, a NASA TLX (Hart and Staveland, 1988) survey was utilized. The scores on the NASA TLX survey were obtained from the average of the products' rating points and weight marks given by the participants. The lower scores for TouchRoller shown in Figure 9 indicate that TouchRoller caused less fatigue than the other devices. TouchRoller obtained a significantly better score than the touchpad in the horizontal menu interface (Wilcoxon test, $Z = -2.166$, $p < .05$). Moreover, the scores of TouchRoller were significantly lower than the d-pad and touchpad for the

| Horizontal menu interface | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 1st session | | | 2nd session | | | 3rd session | | |
| | NF | MT | SD | NF | MT | SD | NF | MT | SD |
| TR | 4 | 6277 | 2117 | 4 | 5393 | 1572 | 1 | 5124 | 1786 |
| DP | 2 | 4960 | 1311 | 4 | 4655 | 1257 | 3 | 4581 | 1183 |
| TP | 8 | 7167 | 1961 | 6 | 6388 | 1841 | 8 | 6292 | 1767 |
| **Vertical menu interface** | | | | | | | | | |
| | 1st session | | | 2nd session | | | 3rd session | | |
| | NF | MT | SD | NF | MT | SD | NF | MT | SD |
| TR | 4 | 5793 | 2044 | 4 | 5321 | 1751 | 5 | 5199 | 2175 |
| DP | 4 | 5627 | 1887 | 3 | 5663 | 1962 | 3 | 5478 | 1882 |
| TP | 11 | 6796 | 2005 | 10 | 6359 | 1902 | 21 | 6217 | 1622 |
| **Grid interface** | | | | | | | | | |
| | 1st session | | | 2nd session | | | 3rd session | | |
| | NF | MT | SD | NF | MT | SD | NF | MT | SD |
| TR | 0 | 1411 | 516 | 2 | 1349 | 542 | 2 | 1328 | 467 |
| DP | 3 | 1179 | 372 | 1 | 1138 | 392 | 3 | 1128 | 304 |
| TP | 12 | 1888 | 738 | 10 | 1834 | 741 | 8 | 1580 | 590 |
| **Web browsing interface** | | | | | | | | | |
| | 1st session | | | 2nd session | | | 3rd session | | |
| | NF | MT | SD | NF | MT | SD | NF | MT | SD |
| TR | 6 | 1941 | 814 | 4 | 1801 | 806 | 9 | 1783 | 751 |
| DP | 22 | 2436 | 1153 | 19 | 2226 | 1106 | 17 | 2116 | 1105 |
| TP | 17 | 1756 | 747 | 19 | 1849 | 615 | 11 | 1318 | 527 |

TR: TouchRoller, DP: d-pad, TP: Touchpad
NF: Number of failed trials, MT: Mean time (ms), SD: Standard deviation of MT

**Table 1.** Mean task completion time for each session.

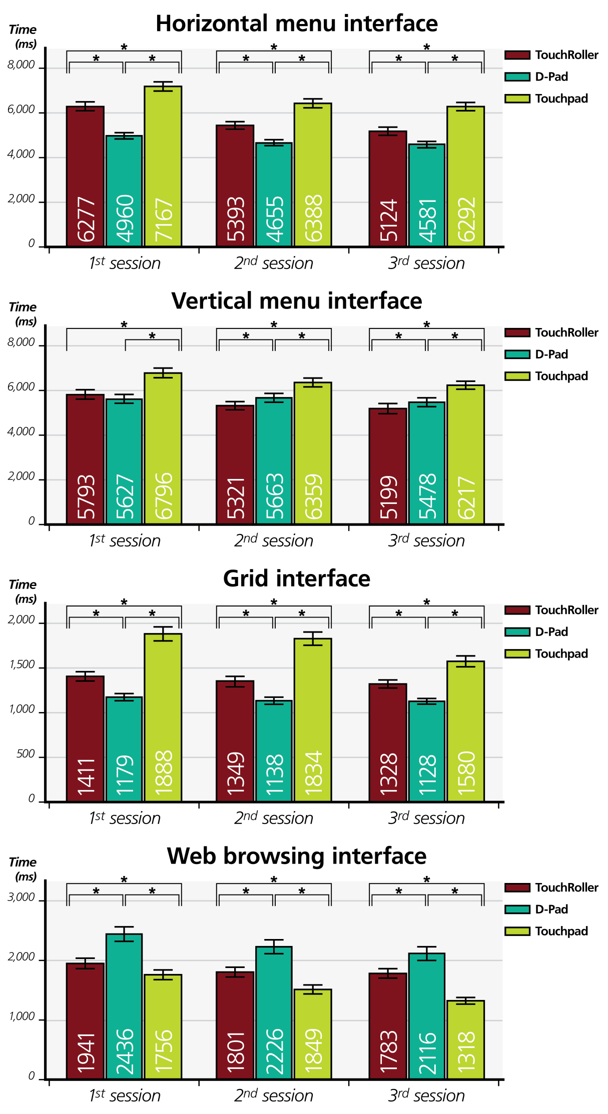

**Figure 8.** Mean task completion time for each device-GUI condition combination. Pairs marked with * have a significant difference.

vertical menu interface (Wilcoxon test, $Z = -2.627$, $p < .01$; $Z = -2.916$, $p < .01$, respectively) and for the grid interface (Wilcoxon test, $Z = -1.994$, $p < .05$; $Z = -3.668$, $p < .001$, respectively). There were no significant differences among the devices for the Web browser interface; however, the scores of TouchRoller were also relatively better than the other devices in this case.

## 7. DISCUSSION

### 7.1. Support of initial claims

We have experimentally shown that TouchRoller has a better performance than a pointing device for menu navigation tasks, as well as a better performance than directional keys for free pointing tasks. In addition to proving our main hypotheses, we now have data to support the claims we made during the development of the TouchRoller concept:

- Directional keys are effective for menu navigation tasks, but are not suitable for pointing tasks. On the other hand, pointing devices are effective for free pointing tasks, but do not perform well in menu navigation tasks.
- The control structure of TouchRoller is largely separate. It is possible to move the finger along the roller while rotating the roller, but the two operations are in general performed separately.

The data from the user experiments precisely supports the first claim. The d-pad was the fastest device in the horizontal menu, vertical menu, and grid interfaces, but was slowest in the Web browser interface. On the other hand, the performance of a pointing device was clearly better for free pointing tasks than for

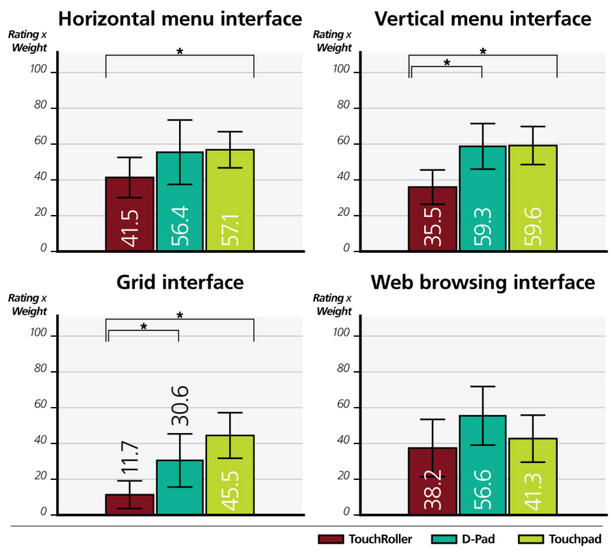

**Figure 9.** NASA TLX survey results. Pairs marked with * have significant differences.

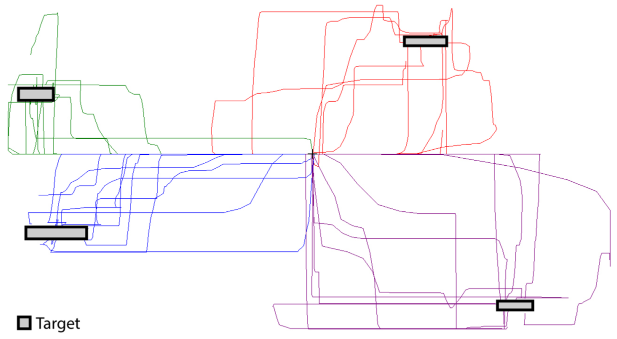

**Figure 10.** Trajectories of a pointer that participants moved with TouchRoller to select a target in each quadrant of the Web browser.

| | 1st session | 2nd session | 3rd session |
|---|---|---|---|
| TouchRoller | 89.30% | 89.66% | 88.83% |
| d-Pad | 100% | 100% | 100% |
| Touchpad | 31.08% | 30.72% | 29.01% |

**Table 2.** Rate of horizontal or vertical distance moved to total distance moved.

menu navigation tasks. The second claim cannot be supported by the experimental results that compare relative performances, but its evidence was clear in the user experiments. Figure 10 shows the trajectories of a pointer moved using TouchRoller. Participants tended to move a pointer horizontally or vertically when they were using TouchRoller in the Web browser. We quantitatively analyzed the trajectories of a pointer moved by the three devices in the experiment to confirm this claim. For each trajectory, the rate of the distance that a pointer moved horizontally or vertically relative to the total moving distance in each trial was computed. The movements of a pointer were regarded as horizontal, vertical, or diagonal movements based on the slope of their tangential lines. Sections that have slope angles relative to the positive x-axis within the range of $(-10, 10)$, $(80, 100)$, $(170, 190)$, and $(260, 280)$ degrees are considered horizontal or vertical movements. They are included in horizontal or vertical movement distances. Table 2 shows the rate that a pointer moved horizontally or vertically when the participants used the three devices. When the participants used TouchRoller, the pointer moved horizontally or vertically about 90% of the total distance moved in a trial. In contrast, the rate was only about 30% when the touchpad was used.

## 7.2. Ergonomic advantages of TouchRoller

The results of the NASA TLX survey show that the participants experienced a lower task workload with the TouchRoller for every GUI condition compared to the d-pad and touchpad. In particular, there were statistically significant differences in the vertical menu and grid interfaces. The two menu interfaces required the participant to make frequent flexion motions of the thumb. It is interesting that the TouchRoller imposes less workload on users in those tasks than the touchpad does. We attribute this NASA TLX result to the cylindrical form of the TouchRoller. When the participants performed the given tasks with the TouchRoller, they naturally indicated a vertical movement of the cursor by rolling the cylinder with the entire distal phalange of the thumb. However, the participants could indicate a vertical movement only with the tip of the thumb on the touchpad, requiring a larger flexion motion. The larger flexion motion of the thumb might have induced a greater task workload for the participants because they needed to hold the remote control with the remaining four fingers. In consequence, we suspect that TouchRoller could allow users to control GUI menus more comfortably in the smart TV environment.

## 7.3. Potential of the d-Pad

One of the unexpected findings in our study was that the d-pad has the potential to be a much better input device than we expected from our everyday experience with a d-pad remote control. In fact, there was difference between the d-pad in a TV remote and the d-pad that we used in our study.

The d-pad in a TV remote usually communicates with a TV by Consumer IR (CIR)[2], while the d-pad used in the current study communicated with the computer by serial communication. Among many standards for CIR, RC5 by Phillips is most common and popular. RC5 is based on modulated IR light and bi-phase signaling. The duration time of each bit is around 1.8 ms, and the total time of an RC5 code is around 24.8 ms. The space between two transmitted codes is 50 bit times or around 88.9 ms. An RC5 code is 14 bits long, so the effective baud rate of RC5 is around 120 baud. This low baud rate alone may have an impact on the performance of a d-pad, but the more serious problem of RC5 is that it sends the same code repeatedly (5 times) for every key press for the sake of robust communication. The result, however, is that the receiver should ignore subsequence RC5 codes for some period once it receives an RC5 code. As a result, a remote can send key press events no faster than one event per around 500 ms. In other words, the effective speed of Consumer IR is less than 30 baud.

On the other hand, the d-pad used in this study sends a single signal per each button press, allowing the TV to react right away to the quick repetition of button presses. The difference of the communication delay can make a significant difference. It was reasonable, however, for us not to consider this delay effect in our comparison study because our goal was not to compare specific product designs but to compare the effect of different input characteristics of devices.

To estimate the effect of the communication delay of CIR on the performance of a d-pad, we compared the following two cases: a d-pad using a serial link (115200 bps) and a d-pad using a simulated CIR link. In the latter case, we let the computer ignore signals from a d-pad for 300 ms once it receives a button event to simulate a CIR case. Ten participants (5 male, 5 female, average age: 24.4 years) were recruited. The same d-pad prototype hardware that was used in the main study was used again in both cases. Participants were asked to select a target in a grid layout 30 times in each case. The result showed that the d-pad in the CIR case was significantly slower than in the serial case ($p < .001$).

The result of the user study described in Figure 8 shows that the d-pad used in that study is similar to the d-pad without delay in this experiment. Therefore, the participants in the user study could use the d-pad better than a d-pad typically equipped in a remote control with a delay caused by Consumer IR.

[2]http://en.wikipedia.org/wiki/Consumer_IR

### 7.4. "Separable" touchpad

A touchpad may be modified to have a separable control structure. For example, we may rectify the touchpad output. In other words, we may make a displacement horizontal (vertical) if its horizontal (vertical) component is dominant. Another possibility is to add a rectangular texture to a touchpad surface that would encourage horizontal or vertical movements. In either case, the resulting touchpad would have a separable control structure and may exhibit performance advantages similar to that of the TouchRoller. This does not, however, invalidate the experimental results and conclusions of this paper. A touchpad in this paper is not a specific touchpad design but a generic touchpad, representative of input devices with an integral control structure and continuous inputs. The result of this paper may apply to such "separable" touchpads equally well, i.e., it may exhibit similar performance advantages that the TouchRoller exhibited because of its separable control structure. Hence, a "separable" touchpad may be worth further exploration in the future.

## 8. CONCLUSION

We proposed TouchRoller as an input device with both a separable control structure and continuous input property. We verified that TouchRoller performs well in both menu selection and free pointing by comparing its performance to directional keys and pointing devices. In particular, we showed that TouchRoller outperforms a touchpad in a menu selection task, and outperforms a d-pad in a pointing task. The result supports our initial expectation that an input device with a separable control structure and continuous input will support two major types of user interfaces in smart TVs.

The results of a NASA TLX test showed that participants felt less fatigue when using TouchRoller than when using other devices. We attribute the positive experimental results to the ergonomic advantages of TouchRoller. A smart TV controller needs to allow users to control the interfaces both efficiently and comfortably with its remote control. We expect that TouchRoller will be a viable option for a smart TV controller because of its ergonomic as well as performance advantages.

## ACKNOWLEDGEMENTS

(Removed for blind review)

## REFERENCES

Aoki, R., Ihara, M., Maeda, A., Kobayashi, M., and Kagami, S. (2011). Expanding kinds of gestures for hierarchical menu selection by unicursal gesture interface. *Consumer Electronics, IEEE Transactions on*, 57(2):731–737.

Bailly, G., Vo, D.-B., Lecolinet, E., and Guiard, Y. (2011). Gesture-aware remote controls: Guidelines and interaction technique. In *Proceedings of the 13th International Conference on Multimodal Interfaces*, ICMI '11, pages 263–270, New York, NY, USA. ACM.

Bernhaupt, R., Obrist, M., and Tscheligi, M. (2007). Usability and usage of iTV services: lessons learned in an Austrian field trial. *Computers in Entertainment*, 5(2)

Bohan, M., Slocum, J., Shaikh, A. D., and Chaparro, B. (2003). Examining first-time usage of the RollerMouse. *Usability News*

Casiez, G. and Chaillou, C. (2005). Effects of dof separation on elastic devices for the navigation in 3d virtual environments with force feedback. In *Eurohaptics Conference 2005 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems 2005. World Haptics 2005. First Joint*, pages 483–486. IEEE.

Casiez, G., Vogel, D., Balakrishnan, R., and Cockburn, A. (2008). The impact of control-display gain on user performance in pointing tasks. *Human Computer Interaction*, 23(3):215–250.

Cesar, P., Bulterman, D. C., and Jansen, A. (2008). Usages of the secondary screen in an interactive television environment: Control, enrich, share, and transfer television content. In *Changing television environments*, pages 168–177. Springer.

Chen, M.-y., Mummert, L., Pillai, P., Hauptmann, A., and Sukthankar, R. (2010). Controlling your TV with gestures. In *Proceedings of the International Conference on Multimedia Information Retrieval*, MIR '10, pages 405–408, New York, NY, USA. ACM.

Choi, S., Han, J., Lee, G., Lee, N., and Lee, W. (2011). RemoteTouch: touch-screen-like interaction in the TV viewing environment. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 393–402.

Cruickshank, L., Tsekleves, E., Whitham, R., Hill, A., and Kondo, K. (2007). Making interactive TV easier to use: Interface design for a second screen approach. *The Design Journal*, 10(3):41–53.

Dezfuli, N., Khalilbeigi, M., Huber, J., Müller, F., and Mühlhäuser, M. (2012). PalmRC: Imaginary palm-based remote control for eyes-free television interaction. In *Proceedings of the 10th European Conference on Interactive Tv and Video*, EuroiTV '12, pages 27–34, New York, NY, USA. ACM.

Dias, T., Variz, M., Jorge, P., and Jesus, R. (2013). Gesture interaction system for social web applications on smart TVs. In *Proceedings of the 10th Conference on Open Research Areas in Information Retrieval*, pages 225–226.

Enns, N. R. N. and MacKenzie, I. S. (1998). Touchpad-based remote control devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 229–230. ACM Press.

Fleury, A., Pedersen, J. S., Baunstrup, M., and Larsen, L. B. (2012). Interactive TV: Interaction and control in second-screen TV consumption. In *10th European Interactive TV Conference*, pages 104–107.

Freeman, W. T. and Weissman, C. D. (1995). Television control by hand gestures. In *International Workshop on Automatic Face and Gesture Recognition*, pages 179–183.

Handel, S. and Imai, S. (1972). The free classification of analyzable and unanalyzable stimuli. *Perception and Psychophysics*, 12(1):108–116.

Hart, S. G. and Staveland, L. E. (1988). Development of NASA-TLX (Task Load Index): results of empirical and theoretical research. In *Human Mental Workload*, volume 52 of *Advances in Psychology*, pages 139–183. North-Holland.

Ishiyama, K. and Yano, S. (2000). A study of characteristics of pointing devices for television operation. In *IEEE International Conference on Systems, Man, and Cybernetics*, volume 2, pages 1307–1312.

Jacob, R. J. K., Sibert, L. E., McFarlane, D. C., and Mullen, Jr., M. P. (1994). Integrality and separability of input devices. *ACM Transactions on Computer-Human Interaction*, 1(1):3–26.

Jones, E., Alexander, J., Andreou, A., Irani, P., and Subramanian, S. (2010). GesText: accelerometer-based gestural text-entry systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2173–2182. ACM.

Ju, G., Angiolillo, J., and Hicks, J. (1994). The challenges of designing a user interface for consumer interactive television. In *IEEE International Conference on Consumer Electronics - Digest of Technical Papers*, pages 114–115.

Kela, J., Korpipää, P., Mäntyjärvi, J., Kallio, S., Savino, G., Jozzo, L., and Marca, D. (2006). Accelerometer-based gesture control for a design environment. *Personal and Ubiquitous Computing*, 10(5):285–299.

Lee, G., Lee, S., Bang, W., and Kim, Y. (2011). A TV pointing device using led directivity. In *Consumer Electronics (ICCE), 2011 IEEE International Conference on*, pages 619–620. IEEE.

Lekakos, G., Chorianopoulos, K., and Spinellis, D. (2001). Information systems in the living room: A case study of personalized interactive TV design. In *Proceedings of the 9th European Conference on Information Systems*, pages 216–222.

Lin, C.-L., Hung, Y.-H., Chen, H.-Y., and Chu, S.-L. (2012). Content-aware smart remote control for android-based TV. In *Consumer Electronics (ICCE), 2012 IEEE International Conference on*, pages 678–679. IEEE.

Lu, K. Y. (2005). Interaction design principles for interactive television. Master's thesis, Georgia Institute of Technology.

MacKenzie, I. S. and Jusoh, S. (2001). An evaluation of two input devices for remote pointing. In *Engineering for Human-Computer Interaction*, volume 2254 of *Lecture Notes in Computer Science*, pages 235–250. Springer.

Martinet, A., Casiez, G., and Grisoni, L. (2012). Integrality and separability of multitouch interaction techniques in 3d manipulation tasks. *Visualization and Computer Graphics, IEEE Transactions on*, 18(3):369–380.

Morito, K. (1995). Remote control input device. US Patent 5,448,240.

Myers, B. A., Bhatnagar, R., Nichols, J., Peck, C. H., Kong, D., Miller, R., and Long, A. C. (2002). Interacting at a distance: measuring the performance of laser pointers and other devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 33–40.

Olsen, D. R. and Nielsen, T. (2001). Laser pointer interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 17–22.

Panasonic (2008). *Panasonic's EZ Touch Remote prototype.* Exhibited at CEATEC Japan 2008, Makuhari Messe, Japan.

Rashid, U., Nacenta, M. A., and Quigley, A. (2012). The cost of display switching: a comparison of mobile, large display and hybrid ui configurations. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pages 99–106. ACM.

Robertson, S., Wharton, C., Ashworth, C., and Franzke, M. (1996). Dual device user interface design: PDAs and interactive television. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 79–86. ACM Press.

Sohn, M. and Lee, G. (2004). Sonarpen: an ultrasonic pointing device for an interactive TV. *Consumer Electronics, IEEE Transactions on*, 50(2):413–419.

Stenger, B., Woodley, T., Kim, T.-K., and Cipolla, R. (2009). A vision-based system for display interaction. In *Proceedings of the 23rd British HCI Group Annual Conference on People and Computers: Celebrating People and Technology*, pages 163–168. British Computer Society.

Trusted Reviews (2013). Samsung 2013 smart TV platform - gesture and voice control, touchpad remote control and second screen support. In *Trusted Reviews*, http://www.trustedreviews.com.

Vatavu, R.-D. (2012). User-defined gestures for free-hand TV control. In *Proceedings of the 10th European Conference on Interactive TV and Video*, EuroiTV '12, pages 45–48, New York, NY, USA. ACM.

Vatavu, R.-D. (2013). There's a world outside your TV: exploring interactions beyond the physical TV screen. In *Proceedings of the 11th european conference on Interactive TV and video*, pages 143–152. ACM.

Vatavu, R.-D. and Zaiti, I.-A. (2014). Leap gestures for TV: Insights from an elicitation study. In *Proceedings of the 2014 ACM International Conference on Interactive Experiences for TV and Online Video*, TVX '14, pages 131–138, New York, NY, USA. ACM.

Wu, H. and Wang, J. (2012). User-defined body gestures for TV-based applications. In *Digital Home (ICDH), 2012 Fourth International Conference on*, pages 415–420. IEEE.

## APPENDIX: DEVICE PARAMETER SELECTION

### 8.1. CD gain of the touchpad prototype

Before the main experiments, we conducted a preliminary experiment to determine an optimal CD gain for the touchpad prototype.

We recruited five university students (4 males and 1 female, average age: 20.4) for the experiment. All of them were familiar with touchpad use. As the purpose of the main experiments was to compare performances in menu selection and free pointing, we used two corresponding tasks in this experiment. The first was a target selection task on a grid consisting of 200 × 200 pixel cells that represented a menu layout. The size of a cell was determined based on the size of an icon in commercially available smart TV. The second task was also a target selection task on a grid, but with a different cell size, $132 \times 28$ pixels. The size and aspect ratio of the cells were chosen to simulate a pointing task with a small target, such as a hyperlink on a Web page. The trial of each task started when a participant dismissed a pop-up and ended when they selected a target cell.

We pre-determined five CD gain values, as shown along the horizontal axis of Figure 11(a). The CD gain conditions were counterbalanced using a Latin square. Each participant performed the two tasks for the five CD gains, thereby completing 10 sessions. A session consisted of 30 trials, and therefore each participant carried out 300 trials in total. The same touchpad prototype used in the main experiment (Figure 7(a)) was used. The distance from a participant to the screen was around 3 m, and the screen size and resolution were 46 in and 1920 × 1080 pixels, respectively.

Experimental results for the touchpad are shown in Figure 11(a). The average task completion times tended to decrease as the CD gain decreased. However, when the CD gain was higher than 18 pixels/mm, the task completion times for different CD gain values were similar. On the other hand, the average number of errors increased rapidly when the CD gain value was increased from 18 to 24 and 30 pixels/mm. Based on these two curves, we decided to set the CD gain for the touchpad prototype to 18 pixels/mm.

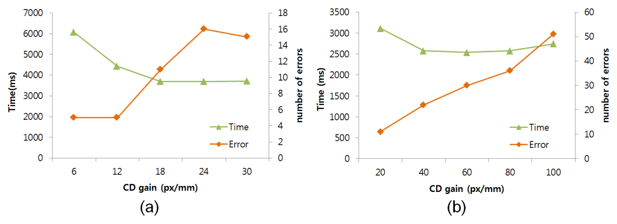

**Figure 11.** Average task completion times and the number of errors for different CD gains for (a) touchpad and (b) TouchRoller.

## 8.2. CD gain of the TouchRoller prototype

We repeated the experiment described above to determine an optimal CD gain for the TouchRoller prototype. Another group of five participants (5 males, average age: 21.4) were recruited. The same experimental tasks and procedure were used. The device was the first TouchRoller prototype described in Section 4.3. This prototype used a rotary encoder with discrete steps (24 detents per revolution), and hence, the vertical movement was not continuous. Therefore, the CD gain in this case was only for horizontal movement.

The five pre-determined CD gain values and corresponding experimental results are shown in Figure 11(b). The average task completion times (average among participants and tasks) for different CD gain values were similar. On the other hand, the number of errors increased rapidly as the CD gain increased. Based on these two curves, we decided to set the CD gain for TouchRoller to 40 pixels/mm.

We did not repeat the same experiment for the second TouchRoller prototype. Although the second prototype has a longer roller than the first, the CD gain determined for the first prototype was equally reasonable and therefore retained. In the second prototype, however, we also needed another CD gain for the vertical movement, since the roller rotation was now continuous. The vertical CD gain was later determined to be 8 pixels/mm, based on the result of a test carried out by the authors of this paper.

## 8.3. Cursor speed for the d-pad prototype

The d-pad is intended mainly for discrete action, however, it was also used to control cursor movement in the main experiments. When a user held down a direction key for a period of time, the cursor started to move continuously. The optimal cursor speed was determined to be 600 pixels/s, based on a previous study that used a d-pad in the same manner (Ishiyama and Yano, 2000). In fact, the cursor speed increased as the user continued to press a direction key. The acceleration curve for the cursor speed vs. holding time is shown in Figure 12(d).

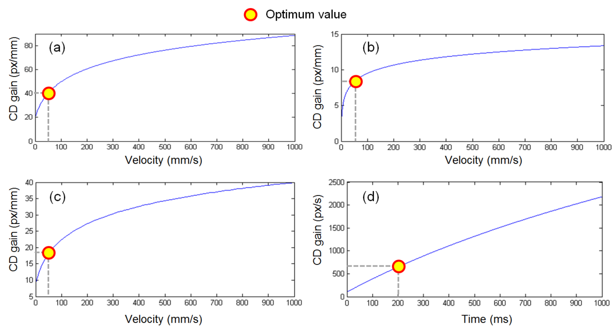

**Figure 12.** Acceleration curves for the three devices: (a) horizontal and (b) vertical TouchRoller controls, (c) touchpad, and (d) d-pad.

## 8.4. Pointer acceleration curves

Most pointing devices such as a mouse or touchpad use pointer acceleration to improve their pointing performance (Casiez et al., 2008). As users will be already accustomed to a touchpad with pointer acceleration, we thought it would be fair to implement pointing acceleration for the touchpad and all other devices as well.

The method for determining the curve of pointer acceleration varied according to device and operating system. In our implementation, we determined the curves for the three devices in the following way. First, we implemented a logarithmic function, as used for a mouse in Microsoft Windows (Casiez et al., 2008). We then allowed the curves to pass the optimum CD gain values when the device speed was 50 mm/s, a typical device speed. The optimum CD gain values are the experimentally determined gains from the previous sections. In the case of the d-pad, a curve was determined such that the cursor speed reached an optimal value, 600 pixels/s, when a user held down a button for around 200 ms. Figure 12 shows the resulting acceleration curves of the three devices.