

# TDome: A Touch-Enabled 6DOF Interactive Device for Multi-Display Environments

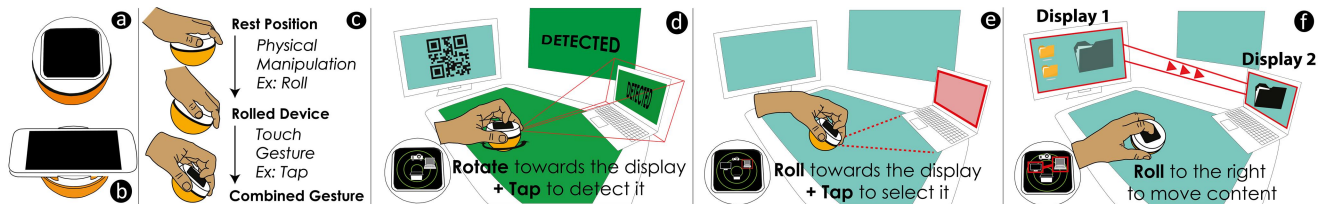
Houssem Saidi<sup>1</sup>, Marcos Serrano<sup>1</sup>, Pourang Irani<sup>2</sup>, Emmanuel Dubois<sup>1</sup>

<sup>1</sup> University of Toulouse – IRIT  
Toulouse, France

{first\_name.last\_name}@irit.fr

<sup>2</sup> University of Manitoba  
Winnipeg, Canada

Irani@cs.umanitoba.ca



**Figure 1.** TDome combines a small (a) or large (b) touchscreen with a dome-like mouse [23]. TDome supports performing combined gestures (c), i.e. a 6 DOF physical manipulation followed by a touch input. TDome facilitates several common tasks in MDEs, such as d) display registration using its embedded camera, e) device selection and f) cross-display data transfer.

## ABSTRACT

The rapid evolution of multi-display environments (MDEs) has created a vacuum in need of novel input devices to optimize interaction in MDEs. In this paper, we propose TDome, a novel touch-enabled 6DOF input and output device to facilitate interactions in MDEs. TDome offers a private display as output, and multiple degrees of freedom as input by combining touch gestures on the display with physical rotation, roll and translation manipulations of the device. TDome allows versatile interactions that address major MDE tasks, which we illustrate through various proof-of-concept implementations: detect surrounding displays, select one display, transfer data across displays, reach distant displays and perform private interactions. We explore TDome's usability and suitability for MDEs through three user studies. First we explore combined *physical+touch* gestures from which we discard uncomfortable combinations. We experimentally validate their feasibility and come up with a set of 71 combined gestures that can be easily performed and efficiently detected. Finally, we collect user feedback to identify natural mappings between gestures and MDE interactions.

## Author Keywords

Multi-display environments; input device; touch input; rolling device.

## ACM Classification Keywords

H.5.2. Information interfaces and presentation (e.g., HCI): Interaction.

## INTRODUCTION

Multi-displays environments (MDEs), combining vertical and horizontal displays of varying shapes and forms, have shown significant value for interacting with heterogeneous data sources and in multiple contexts such as 3D exploration [7], collaborative scenarios [12], crisis management [14] and scientific data visualisation [33]. MDEs offer numerous advantages for organizing information across displays, for enhancing individual and group work, for providing support to peripheral information and for extending the interaction space. The incidental emergence of MDEs has resulted in a device vacuum: to our knowledge no device has been specifically implemented for optimizing interactions in such spaces.

Researchers have mainly proposed adapting existing devices to tackle individual MDE tasks, such as the mouse for multi-monitor pointing [5], or smartphones for cross-display data-transfer or distant pointing [11,22]. However such adaptations can result in undesirable side effects: mice are not appropriate when the user is standing [22] and smartphones held in mid-air can be tiring and cumbersome for long interactions [16]. Recent research has demonstrated the use of wearable devices to perform cross-device interactions [18,32]. However, current wearables lack proper input mechanisms and mainly serve private purposes. If MDEs are to become the office of the future, as envisioned by many [26,28], can we design a device specifically tuned for such an environment? Adopting a unique device would avoid the homing effect when switching from one device to another, enhance privacy in such environments through personal data control and visualization, lead to a coherent set of interactions with the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [Permissions@acm.org](mailto:Permissions@acm.org).

CHI 2017, May 06–11, 2017, Denver, CO, USA

© 2017 ACM. ISBN 978-1-4503-4655-9/17/05...\$15.00

DOI: <http://dx.doi.org/10.1145/3025453.3025661>

varied MDE applications, and ultimately contribute to a more fluid task flow, a key element in MDEs [4].

To this end we present a novel touch-enabled device, TDome, designed to facilitate interactions and address a range of tasks in MDEs [9,32]. TDome is the combination of a touchscreen, a dome-like Mouse [23] providing 6 DOF, and a camera that can sense the environment. TDome thus inherits properties of other existing mouse-like devices but includes many novel features to tackle the needs of common MDE tasks [9,32]: TDome identifies the spatial layout of displays (Figure 1-d); facilitates distant interaction and data transfer across displays (Figure 1-ef); and enables personal interactions by using the touchscreen as a private output medium. In our work we designed and implemented different techniques employing two versions of TDome (small and large touchscreen) to address these MDE tasks.

In this paper we address two major challenges for applying TDome in MDEs: first, the device's usability, which demands the user to coordinate a physical manipulation with a touch gesture (we refer to as combined gestures - see Figure 1-c); second, the mapping between TDome gestures and MDE tasks. To validate TDome's usability and suitability for MDEs, we conducted three user studies. We first carry out a formative study to discard gestures deemed too uncomfortable. We followed this with a controlled system validation in which we identified the success rate and performance of combined gestures. Results show that users can comfortably and precisely perform 54 combined gestures with TDome when it is equipped with a large touchscreen and 17 when it has a small touchscreen. Finally, using this set of gestures, we collected user feedback on the best mappings from TDome gestures to common MDE tasks. Results reveal that users take benefit of the semantics of TDome gestures to easily map them to certain tasks, such as rolling the device to view private information, or performing a Drag on the touchscreen to send content across displays.

We offer two contributions. (i) We introduce TDome, a novel device aimed directly at facilitating MDE interactions and discuss its properties with regards to MDE common tasks. (ii) We introduce a diverse range of MDE interaction possibilities with our prototypes. To validate these contributions, we identify the most usable TDome interactions. We then obtain qualitative feedback on mappings from TDome features to MDE tasks.

## RELATED WORK

We present a range of techniques designed to support interaction in MDEs. In most cases, a new technique was implemented using an already existing device.

### Early multi-display interaction techniques

While mouse input is suited for interactions with multiple desktop monitors [5], such a device does not adapt well to multi-display environments (MDEs), where displays may be scattered within the physical space [34]. One of the

earliest solutions, Pick'n Drop [27], proposes the use of a stylus to transfer information from one device to another. The stylus has also been used for pointing [20], a particularly difficult task when displays are large or far from the user. However, due to the limited DOF of such devices, these solutions focus on specific interaction techniques but do not cover the broad set of MDE tasks.

### Adapting mobile and wearable devices for MDEs

Performing physical gestures with mobile devices leverages significantly more DOF than those available with existing devices, such as mice. The main reason is that such devices combine a number of sensors that expand the input/output space (e.g. touch, tilt). Examples include the use of mobiles for pointing [22], for continuous map navigation [10,21], for copy-and-paste operations [30] or for interacting with distant 3D content [8]. The camera on the mobile device has been used to transfer data between MDE displays [11,13]. Another common approach is to use mobile devices for multi-display overview+detail tasks [3,7]. However, holding a mobile device in mid-air can be tiring [16], especially as mid-air manipulations lack precision. To overcome this problem, mobile devices can be actuated [31,19], which poses new limitations (robustness, speed).

Researchers have recently proposed the use of wearable devices for MDE operations, such as data transfer [18, 32]. For instance, Gluey [32] is a user interface based on the combination of a head-worn-display with a camera, which facilitates seamless input transitions and data movement across displays. Such devices offer mostly personal capabilities in MDEs.

### Roly-Poly Mouse

A number of multi-DOF input devices have been proposed in the literature. Although they were not specifically designed in the context of MDEs, their capabilities are related to those of TDome. TDome's rounded shape allows tilting the device, similar to the Rockin' Mouse [2] or the VideoMouse [17], and is directly inspired by a more recent device, the Roly-Poly Mouse (RPM) [23]. Using a completely rounded bottom as RPM has been shown to provide larger amplitude of movement than previous tilting devices, and it also enables compound gestures (see Table 1 in [23] for a summary on the differences between RPM and previous tilting and multi-DOF mice).

While the rounded dome-like shape of RPM offers multiple degrees of freedom, it hinders the device's stability. Unintended physical manipulations (e.g. Roll during Translation) are common on devices with such a form factor (cf. study 1 in [23]). In our work we correct this problem by combining physical manipulations with touch gestures, to ensure the robustness of TDome interactions.

The use of a touchscreen coupled with an input device is an approach already adopted in the design of the LensMouse [36]. Other mice have also proposed the use of multi-touch [6]. However, all of them coupled a 2D mouse with a

touchscreen, while the TDome uses a 6D mouse. This brings a new challenge in terms of device usability that we address in this paper through two user studies.

### TDOME OVERVIEW

TDome is a touch-enabled multi-DOF input device that embodies features and a form factor that facilitate MDE interactive tasks. This unique device results from the composition of a touchscreen with a dome-like Mouse, providing rotation, Roll, Translation and Lift-Up motions (6 DOF). The device also includes a camera that can sense the environment. We present an illustrative usage scenario with TDome prior to presenting its features.

### Usage scenario

Harry is an engineer working on a smart campus project that monitors data collected by multiple sensors on the university. To visualize and interact with the large datasets of energy consumption, the university has set up a multi-display environment composed of several displays, a large projection wall and two TDome devices.

As Harry enters the room to start his daily supervision of the energy data, he grabs one TDome and uses it to initialize the multi-display environment by simply pointing at each active display. He then selects the wall projection by rolling the device toward the wall. Harry decides to Spread the data visualisation across two displays: he selects both displays with TDome and transfers the visualizations from one to the other with a TDome gesture. As he wants to look closer at information on the second display, he grabs TDome and walks towards the display, using the device in mid-air to perform a zoom on the data for a closer look.

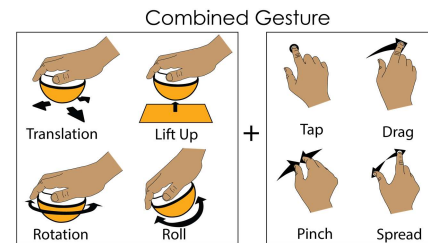
Later that day, Mary enters the room and grabs the second TDome. They have a meeting to explore the university map to mark points of interest. Harry and Mary take their personal smartphones and couple them with each TDome to benefit from personal interactions. Each smartphone shows a personal view of the projected map, which allows them to add and access personal annotations. Before ending, Harry wants to log onto the campus website and upload his annotations: he rolls TDome towards himself to display a virtual keyboard on the device's touchscreen and enter his personal password discreetly on the login page, displayed on the tabletop.

This scenario illustrates how TDome allows users to detect surroundings displays arrangement, select one display, move content between displays, reach to content at distant displays and perform personal interactions on TDome.

### Device Manipulation

Interacting with TDome requires the explicit combination of a physical manipulation with a tactile gesture on the touchscreen. The sequential combination of both actions acts as a delimiter whose accidental activation is unlikely, as demonstrated in our controlled evaluation (below). This approach reduces the risk of issuing a command after performing a physical manipulation inadvertently and

improves the robustness of the device. As illustrated in Figure 2, four different physical manipulations (Translations, Roll, Rotation and Lift-Up) can be combined with four different touch gestures (Tap, Drag, Pinch, Spread) for using TDome.



**Figure 2.** TDome allows performing combined gestures, i.e. a physical manipulation followed by a touch gesture

Initially, we favoured a one-handed interaction where the dominant hand was used to perform the physical manipulation on the device and the touch gestures on the display. But our preliminary tests revealed that some gestures were easier to perform in a bimanual mode, thus extending the touch vocabulary.

### TDome versions: small and large touchscreen

We implemented two design variations of TDome resulting from different device composition alternatives [24]: one with a small touchscreen inserted into the spherical shell (Small version) and one with a larger touchscreen laid on top of the spherical shell (Large version). As these two versions were meant to be complementary, we favoured the possibility of rapidly switching them as opposed to having two separate devices. This opens interesting possibilities, such as switching to the large touchscreen when a larger display is needed.

### IMPLEMENTATION

#### TDome core elements

We used the RPMouse [23] design guidelines to define the dimension of our device: a diameter of 8cm (~ 3.15in) was the easiest to handle and manipulate. As with the original RPMouse, we weighted the device with putty so that the device returns to its initial upright position when released (roly-poly toy principle).

Regarding the touchscreens, we implemented both the Small and Large versions. To restrict our device to the selected size, we had to limit the small screen size to less than 8cm. To create the Small version, we removed the bracelet from an Android smartwatch SimValley AW-414 (63g, 45x44x14mm, 28x28mm touchscreen) and enclosed the smartwatch into TDome. To implement the Large version, we used a Galaxy S4 smartphone (5 in, 134g, 137x70x8mm). We used the smartwatch's camera, which is situated on the edge of the watch, to provide TDome with a horizontal camera view. The camera has a 3MP sensor and a resolution of 1728x1728 pixels.

To support device modularity, the interchange of both touchscreens had to be easy and quick. We thus 3D printed two plastic adaptors that can be adjusted on a 3D printed base: the first one holds the watch while the second one fixes the phone using a magnet (Figure 3). The two plastic adaptors are very rapidly interchangeable. Altogether, the Small version, involving a smartwatch, weighted 207g in total and the Large version, involving a smartphone, weighted 297g. We used TCP sockets over a local Wifi network to connect the watch to the main computer.



**Figure 3. Arrangement of TDome elements for the Small version (left). Both TDome versions are rapidly interchangeable (right).**

### Physical manipulation detection

The spherical shell holds an x-IMU of x-io Technologies (48g,  $57 \times 38 \times 21$  mm) to detect the Roll and Rotation of the device in 3D. The IMU is composed of a triple-axis gyroscope, accelerometer and magnetometer. The refresh rate of the sensors goes up to 512Hz and we used Bluetooth to connect the IMU with the computer. The IMU offered an angular precision of  $1^\circ$ . We 3D printed a holder to fit the IMU in a horizontal position inside the TDome (Figure 3).

To detect the displacement of the device, we used an infrared bezel (Zaagtech, 42") that generated TUIO events. We implemented a filtering process to discard touch events that were detected when fingers touched the surface around the device. Thresholds were also empirically defined to avoid the detection of unwanted Translations, Rolls or Rotations: user's physical manipulations must reach a minimum amplitude to be detected (5cm for Translation,  $30^\circ$  for Roll,  $45^\circ$  for Rotation). Lift-Up was detected as soon as TDome was no longer in contact with the table.

### SUITABILITY OF TDOME FOR MDES

One of the key characteristics of Multi-Display Environments (MDEs) is their heterogeneous nature. For the majority, MDEs are composed of displays offering different properties (size, orientation, input capabilities), which have led to a set of major interaction requirements [9,32]: *input redirection* (i.e. redirect input channels to different displays), *output redirection* (i.e. move content between displays), *physical relationship* (i.e. possess high-level information on the spatial layout of the displays), *reachability* (i.e. interact with a distant display) and *personal data management* (i.e. personal input and output interaction). TDome properties suit these interaction requirements.

### Spatial sensing

TDome physical manipulations allow performing 3D pointing in the surrounding space. Combined with the on-board camera, it allows sensing the environment. This can be used to detect and locate nearby displays, creating a spatial layout of the MDE displays represented through a radar-view (*physical relationship*).

### Input interaction

TDome allows up to 3 types of 2D pointing: by moving the device, by rolling it or by interacting with the touchscreen. These ranges of positioning facilitate *input redirection*. This also offers input that best suits a given display, such as a cursor for precise tasks, or touch input for coarser input.

### Output interaction

The touchscreen display can be used as a visual buffer to move data among displays in MDEs (*output redirection*). It may also be useful to display a zoomed-in version of a selected area on a distant display (*reachability*). The built-in vibratory capabilities are an alternative to discretely provide the user with private information (*personal data management*).

Through the easy interchange of the Small and Large TDome versions, the user can adopt the most appropriate display for each task; e.g., to visualize large graphs, the user can choose the Large version, but to display the compact radar-view (i.e. a view of the MDE spatial layout), a smaller display is more appropriate (*output redirection*).

### Mid-air interaction

Two of TDome's physical manipulations (Roll and Rotate) can be used in mid-air, thus facilitating physical displacements to interact with distant displays (*reachability*). It also offers more flexibility to the user to ensure the privacy for some of its tasks (*personal data management*).

### Form factor

TDome's tilting capabilities facilitate orienting the device towards oneself for private input and output interaction (*personal data management*); and attaching their personal smartphone to TDome's base allows users to access their personal applications and data (*personal data management*).

### TDOME MDE INTERACTION TECHNIQUES

We now introduce a set of proof-of-concept prototypes illustrating how the previous properties contribute to facilitate interaction in MDEs.

### Display layout registration

To fulfil the physical relationship and arrangement requirement, we implemented a semi-automatic acquisition of the displays layout in the MDE. This technique allows detecting the displays and building a radar view interface of them, which can be later exploited to interact with the displays of the environment.



During the detection phase, TDome detects a QR code ascribed to each display (better recognition algorithms may not necessitate codes for detection as demonstrated by HuddleLamp [25]). The user orients TDome toward each display successively, so that the device's on-board camera detects the QR codes (Figure 4 – left). Once the QR code is recognized, the user taps the touchscreen to terminate the identification: the detected display is assigned a position in the environment thanks to the incorporated IMU.



**Figure 4:** TDome's on-board camera detects displays (left) and creates a radar-view of the spatial layout (center). Then the user can select a display by Rolling + Tapping towards it (right).

The user progressively creates a radar view describing the relative position of all detected displays, with TDome in its center (Figure 4 – center). The user can manually adjust the distance of each display to TDome on the radar view. Once created, the radar view can be used with a Roll + Tap on TDome to select a specific display, by rolling TDome in the direction of the display and tapping on the touchscreen to validate (Figure 4 – right).

#### Input redirection

One recurrent need in MDEs is to manage input redirection. In addition to changing focus from one display to another, TDome offers an input interaction that matches the input possibility to the display it is connected to. TDome can be used as a touch input device through its embedded touchscreen, as a mouse with its translation capability and as a 3D Mouse with its rotation and tilting capabilities depending on the input capability of the display it is redirected to.

For instance, to interact with a map on a distant touchscreen, the user can perform a Roll + Drag on TDome to pan, and a Lift-Up + Pinch on TDome to zoom. Both touchscreen gestures (Drag and Pinch) are the same than what would be used on the distant touch display. While using only TDome's touchscreen gestures would be possible, using them in combination with the physical gestures (Roll or Translate) ensures a high recognition rate, prevents false positives, as demonstrated by our controlled study presented below and offers additional controls: the Roll angle may impact the panning speed.

#### Moving content between displays

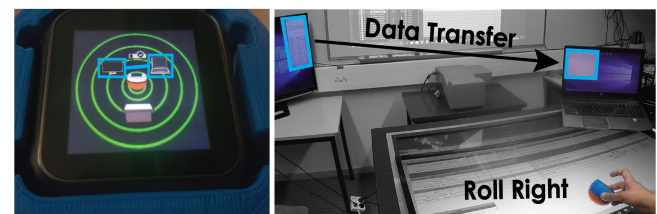
We developed two interaction techniques to move content from one display to another. The Translation + Pinch/Spread technique combines a physical manipulation of TDome to select a display and a gesture on the

touchscreen to grab or place some content on the selected display (Figure 5). In our implementation, Translation + Pinch grabs the application of the screen selected by the translation's direction, and displays it on the tabletop; while Translation + Spread sends the tabletop application to the screen situated in the translation's direction. Usually, Pinch and Spread are two well-established gestures for zooming in map or photo applications. Therefore, we plan to design appropriate feedback to avoid any confusion.



**Figure 5:** A Translation + Spread gesture sends the tabletop content (left) to a secondary display (right).

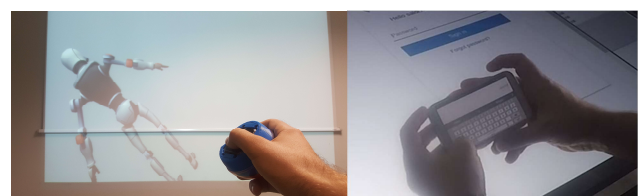
We implemented a second technique using the radar view on TDome to create a virtual information tunnel between two displays (Figure 6-left). The user creates the tunnel by sequentially selecting two displays on the radar view. Once the tunnel is defined, the user can move content along the tunnel with a Roll + Tap on TDome: rolling is performed in the spatial direction of the second display (i.e. a Roll to the right if the display is on the right of the first one); a Tap gesture finalizes the transfer.



**Figure 6:** Using the virtual tunnel technique to transfer information between displays.

#### Mid-air interaction

To support the reachability requirement, TDome provides support to interact with distant displays, i.e. beyond the user's reach. Given the size, shape and wireless design of TDome, the user can physically move to the distant display and perform mid-air interactions with TDome (Figure 7 - left).



**Figure 7:** Illustrating mid-air interaction with TDome (left) and privacy conservation when typing a password (right).

#### Password input

To preserve confidential information, the user can roll the device towards himself or lift the device to visualize and

input content privately. For instance, TDome's large touchscreen can be used as a private virtual keyboard to input a password on a surrounding display (Figure 7 – right). TDome can also be used to visualize a private detailed view of a public context.

### Other techniques

Beyond effectively supporting essential interactions in MDE, TDome can be used for other common tasks such as controlling a pie menu on a distant display, supporting multi-clipboard copy and paste, and pointing on distant displays. We implemented all these interaction techniques using different combinations of physical manipulations and touch gestures.

### Resulting challenges for TDome usage

These techniques illustrate how TDome contributes to the execution of relevant interactive situations in MDEs and how it is useful and sufficient to address major MDE interactions. Using a single device contributes to a more fluid interaction in MDEs by maintaining the user in the flow of her activity [4].

Informal tests also provided some early feedback on the importance of precision and on the required number of available gestures: a precise control of the device is important to perform spatial interactions, such as rolling to select a display; and the user requires a wide set of gestures to cover the multiple set of controls and interactions across displays. Therefore, conferring the highest usability level to TDome is essential to ensure MDEs will take full advantage of the device properties.

For these reasons, we first focused on exploring the usability of the device itself. To this end we performed a user experiment dedicated to identifying the set of most precise and robust TDome gestures.

### EXPLORATORY STUDY ON TDOME GESTURES

The goal of this exploratory study was to inform the implementation of input gestures combining physical manipulations with touch input, by studying only their comfort and collecting initial user feedback. Ultimately we wanted to discard gestures that would be deemed too uncomfortable. While literature on physiology could be anticipatory, it would not help in identifying all the appropriate combinations of wrist gestures and multi-touch finger input. For this reason, we did not want to discard any gesture immediately and ran this exploratory study to reduce the initial gesture design space.

#### Protocol

We carried this exploratory study with 4 participants (all right-handed) from the local university. We instructed participants to manipulate the TDome with their dominant hand. During the experiment they were confronted with the two different versions of TDome (Small and Large). In both settings, they tested three physical manipulations (Roll and Translate in 8 different directions, Rotate in two directions, Lift-Up) in combination with four touch gestures (Tap,

Drag, Pinch, and Spread). Pinch and Spread gestures being more complex to perform, participants repeated these gestures twice: once with the dominant hand and once with the non-dominant hand (e.g. in a bi-manual setting).

Participants performed 2 TDome versions  $\times$  19 physical manipulations  $\times$  6 touch gestures = 228 combined gestures per participant. We asked participants to repeat each combined gesture 3 times, i.e. each participant performed 684 trials. We asked them to rate each gesture combination from 1 (comfortable) to 5 (uncomfortable) to help them verbalize their opinion and comment on their ratings. We report on their qualitative comments.

### Results

Participants were very positive about performing the following gestures both with the Small and Large versions of TDome:

- Tap and Drag combined with any physical manipulation (Translation, Rotation, Roll or Lift-Up).
- Pinch and Spread in a bi-manual setting (one hand manipulates the rolling device while the other touches the display) when combined with a Translation, Rotation or Lift-Up.

However some other gestures seemed too uncomfortable to be performed:

- Performing Pinch and Spread with a single hand was always deemed very uncomfortable when combined with any physical gestures and for both TDome versions (Small and Large).
- Performing Pinch and Spread in a bi-manual setting in combination with a Roll gesture was perceived to be very uncomfortable.

We decided to remove these uncomfortable gestures (Pinch and Spread with a single hand or in combination with Roll) from our subsequent work.

### CONTROLLED EXPERIMENT

The goal of this controlled experiment was to validate the feasibility of combined gestures, i.e. physical manipulation + touch input. We hypothesize that certain touch gestures could be difficult to perform on the Small version, on which certain combinations could lead to errors.

#### Combined Gestures

From the previous exploratory study, we decided to use two touch gestures with one hand: Tap and Drag. Gestures using two fingers, i.e. Spread and Pinch, were performed with two hands: one hand held the device while the other performed the touch gesture. These touch gestures were used in combination with a Translation, a Roll, a Rotation and a Lift-Up of TDome.

#### Task

Participants were required to perform each gesture, following a visual indication on a tabletop display. TDome was placed in an initial position at the center of the surface,

indicated by a visual feedback. We let users hold the device as they pleased. We asked participants to perform the gestures as fast as possible with high accuracy. We provided continuous visual feedback indicating the state of the device (position, Roll and Rotation) as well as touch gestures on the display. We provided them with knowledge of result and in case of error we indicated which gesture (physical manipulation and/or touch gesture) had been erroneously performed. Each trial started when the user pressed a button on the tabletop, which displayed the instructions, and ended when a combined gesture had been recognised.

### Participants

We recruited 12 participants (3 female), aged 27.5 years on average ( $SD=4.89$ ) from the local university. 11 of them were right-handed and 3 of them took part in the exploratory study.

### Apparatus

We used the TDome implementation described earlier. The device was used on a tabletop display ( $96 \times 72$ cm) of 102 cm height thus requiring the user to stand during the experiment. We used the display in an area limited to the size of the infrared bezel (42 inches,  $1920 \times 1080$ px).

### Design and protocol

The experiment followed a  $2 \times 4 \times 4$  within-subjects design, with Display (Small, Large), Physical manipulation (Translate, Roll, Rotate and Lift-Up) and Touch gesture (Tap, Drag, Pinch and Spread) as factors. We did not test the condition combining Roll with Pinch/Spread, as this combination appeared to be highly uncomfortable in our pre-study. We also decided to study one random translation direction to limit the experiment length: previous studies on RPM [36] showed that all translation directions were as easy to perform. For the other physical manipulations, participants performed eight Roll directions and two Rotations (left/right).

Our pre-study also showed that Pinch and Spread gestures seemed more difficult than Tap and Drag. Therefore, we paired Tap with Pinch, and Drag with Spread to balance the different blocks length and difficulty. Trials were grouped in four blocks: one block corresponded to one Display and two touch gestures (Tap/Pinch or Drag/Spread).

The four blocks were counterbalanced across participants using a  $4 \times 4$  Latin Square. For each block, we ordered touch by difficulty: first Tap or Drag, then Pinch or Spread. For each set of trials corresponding to one touch gesture, the physical manipulations were ordered in a predefined way (Lift-Up, Translation, Roll and Rotation) because a random sorting would have made the instructions difficult to follow. Each combined gesture was repeated three times. Completing the four blocks took approximately 25 minutes.

The study started with a training set made of the same four blocks than in the experiment. The training consisted of 94

trials and took approximately 20 minutes. After the training, each participant performed 192 trials:

- 144 trials for the Tap and Drag: 2 Displays  $\times$  12 Physical Manipulations (1 Translation + 8 Rolls + 2 Rotations + 1 Lift-Up)  $\times$  2 Touch gestures  $\times$  3 repetitions.
- 48 trials for the Pinch and Spread: 2 Displays  $\times$  4 Physical Manipulations (1 Translation + 2 Rotations + 1 Lift-Up)  $\times$  2 Touch gestures  $\times$  3 repetitions.

We collected  $192 \times 12$  participants = 2304 trials in total, which took approximately 45 minutes for each participant.

### Collected Data

We logged all gestures from start to finish. We calculated success rates, completion time from instruction onset to validation, unintended touch gestures on the Display and amplitude of the physical manipulations. We classified errors in three categories according to the gesture that had been erroneously performed: physical, touch or both. Finally, we asked participants to rate each condition on a 1-5 Likert scale on perceived difficulty.

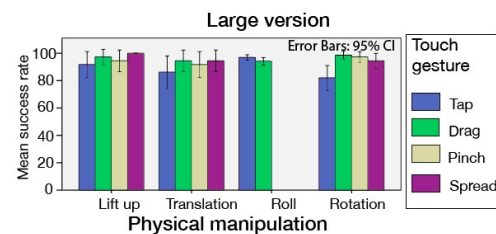
### RESULTS

A Shapiro-Wilk test established that the data was not normal and we could not normalize it. Therefore we used a Friedman test (we report  $\chi^2$  and  $p$ ) to compare more than 2 conditions, and Wilcoxon tests otherwise (we report  $p$  value). Where appropriate, we used a Bonferroni correction. We first discuss the success rate for the Small and Large versions separately as a Wilcoxon test showed a significant effect of Display on success rate ( $p < .001$ ).

#### Success rate: Large version

When using the Large version, we found no significant effect of Touch gestures (Friedman:  $\chi^2=3.87$ ,  $p=0.2$ ) or Physical manipulations (Friedman:  $\chi^2=4.1$ ,  $p=0.2$ ) on success rate. Overall, success rate with the Large version was 94.44%. Errors were distributed among Physical Manipulations (2.52%) and Touch gestures (2.86%).

A Friedman test reveals a significant effect of Touch gesture on success rate when performing a Rotation ( $\chi^2=13.32$ ,  $p=.003$ ): a Wilcoxon test reveals a significant difference between Tap and Drag (81.94% vs. 98.61%;  $p=.022$ ) and between Tap and Pinch (81.94% vs. 97.22%;  $p=.045$ ).

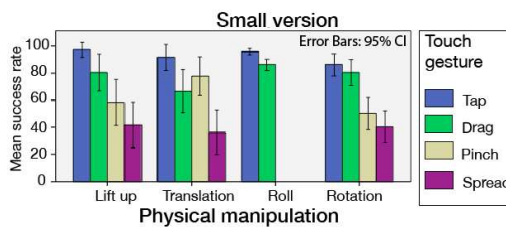


**Figure 8.** Mean success rate for each combination of Physical manipulation and Touch gesture when using the Large version.

We observed that when instructing participants to perform a Rotation + Tap combined gesture, 91% of the erroneously detected touch gestures are Drag gestures. Performing a Rotation induces a wrist distortion that may affect the user's ability to precisely tap the display without swiping the finger: this may explain why a Drag is easier to perform than a Tap. Spread and Pinch are not affected by the wrist rotation since they are performed in a bi-manual setting (Figure 8).

#### Success rate: Small version

When using the Small version, a Friedman test revealed a significant effect of Physical Manipulation ( $\chi^2=17.46$ ,  $p<.001$ ) and Touch gestures ( $\chi^2=33.56$ ,  $p<.001$ ) on success rate. We analyse the success rates for each combined gesture, i.e. the combined Physical manipulation and Touch gesture.



**Figure 9. Mean success rate for each combination of Physical manipulation and Touch gesture when using the Small version.**

A Friedman test reveals a significant effect of Touch gestures on success rate when performing a Lift-Up ( $\chi^2=16$ ,  $p=.001$ ), a Translation ( $\chi^2=15.75$ ,  $p=.001$ ), a Roll ( $\chi^2=6.4$ ,  $p=.01$ ) or a Rotation ( $\chi^2=21.6$ ,  $p<.001$ ):

- Lift-Up: a Wilcoxon test reveals a significant difference between Tap and Spread (92.22% vs. 41.67%;  $p=.001$ ) and Tap and Pinch (92.22% vs. 58.33;  $p=.04$ ). Success rate with Drag is 80.56%.
- Translation: a Wilcoxon test reveals a significant difference between Tap and Spread (91.67% vs. 36.11%;  $p=.001$ ) and between Pinch and Spread (77.78% vs. 36.11%;  $p=.02$ ). Success rate with Drag is 66.67%.
- Roll: a Wilcoxon test reveals a significant difference between Tap and Drag (95.83% vs. 86.11%;  $p=.04$ ).
- Rotation: a Wilcoxon test reveals a significant difference between Tap (86.11%) / Drag (80.56%) and Spread (40.28%) / Pinch (50.0%;  $p<.02$ ).

#### Completion time

A Wilcoxon test shows no difference between the Small and the Large versions ( $p=.08$ ). Overall, it took participants 2.5 seconds to perform a combined gesture. While we found some differences across gesture combinations, all of them are compatible with the micro-interactions concept [1], i.e. fast interactions that take less than 4s completion: all times ranged between 2.1s and 2.7s.

#### Unintentional touches

We recorded unintended touches on the Small and Large versions. Overall, results were similar for both versions: we detected unintended touches in 2% of the trials. These touches did not necessarily raise errors. The sequential use of a touch interaction after a physical gesture prevents from launching a command unintentionally.

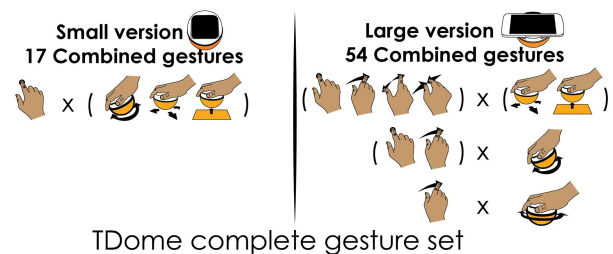
#### Subjective feedback

When considering the physical manipulations, results show that with the Small version, more than 50% of the participants found easy or very easy (4 or 5 on Likert scale) to perform a combined gesture involving a Roll, Translation or Lift-Up. In the case of the Large version, more than 75% of participants rated these gestures as easy or very easy.

When considering the touch gestures, we observed that with the Small version more than 50% of participants found difficult or very difficult (1 or 2 on Likert scale) to perform a combined gesture involving a Spread or Pinch. With the Large version, 60% or more of the participants found easy or very easy to perform combined gestures involving any kind of touch gesture.

#### Summary

Results reveal differences between the Small and Large versions (Figure 10). With the Small version, the experiment reveals that 17 combined gestures can be comfortably and efficiently performed: those based on the combination of a Roll (8 directions), a Translation (8 directions) or a Lift-Up with a Tap gesture (with a success rate of 95.83%, 91.67% and 92.22% respectively).



**Figure 10. Summary of the 17 (Small) + 54 (Large) combined gestures which offer a good usability and performance.**

With the Large version, the experiment reveals that 54 combined gestures can be comfortably and efficiently performed: 16 results from the combination of a Roll (95.49% success rate) with Tap or Drag gesture, 36 result from the combination of a Translation (91.67% success rate), or Lift-Up (95.83% success rate) with one of the four touch gestures (Tap, Drag, Pinch, Spread) and 2 results from the combination of a Rotation with a Drag (98.61% success rate).

#### MAPPING TDOME GESTURES TO MDE TASKS

We elicited user input through a user study to explore how the selected set of gestures from our previous experiment can be mapped to MDE interactive tasks.



## Overview and rationale

We asked users to choose, for each TDome task, one gesture from the set of gestures selected in the controlled experiment.

## MDE tasks considered

From our scenarios, we considered the following 7 tasks: pointing on a distant display; zooming on a distant display; displacing a window from one display to another (horizontal tunnel, vertical tunnel); sending a window from Tabletop/TDome (user position) to a distant display and vice-versa; selecting an icon on the radar view; panning and zooming a focused view of a distant context; and typing on a private keyboard.

## Participants

12 (1 female) students and researchers from the local university volunteered for this study. They were aged 31.9 years on average (SD=9). Five of them took part in the previous studies.

## Procedure

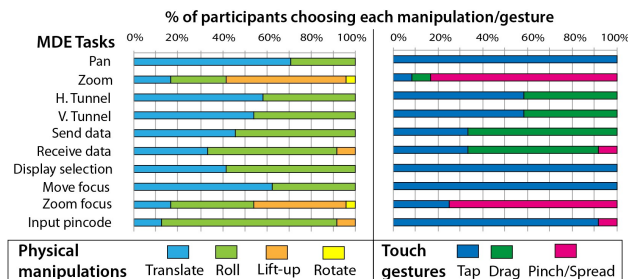
Participants were given the two TDome versions (Small and Large) and were situated in an MDE environment comprised of a tabletop, 4 displays and 1 video-projection. We familiarized our participants with TDome capabilities by showing them a video illustrating the combined gestures (without showing any interactive task). For each combined gesture, we asked participants to perform it themselves with both versions of TDome. Then, we asked participants to select and justify, for each task and each TDome version, which gesture they preferred. The session took about 15 minutes.

## Collected data

Every user generated one sheet with a summary of the gestures chosen for each task and TDome version. We recorded users' verbal comments.

## Results

Amongst all available combined gestures, only one was never used in our study (Lift-Up + Drag). Overall, participants took advantage of the gestures diversity to match the different tasks. The agreement scores [36] of the combined gestures (Physical manipulation + touch gestures) range between 0.3 and 0.6. These scores are in line with previous studies [36]. For the sake of clarity, we detail separately for each task the choice of physical manipulations and touch input.



**Figure 11. Percentage of users that chose each physical gesture on both versions of TDome (left) and each touch gesture on the Large version (right) for MDE tasks.**

## Physical manipulations

Our results were similar for the Small and Large versions concerning which physical gesture to use. Thus we report both results together (i.e. 24 gestures per task).

Two physical gestures were used more often: Translate and Roll (Figure 11-left). For some tasks, one was preferred over the other: *Translation* for panning (17/24), or for moving a focus (15/24); *Roll* for private pincode input (19/24). For other tasks, such as redirecting data using the tunnel, output redirection or display selection, there was no clear preference for one of these two gestures.

The Lift-Up gesture was used for zooming 13 times (i.e. lifting up the device activates zoom mode). Rotation was used only one time for each of our zooming tasks.

## Touch gestures

While only the Tap gesture is feasible on the Small version, users selected different gestures on the Large version according to the task (Figure 11-right). For instance, Pinch and Spread were preferred for zooming (10/12), and Drag was preferred for sending content from the tabletop to other displays (8/12). Taping was the preferred gesture for map panning (12/12), display selection (12/12) or pincode input (11/12).

## DISCUSSION AND PERSPECTIVES

### TDome benefits

We presented two versions of TDome: a Small version with an integrated touchscreen and a Large version based on attaching a smartphone. TDome's unique features offer several advantages to interact with MDEs:

- TDome supports performing multiple combined gestures involving a physical manipulation of the device (Translation, Roll, Rotation or Lift-Up) followed by a touch gesture on the touchscreen (Tap, Drag, Pinch or Spread). Such a combination prevents from unintended activations due to parasite touches on the touchscreen;
- The combined gestures from our final set represent good candidates to support rapid access to interactive commands;
- The two TDome versions are easily interchangeable and offer complementary functionalities: the Small version is useful to rapidly launch shortcuts, while the Large version offers a larger display area and supports multi-touch gestures (Pinch and Spread);
- Rolling TDome can be used to interact with multi-dimensional data through continuous physical gestures, as demonstrated earlier [23];
- The embedded display can be used as a personal display area to augment output visualisation, such as in overview + detail techniques [29];

- The embedded display can also be used to show feedback of the TDome interactions, such as displaying a copied object for the copy/paste technique.

#### **Guidelines for mapping TDome gestures to MDE tasks**

We propose a set of guidelines to map TDome gestures to MDE tasks based on our mapping study as well as on our experience developing TDome interaction techniques:

- TDome offers a diversity and large number of possible gestures of which users can take advantage as illustrated in our mapping study. Some of these gestures have natural mappings with MDE tasks, such as Rolling towards oneself to display private information, Pinch and Spread for zooming or dragging for sending data to another display. Appropriately combined with a physical manipulation or a touch gesture, these should become the “default” combined gestures with TDome on MDEs.
- While some mappings are obvious and have a large consensus, others are sometimes split between two types of gestures (usually Roll or Translation): this suggests that using TDome in an MDE could benefit from a device personalization step wherein the user defines his preferred mapping, especially for output redirection.
- Interacting in an MDE involves system tasks (i.e. tasks related to the environment, such as display selection) and application tasks. Since these tasks could be assigned to the same TDome gestures, there is a need for a mode switching gesture. The Lift-Up manipulation combined with touch input, is a good candidate as it was considered for switching between pan and zoom tasks in our mapping study

#### **Memorability of a large number of gestures**

TDome offers a large set of usable combined gestures. This diversity of available controls is particularly relevant to tackle tasks in MDEs. However, increasing the amount of controls might make them hard to memorize. The physical nature of these combined gestures can help cluster them according to the physical manipulation, as shown in our mapping study. Further experiments are required to identify how such clustering can improve gestures or command memorization.

#### **Collaboration**

MDEs are naturally designed to support collaboration. We can envision multiple TDome like devices, each controlled by the MDE’s users. However, in such cases input and output redirection mechanisms would need to be effectively controlled. Control mechanisms have been proposed by others [14] to handle synchronization, locking and input conflicts, and future iterations of TDome will adapt or build on such proposals.

#### **Future work**

In this paper we focused on the suitability of TDome capabilities for MDE tasks and the feasibility of its combined gestures. TDome interaction techniques still need

to be fine tuned and future work should compare their performance with a baseline for each MDE task. Theoretically, since TDome integrates the same capabilities as existing MDE devices, we hypothesise that it can perform similarly for each individual MDE task. For instance, TDome can perform translations like a mouse, and has the same touch and mid-air capabilities as a smartphone. Moreover, since TDome is a unique device that supports a range of core MDE tasks, it should improve the overall performance by reducing homing transition times and promoting the interaction flow. Therefore, beyond individual controlled comparisons, it would be interesting to carry a longitudinal study. We leave these studies for future work.

Beyond these aspects, we plan to focus on user expertise of TDome techniques: most menu or command techniques consider novice and expert modes as well as the transition from novice to expert [15]. In our work we focused on how the combined gestures are performed. It will be interesting to design techniques that efficiently support both novice and expert users and the transition from one group to the other, as done with the Marking Menus [37].

We also plan to investigate the extension of our physical manipulation gestures by adding thresholds. For instance, each Roll gesture could launch two different commands according to the Roll amplitude (under or over 42° according to our study results). Technical alternatives also need to be investigated to replace the infrared bezel used to detect the TDome translations. We are currently exploring the application of conductive paint on the external surface of TDome, which will allow using the device on any capacitive surface.

#### **CONCLUSION**

We presented TDome, a device designed for interactions in MDEs. We designed two TDome prototypes: a Small version with an integrated touchscreen and a Large version based on attaching a smartphone. We explored combined gestures involving a physical manipulation (Translation, Roll, Rotation or Lift-Up) followed by a touch gesture (Tap, Drag, Pinch or Spread) through a 3-steps process. First, an exploratory study focusing on comfort established that 60 combined gestures could be comfortably performed. Second, a controlled experiment evaluated the user's performance as well as the subjective perceived difficulty. Results revealed that the number of gestures that can be precisely and easily performed is 17 with the Small version, and 54 with the Large version. Finally, a user survey explored the mappings between these gestures and MDE tasks. Results show that some combined gestures are more prone to be used in specific tasks than other. In general, we find participants are able to match TDome features to MDE tasks

**ACKNOWLEDGEMENTS**

This work is funded by the neOCampus project of the University Paul Sabatier (Toulouse, France) and part of the AP2 project (research grant AP2 ANR-15-CE23-0001).

**REFERENCES**

1. Daniel L. Ashbrook. 2010. *Enabling Mobile Microinteractions*. Ph.D. Dissertation. Georgia Institute of Technology, Atlanta, GA, USA. Advisor(s) Thad E. Starner. AAI3414437
2. Ravin Balakrishnan, Thomas Baudel, Gordon Kurtenbach, and George Fitzmaurice. 1997. The Rockin'Mouse: integral 3D manipulation on a plane. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems (CHI '97)*. ACM, New York, NY, USA, 311-318. DOI=<http://dx.doi.org/10.1145/258549.258778>
3. Dominikus Baur, Sebastian Boring, and Steven Feiner. 2012. Virtual projection: exploring optical projection as a metaphor for multi-device interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 1693-1702. DOI=<http://dx.doi.org/10.1145/2207676.2208297>
4. Benjamin B. Bederson. 2004. Interfaces for staying in the flow. *Ubiquity* 2004, September (September 2004), 1-1. DOI=<http://dx.doi.org/10.1145/1074068.1074069>
5. Hrvoje Benko and Steven Feiner. 2005. Multi-monitor mouse. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems (CHI EA '05)*. ACM, New York, NY, USA, 1208-1211. DOI=<http://dx.doi.org/10.1145/1056808.1056878>
6. Hrvoje Benko, Shahram Izadi, Andrew D. Wilson, Xiang Cao, Dan Rosenfeld, and Ken Hinckley. 2010. Design and evaluation of interaction models for multi-touch mice. In *Proceedings of Graphics Interface 2010 (GI '10)*. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 253-260.
7. Louis-Pierre Bergé, Marcos Serrano, Gary Perelman, and Emmanuel Dubois. 2014. Exploring smartphone-based interaction with overview+detail interfaces on 3D public displays. In *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services (MobileHCI '14)*. ACM, New York, NY, USA, 125-134. DOI: <http://dx.doi.org/10.1145/2628363.2628374>
8. Louis-Pierre Bergé, Emmanuel Dubois, and Mathieu Raynal. 2015. Design and Evaluation of an "Around the SmartPhone" Technique for 3D Manipulations on Distant Display. In *Proceedings of the 3rd ACM Symposium on Spatial User Interaction (SUI '15)*. ACM, New York, NY, USA, 69-78. DOI=<http://dx.doi.org/10.1145/2788940.2788941>
9. Boring, S. 2007. Interacting in Multi-Display Environments. In *DC Ubicomp'07*. ACM.
10. Sebastian Boring, Marko Jurmu, and Andreas Butz. 2009. Scroll, tilt or move it: using mobile phones to continuously control pointers on large public displays. In *Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group: Design: Open 24/7 (OZCHI '09)*. ACM, New York, NY, USA, 161-168. DOI=<http://dx.doi.org/10.1145/1738826.1738853>
11. Sebastian Boring, Dominikus Baur, Andreas Butz, Sean Gustafson, and Patrick Baudisch. 2010. Touch projector: mobile interaction through video. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 2287-2296. DOI=<http://dx.doi.org/10.1145/1753326.1753671>
12. Christophe Bortolaso, Matthew Oskamp, T.C. Nicholas Graham, and Doug Brown. 2013. OrMiS: a tabletop interface for simulation-based training. In *Proceedings of the 2013 ACM international conference on Interactive tabletops and surfaces (ITS '13)*. ACM, New York, NY, USA, 145-154. DOI=<http://dx.doi.org/10.1145/2512349.2512792>
13. Tsung-Hsiang Chang and Yang Li. 2011. Deep shot: a framework for migrating tasks across devices using mobile phone cameras. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 2163-2172. DOI=<http://dx.doi.org/10.1145/1978942.1979257>
14. Olivier Chapuis, Anastasia Bezerianos, and Stelios Frantziskakis. 2014. Smarties: an input system for wall display development. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 2763-2772. DOI=<http://dx.doi.org/10.1145/2556288.2556956>
15. Andy Cockburn, Carl Gutwin, Joey Scarr, and Sylvain Malacria. 2014. Supporting Novice to Expert Transitions in User Interfaces. *ACM Comput. Surv.* 47, 2, Article 31 (November 2014), 36 pages. DOI: <http://dx.doi.org/10.1145/2659796>
16. Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. 2014. Consumed endurance: a metric to quantify arm fatigue of mid-air interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 1063-1072. DOI: <http://dx.doi.org/10.1145/2556288.2557130>
17. Ken Hinckley, Mike Sinclair, Erik Hanson, Richard Szeliski, and Matt Conway. 1999. The VideoMouse: a camera-based multi-degree-of-freedom input device. In *Proceedings of the 12th annual ACM symposium on User interface software and technology (UIST '99)*. ACM, New York, NY, USA, 103-112. DOI=<http://dx.doi.org/10.1145/320719.322591>

18. Steven Houben and Nicolai Marquardt. 2015. WatchConnect: A Toolkit for Prototyping Smartwatch-Centric Cross-Device Applications. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (CHI '15). ACM, New York, NY, USA, 1247-1256. DOI: <http://dx.doi.org/10.1145/2702123.2702215>
19. Chang-Min Kim and Tek-Jin Nam. 2015. G-raff: An Elevating Tangible Block for Spatial Tabletop Interaction. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (CHI '15). ACM, New York, NY, USA, 4161-4164. DOI: <http://dx.doi.org/10.1145/2702123.2702444>
20. Miguel A. Nacenta, Dzmitry Aliakseyeu, Sriram Subramanian, and Carl Gutwin. 2005. A comparison of techniques for multi-display reaching. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '05). ACM, New York, NY, USA, 371-380. DOI: <http://dx.doi.org/10.1145/1054972.1055024>
21. Mathieu Nancel, Julie Wagner, Emmanuel Pietriga, Olivier Chapuis, and Wendy Mackay. 2011. Mid-air pan-and-zoom on wall-sized displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '11). ACM, New York, NY, USA, 177-186. DOI: <http://dx.doi.org/10.1145/1978942.1978969>
22. Mathieu Nancel, Olivier Chapuis, Emmanuel Pietriga, Xing-Dong Yang, Pourang P. Irani, and Michel Beaudouin-Lafon. 2013. High-precision pointing on large wall displays using small handheld devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '13). ACM, New York, NY, USA, 831-840. DOI: <http://dx.doi.org/10.1145/2470654.2470773>
23. Gary Perelman, Marcos Serrano, Mathieu Raynal, Celia Picard, Mustapha Derras, and Emmanuel Dubois. 2015. The Roly-Poly Mouse: Designing a Rolling Input Device Unifying 2D and 3D Interaction. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (CHI '15). ACM, New York, NY, USA, 327-336. DOI: <http://dx.doi.org/10.1145/2702123.2702244>
24. Gary Perelman, Marcos Serrano, Mathieu Raynal, Celia Picard, Mustapha Derras, and Emmanuel Dubois. 2016. DECO: A Design Space for Device Composition. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems* (DIS '16). ACM, New York, NY, USA, 435-446. DOI: <http://dx.doi.org/10.1145/2901790.2901893>
25. Roman Rädle, Hans-Christian Jetter, Nicolai Marquardt, Harald Reiterer, and Yvonne Rogers. 2014. HuddleLamp: Spatially-Aware Mobile Displays for Ad-hoc Around-the-Table Collaboration. In *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces* (ITS '14). ACM, New York, NY, USA, 45-54. DOI: <http://dx.doi.org/10.1145/2669485.2669500>
26. Ramesh Raskar, Greg Welch, Matt Cutts, Adam Lake, Lev Stesin, and Henry Fuchs. 1998. The office of the future: a unified approach to image-based modeling and spatially immersive displays. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (SIGGRAPH '98). ACM, New York, NY, USA, 179-188. DOI: <http://dx.doi.org/10.1145/280814.280861>
27. Jun Rekimoto. 1997. Pick-and-drop: a direct manipulation technique for multiple computer environments. In *Proceedings of the 10th annual ACM symposium on User interface software and technology* (UIST '97). ACM, New York, NY, USA, 31-39. DOI: <http://dx.doi.org/10.1145/263407.263505>
28. Jun Rekimoto and Masanori Saitoh. 1999. Augmented surfaces: a spatially continuous work space for hybrid computing environments. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (CHI '99). ACM, New York, NY, USA, 378-385. DOI: <http://dx.doi.org/10.1145/302979.303113>
29. Housseem Saidi, Marcos Serrano, and Emmanuel Dubois. 2016. Investigating the effects of splitting detailed views in Overview+Detail interfaces. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services* (MobileHCI '16). ACM, New York, NY, USA, 180-184. DOI: <http://dx.doi.org/10.1145/2935334.2935341>
30. Dominik Schmidt, Corina Sas, and Hans Gellersen. 2013. Personal clipboards for individual copy-and-paste on shared multi-user surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '13). ACM, New York, NY, USA, 3335-3344. DOI: <http://dx.doi.org/10.1145/2470654.2466457>
31. Julian Seifert, Sebastian Boring, Christian Winkler, Florian Schaub, Fabian Schwab, Steffen Herrdum, Fabian Maier, Daniel Mayer, and Enrico Rukzio. 2014. Hover Pad: interacting with autonomous and self-actuated displays in space. In *Proceedings of the 27th annual ACM symposium on User interface software and technology* (UIST '14). ACM, New York, NY, USA, 139-147. DOI: <http://dx.doi.org/10.1145/2642918.2647385>
32. Marcos Serrano, Barrett Ens, Xing-Dong Yang, and Pourang Irani. 2015. Gluey: Developing a Head-Worn Display Interface to Unify the Interaction Experience in Distributed Display Environments. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and*



- Services* (MobileHCI '15). ACM, New York, NY, USA, 161-171.  
DOI=<http://dx.doi.org/10.1145/2785830.2785838>
33. Nicole Sultanum, Sowmya Somanath, Ehud Sharlin, and Mario Costa Sousa. 2011. "Point it, split it, peel it, view it": techniques for interactive reservoir visualization on tabletops. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces* (ITS '11). ACM, New York, NY, USA, 192-201. DOI=<http://dx.doi.org/10.1145/2076354.2076390>
34. Manuela Waldner, Ernst Kruijff, and Dieter Schmalstieg. 2010. Bridging gaps with pointer warping in multi-display environments. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries* (NordiCHI '10). ACM, New York, NY, USA, 813-816.  
DOI=10.1145/1868914.1869036  
<http://doi.acm.org/10.1145/1868914.1869036>
35. Jacob O. Wobbrock, Meredith Ringel Morris, and Andrew D. Wilson. 2009. User-defined gestures for surface computing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '09). ACM, New York, NY, USA, 1083-1092.  
DOI=<http://dx.doi.org/10.1145/1518701.1518866>
36. Xing-Dong Yang, Edward Mak, David McCallum, Pourang Irani, Xiang Cao, and Shahram Izadi. 2010. LensMouse: augmenting the mouse with an interactive touch display. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '10). ACM, New York, NY, USA, 2431-2440.  
DOI=<http://dx.doi.org/10.1145/1753326.1753695>
37. Shengdong Zhao and Ravin Balakrishnan. 2004. Simple vs. compound mark hierarchical marking menus. In *Proceedings of the 17th annual ACM symposium on User interface software and technology* (UIST '04). ACM, New York, NY, USA, 33-42.  
DOI=<http://dx.doi.org/10.1145/1029632.1029639>