SurfAirs: Surface + Mid-air Input for Large Vertical Displays

Emmanuel Courtoux, Caroline Appert, Olivier Chapuis

To cite this version:

Emmanuel Courtoux, Caroline Appert, Olivier Chapuis. SurfAirs: Surface + Mid-air Input for Large Vertical Displays. CHI 2023 - International conference on Human factors in computing systems, Apr 2023, Hamburg, Germany. 15 p., 10.1145/3544548.3580877 . hal-04009207

HAL Id: hal-04009207

https://hal.science/hal-04009207

Submitted on 1 Mar 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

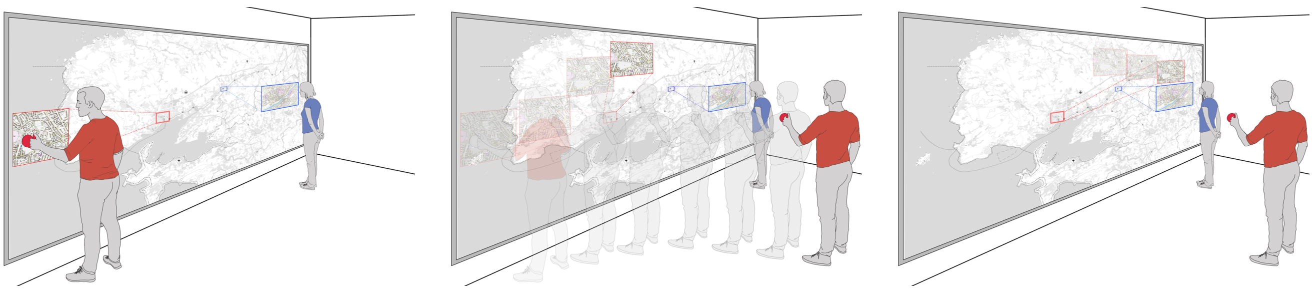
L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

SurfAirs: Surface + Mid-air Input for Large Vertical Displays

Emmanuel Courtoux
Université Paris-Saclay, CNRS, Inria
Orsay, France

Caroline Appert
Université Paris-Saclay, CNRS, Inria
Orsay, France

Olivier Chapuis
Université Paris-Saclay, CNRS, Inria
Orsay, France



1) Surface Control 2) Surface-to-Mid-air Continuous Control 3) Mid-air Control

Figure 1: Interacting with a *SurfAir* to drag an object across the wall. 1) The user on the left adjusts the zoom factor in the inset window by rotating the *SurfAir* on the surface. 2) He then initiates a drag-and-drop action to move that window by sliding the *SurfAir* on the surface. In order to get a wide viewing angle on the scene and to cover a large distance quickly, he steps backward. Since the *SurfAir* can transition from surface control to air control, the user can move freely in space while carrying on his drag-and-drop action. 3) He drops the window close to the other user without having to enter her personal physical space.

ABSTRACT

Large vertical surfaces such as wall displays allow users to work with a very large and high-resolution workspace. Such displays promote physical navigation: users can step close to the display to see details, but also move away to get a wider view of the workspace. In terms of input, current solutions usually combine direct touch on the wall with input on a handheld device, disconnecting close and distant input rather than treating it as a continuum. We present *SurfAirs*, which are physical controllers that users can manipulate on screen (*surface input*), in the air (*mid-air input*), and transition from the surface to the air during a single manipulation (*hybrid input*). We report on two user studies that compare *SurfAirs*' performance with bare-hand input for both mid-air and hybrid input. Participants prefer and perform better with *SurfAirs*.

CCS CONCEPTS

• **Human-centered computing** → **Haptic devices**; **Gestural input**.

KEYWORDS

Tangible Input; Wall Displays; Remote Controllers; Bare-hand input.

CHI'23, Author's version, Courtoux et al.
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM. This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23), April 23–28, 2023, Hamburg, Germany*, https://doi.org/10.1145/3544548.3580877.

ACM Reference Format:
Emmanuel Courtoux, Caroline Appert, and Olivier Chapuis. 2023. *SurfAirs*: Surface + Mid-air Input for Large Vertical Displays. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23), April 23–28, 2023, Hamburg, Germany*. ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3544548.3580877

1 INTRODUCTION

Large vertical surfaces such as high-resolution wall displays allow users to work with a very large information space. Such displays are particularly suited to, *e.g.*, interacting with geographic data [3, 12], performing visual analysis [33, 39, 60], sorting large collections of data [28, 41], or even playing games collaboratively [66]. Large vertical surfaces enable two levels of interaction. Users can come close to the display to see details, but they can also step back to get a wider view of the workspace [3, 28]. The interaction space is thus not limited to the display surface itself but also encompasses the physical space in front of it, where users should be able to move freely. Designing input techniques that effectively work across this large interaction space is challenging, however.

Bare-hand input may be considered the most intuitive interaction technique for interacting with wall displays. Users can rely on mid-air gestures [44, 51, 65] to interact from afar, and on direct touch gestures on the wall to interact up close. But while bare-hand input has the advantage of keeping users' hands free, it also raises important challenges in terms of interaction design. In particular, as the design space of hand movements is unstructured yet constrained and user-dependent, designing bare-hand input that is steady, accurately recognized and that does not collide with regular hand movements is very difficult. Even an action as simple as clicking is challenging to design and implement [4, 65]. As a result, interaction with wall displays often involves a handheld device that serves as a remote controller (*e.g.* [9, 33, 45, 60, 67]). While this offers a good solution for indirect control, it requires users to always hold a device and it does not integrate smoothly with direct touch interaction on the wall display.

Taking inspiration from projects with tabletops such as [35, 63, 68], some systems rely on tangible controllers to interact with vertical displays [12, 23, 40, 66]. Recent empirical results suggest that such tangible controllers are even more efficient and more comfortable than touch gestures for manipulating virtual objects displayed on the wall [12]. When their design makes it possible for users to attach them to the surface [12, 23, 40], users can free their hands at will. They can also easily switch between multiple tangibles.

However, existing tangible controllers are limited as they support interaction either on the display surface itself or in the air, but not both. This is a strong limitation with vertical displays that afford movements in space to interact either up close or from afar. In this paper, we contribute *SurfAirs*, a new generation of controllers for vertical displays that support various interaction styles. As Figure 1 illustrates, users can manipulate them on screen (*surface input*), in the air (*mid-air input*) and transition from the surface to the air during a single manipulation (*hybrid input*). Like WallTokens [12] or Geckos [40], a *SurfAir* can be attached to, and detached from the vertical surface, making it easy for users to free their hands or grab another *SurfAir*.

After a review of related work, we present our contributions: 1) a characterization of hybrid input on wall displays; 2) the prototyping of physical controllers that support such hybrid input; 3) a user study that evaluates their performance for mid-air interaction only; and 4) a user study that evaluates their performance for both surface and mid-air input as well as for transitioning between these two types of input. In both studies, we compare *SurfAirs* with bare-hand input as a baseline. Participants prefer and perform better with *SurfAirs*.

2 RELATED WORK

In this section, we review previous work about both hybrid interaction, which combines direct touch and mid-air control, and about controllers for vertical displays.

2.1 Combining Surface and Mid-air Control

To address issues related to touch input on small-sized screens, several research prototypes (*e.g.* [6, 7, 34, 38]) augment handheld devices with sensors to track hand input in the air around the device. But the advantages of extending touch to the air around or above the screen are not limited to small devices. Marquardt *et al.* [43] propose such a *continuous interaction space* above a tabletop, and list the many interaction techniques that it enables to, *e.g.*, interact from up-close or afar, grab out-of-reach objects or perform high-precision manipulations. Prior to the conceptualization by Marquardt *et al.*, projects such as SecondLight [26] had investigated the detection of users' hand and objects beyond the tabletop's surface from a technological standpoint and proposed actual tabletop prototypes. For example, Hilliges *et al.* use this continuous interaction space to enable intuitive manipulations of virtual objects [24]. Extended interaction spaces can also be very useful in multi-display environments such as LightSpace [70] where users need both *local power* for interacting with the current display and *remote powers* for interacting with distant displays [48].

Such hybrid input is particularly important for wall displays as users move physically in front of the display. Some techniques for wall displays rely on the notion of proxemics and consider the user's distance to the screen as an input parameter to, *e.g.*, extend users' reach [58]; adapt the presentation of the information space [32]; and even display personal information when the user is close-enough for their body to hide this information from others [64]. As mentioned above, physical navigation is key when working with large information spaces [2, 3]. It is valued over virtual navigation [29], and even more so for difficult tasks [42]. Hybrid input can also facilitate collaborative work. On the one hand, touch input can help switch between different collaboration styles [28] and handle concurrent access [30]. On the other hand, touch input can also cause physical conflicts or hide content from other users [22, 30]. This advocates for enabling interaction both up-close and from afar.

Some systems implement hybrid input for wall displays, enabling interaction through either direct touch or mid-air gestures. For instance, Schick *et al.* [56] rely on RGB cameras and computer vision methods to *extend touch* with a raycast along the user's arm. In a similar spirit, Jakobsen *et al.* extend a multitouch wall display with optical tracking to support mid-air raycast [31]. They report on studies that compare touch and mid-air input with contrasted observations. Participants preferred touch to interact with small objects, but tended to choose mid-air input over touch for large-scale manipulations or when they needed a large viewing angle. Like the Pointable technique [4] for tabletops or the MirrorTouch public display [47], the wall display in Jakobsen *et al.*'s studies does not integrate touch and mid-air input. Users can rely on either touch or mid-air, but they cannot, for instance, initiate a movement with a sliding gesture on the surface and continue with that same movement in the air like the Talaria technique does [54].

All the systems mentioned above rely on bare-hand, device-free techniques. Such input has the great advantage of leaving users' hands free, but it also has downsides. In particular, it suffers from a lack of haptic feedback and requires users to learn specific hand postures to perform actions as simple as mode switching [59] or clicking [65]. In addition to the difficulty of learning and performing postures such as ThumbTrigger [18, 65], SideTrigger [4] or Multirays [44], bare-hand input potentially conflicts with users' movements that are not intended to be interpreted by the system. This is particularly true when users move in space and discuss with each other in a collaborative context.

2.2 Physical Controllers for Vertical Displays

Several research prototypes complement or replace bare-hand input with an off-the-shelf input device. The device can be a basic input device (such as a mouse [29, 51] or a multi-touch trackpad [42]) or it can be a personal handheld device [9, 39, 45, 51, 60]. Relying on an additional device enables efficient implementation of indirect pointing through *e.g.*, an acceleration function and a mode switch between absolute and relative cursor control [45, 50]. In addition to acting as a pointing device, the screen of a handheld device can be used to recognize simple touch actions such as double tap or slide gestures [39]. The device's screen can even host some UI components to invoke commands and adjust parameters in the form of software components [9, 60] or physical components [33].

Rather than relying on conventional devices, some projects fall in the category of Tangible User Interfaces with tailored, *ad hoc* tangibles. Tangible User Interaction (TUI) means interacting with digital information through the physical environment. What TUI encompasses is debatable. For example, taken literally, a mouse is a physical object in the environment and in that sense can be seen as a tangible interaction device. However, the HCI community tacitly agrees that TUI does not encompass traditional workstation and input devices. This can be achieved by giving a physical shape to digital information [25], by making everyday objects play an active role in the digital environment [69] or by having one or several physical, specialized controllers that can multiplex input or output in space [15, 35, 61]. This latter approach is particularly relevant for environments that involve a large interaction space. Moreover, the feedthrough provided by the manipulation of physical artefacts can help increase group awareness and facilitate collaboration [49].

Regarding mid-air interaction, tangibles have received more attention for immersive environments [1, 5, 11, 13, 14, 17, 37] than for wall displays. In the specific context of wall displays, tangibles have been almost exclusively proposed for interaction on the display itself. The seminal pick-and-drop technique [55] allows users to tap an object with a pen to pick it, and then tap again to drop it elsewhere. As one pen is associated with one user, pick-and-drop multiplexes input in space, enabling several users to perform concurrent manipulations. In the *Seconds matter* project [16], Fraser *et al.* study the transition from off-screen space to on-screen space for pen input on a digital whiteboard. In their system, two users (say U_1 and U_2) collaborate synchronously but are located in two distant sites. The location of U_1's pen relative to U_1's whiteboard is displayed on U_2's whiteboard so that U_2 can anticipate where interaction will take place and thus better synchronize their own actions with that of U_1. Several projects propose other types of tangibles for vertical displays but for surface manipulation only. For example, in Miners [66], users rely on touch input combined with small tangibles in the form of tokens that are similar to TouchTokens [46]. A tangible in the Miners game identifies a user. When in contact with the wall, it delineates an area where touch input is associated with that user. Some tangibles can also be attached to the wall [23, 40] and used as tangible widgets in the spirit of what SLAP widgets enable on tabletops [68]. WallTokens [12] are tangibles that can be both attached to, or slid over, the surface. They can thus be used as either widgets or controllers. Empirical observations reveal the advantages of such tangibles over touch gestures in terms of comfort and speed when manipulating virtual objects displayed on a surface, either horizontal [20, 61] or vertical [12].

Relying on a physical controller for interacting with wall displays can address issues related to bare-hand input. Physical controllers can be organized into two categories: personal devices or *ad hoc* tangibles. While using a personal device to interact with a wall display enables powerful interaction techniques, it also means constantly holding this personal device. In addition to the fatigue this can cause, users do not have their hands free, which can be annoying when interacting with other artefacts as well as when performing bi-manual actions on the wall itself. Also, as mentioned earlier, such device-based interaction is indirect and does not integrate well with direct touch interaction on the wall, that users like to rely on [31].

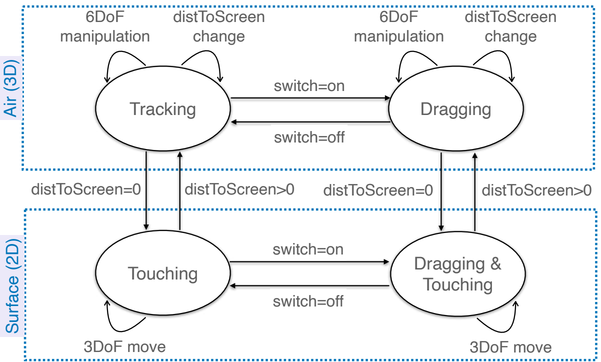


Figure 2: Interaction states and transitions for a hybrid controller.

A couple of research projects equip smartphones with additional sensors to recognize a contact between the smartphone and an external touchscreen [53, 57], enabling users with *phone touch* input. However, the form factor of a smartphone makes such direct *phone touch* actions typically limited to basic taps. On the opposite, tangibles that are tailored to wall displays can support richer surface manipulations on the wall itself such as sliding and rotating, which can even be combined with touch actions [12, 66]. Such custom tangibles can even be designed so that users can attach them to the wall to free their hands for other actions [12, 23, 40]. However, all existing tangibles for vertical displays are tracked only when in contact with the display. On the opposite, the *SurfAir* controllers that we propose are actual hybrid physical controllers that implement the concept of *Off-Surface Tangibles*, which has been sketched only by Cherek *et al.* in the context of tabletops [10]. *SurfAirs* are tracked both on the surface and in the air, and enable interactions that span across the two.

3 CONTROLLERS FOR HYBRID INPUT WITH WALL DISPLAYS

Hybrid controllers can treat the surface and the air in front of it as a *continuous interaction space* [43]. Marquardt *et al.* have introduced this concept in the context of tabletops, demonstrating how such an extended interaction space can not only improve existing interaction techniques but enable novel techniques as well. In this section, we look at hybrid interaction from a lower-level perspective, describing it at the input device level (*i.e.*, the physical controller, not the interaction technique). We list the low-level properties that *hybrid* (touch+air) brings over *touch only*. To quantify the expressive power of a hybrid controller, we also characterize it in terms of input states and transitions in the spirit of Buxton's 3-state model [8]. As opposed to a surface controller that typically supports only two states (Out of Range and Touching), a hybrid controller is much more expressive. Figure 2 details the four states and multiple transitions within and across those states.

Distance-independent Touch. The most fundamental property that hybrid controllers bring over surface controllers is the ability for users to interact from afar and benefit from a large viewing angle. Enabling users to *touch from afar* requires tracking the position of the controller in the air (3D) in order to implement some raycast from the controller to the wall display (Tracking state in Figure 2-Right). It also requires mounting a switch on the controller to change from the Tracking state to the Dragging state without having to touch the wall. The addition of a switch has the positive side-effect of enriching surface interaction as well. When already in contact (Touching), users can activate the switch to enter an additional Dragging & Touching state.

Ideally, users should be able to smoothly transition from the Touching to the Dragging state (and vice versa) to perform continuous interactions while being able to adjust their physical position in space. This typically happens in the case of quasi-modal interactions such as drag-and-drops or area-based selections (rubber band or lasso). It can be because such interactions span a large distance but require precision as well. For instance, users can start interacting on the surface to precisely select a location or object and then step back to get a wider viewing angle and end their interaction far from where it started. Or, reciprocally, they can have to come in contact at the end of their interaction to benefit from the guidance of the surface to drop an object at a precise location. Transitioning from on surface to the air is also sometimes necessary because of physical constraints. For instance, regions of a large display that are too low or too high can be uncomfortable or impossible to reach while interacting on surface. Similarly, in a multi-user context, some regions of the display might not be available for on-surface interaction without disturbing other users. In such cases, being able to transition from the Touching to the Dragging state enables users to adapt their position and viewing angle without interrupting their ongoing interaction.

From 2D control to 3D control. In the case of a surface controller that must be kept in contact with the surface, users can manipulate that controller along a limited number of Degrees of Freedom (DoF). The surface provides haptic support that facilitates and improves the precision of surface manipulations but it also limits manipulations to translations on the surface itself (2 DoFs) and rotations along the axis that is orthogonal to the surface (1 DoF). On the opposite, when in the air, the controller can be positioned and oriented in 3D, enabling less steady but richer, 6-DoF manipulations (3D translation + 3D rotation). Interaction designers can use all of these degrees of freedom to enable full 3D manipulations. They can also use only a subset of these degrees of freedom to cope with some limitations of surface controllers for 2D manipulations. For example, a surface controller usually does not support scaling operations as the physical object cannot be stretched or compressed [13]. A hybrid controller can enable modal input to activate different virtual actions with the same physical action on the controller. For instance, rotating or scaling a virtual object could be achieved by either rotating the controller on or above the surface.

Distance-to-screen as a parameter. In addition to the absolute 3D positioning of the controller, interaction designers can use the position of the controller relative to the screen as an input parameter. This distance can be used as a *discrete* parameter to create a personal layer to enable interaction with personal data only when close enough to the screen [64], or even a series of discrete layers as in [21]. The distance to the screen can also be used as a *continuous* parameter. For example, distance can control the precision at which an action is performed in the spirit of high-precision sliders [6, 43].

4 *SURFAIRS*

This section describes the *SurfAir* controllers that we prototyped to implement the concept of hybrid input. As illustrated in Figure 3, users can interact with *SurfAirs* from different distances: on the surface itself or at varying distances that provide different viewing angles on the scene.

4.1 Design Requirements

Our main design requirement is to allow users to perform manipulations from close and afar as well as manipulations over varying distances. Our goal is to build a controller that users can *manipulate consistently on the surface and in the air*. We discard solutions that are based on a combination of finger-based input for surface interaction and controller-based input for distant interaction for the sake of a continuous interaction space where users can perform precise and comfortable input. Indeed, a touch+controller solution would either require users to perform some interaction with their non-dominant hand, thus loosing precision, or it would require them to move the controller from one hand to the other, thus breaking the continuity of interaction. To address these issues, we target a physical controller that users can manipulate with their dominant hand alone and in a consistent way for different types of input. Such a controller should be both *usable as a surface controller* so that users can perform precise manipulations by taking advantage of the guidance provided by the surface (as the user on the left in Figure 3) and *usable as a mid-air controller that supports eyes-free manipulation* so that users can interact from a distance without having their attention divided between the controller and the action that is taking place on the distant wall screen (as the user in the middle in Figure 3).

In addition to support for hybrid input, designing controllers for large vertical screens entails two additional requirements. First, large vertical displays are often used in a multi-user context. The system should thus support the *concurrent use of multiple controllers*. This not only makes it possible for users to work collaboratively but also for them to use multiple controllers as they would use multiple specialized tools. Second, the controllers should *comply with verticality*. In particular, users should be able to leave them on the surface to either free their hands for other purposes (*e.g.*, switching to another controller or answering a phone call) or just leave the controllers in place for later use.

4.2 Form Factor

We adopted a modular fabrication process with several components that we assemble together by means of screwing and interlocking. This modular approach makes *SurfAirs* customizable to some extent. During our iterative design process, it had the advantage of making it fast and easy to test and refine the different parts of *SurfAirs*. In our case, it was particularly useful to test different handle designs, but it could also make it easy for customizing their appearance. For instance, creating a set of tokens that have different appearances simply requires designing bases with varying shapes and combine them with a given handle design.

Figure 3: *SurfAirs* are custom physical controllers that users can manipulate both on the surface and in the air. In this scene, three users interact with *SurfAirs*: i) the user on the left uses a *SurfAir* as a surface controller, sliding and rotating it to pan and zoom in the inset window; ii) the user in the middle interacts from afar as she needs a wide view to perform a large lasso selection (in red); and iii) the user on the right is adjusting the position of the area to be magnified. He picked the viewport's proxy on the wall, and then stepped back to perform a large movement towards North-West without disturbing other users.

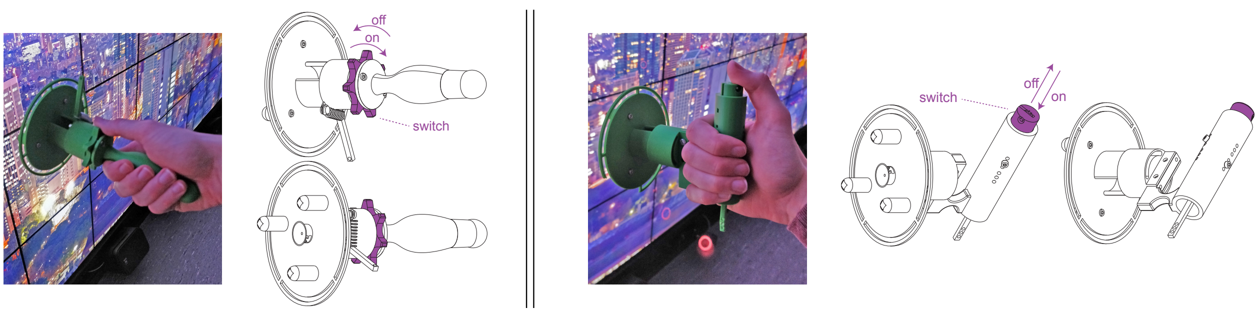


Figure 4: *SurfAir* prototypes. (Left) *Torch* supports a torch-like grip. (Right) *Handle* supports a door handle-like grip.

When designing our *SurfAirs*, we first took inspiration from the simple suction cup mechanism for attaching and detaching a tangible on a vertical display that has been recently introduced in the WallTokens project [12]. Like a WallToken, a *SurfAir* is mounted on a base that features three feet and a central suction cup. By default, the suction cup is not in contact with the screen, so that users can slide the *SurfAir* on the surface without experiencing any friction. When they want to attach the *SurfAir* to the surface, they push its handle. This has the effect of bringing the suction cup in contact with the surface to stick the *SurfAir* in place.

A *SurfAir*'s handle is more elaborate than the simple door-knob-like handle of a WallToken. In particular, it features a switch. This requires combining several components, which leads to a different balance in terms of weight. In order to make the whole controller stable when users move, push and pull it, we designed a robust connection between the base and the handle involving rail-guided cylinders (Figure 4). Thus, the *SurfAir* remains steady when the user pushes the handle to attach it to the surface. Most of the design process then focused on finding a good trade-off between comfort when pointing and stability when activating the switch. In particular, not all fingers should be involved in the grip's handle so that users can easily free one of them to manipulate the switch. This led us to consider two types of grip.

Our first design, *Torch*, builds on the metaphor of a torch light (Figure 4-Left). The thumb is not necessarily involved in the grip and is independent enough to activate a switch. As illustrated in Figure 4-Left, the cylinder-shape of a *Torch*'s handle is completely symmetric and can be grasped in any orientation. While this gives the user flexibility in how they grab and hold the controller, it also means that the switch must be such that it can be activated regardless of how the user grabs the controller. To ensure that the switch can be activated in any orientation, we designed it as a gear rather than a simple push button. This gear is connected to the base with a spring so that users can rotate it with their thumb tip to switch on or to switch off. The effort that is necessary to activate this gear-like switch is minimal as it requires only a rotation of small amplitude ($\sim 15°$) for the system to detect a change in switch state reliably. The direction of the force that users apply is around the

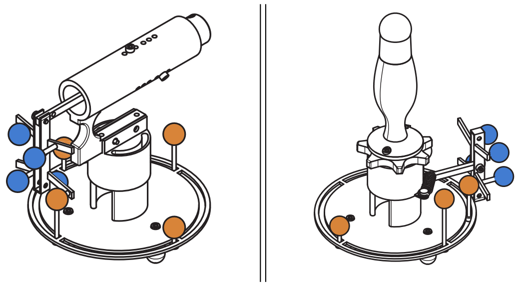


Figure 5: Infrared marker positioning for tracking the position of a *SurfAir* and the on/off state of its switch for the *Handle SurfAir* (Left) and *Torch SurfAir* (Right).

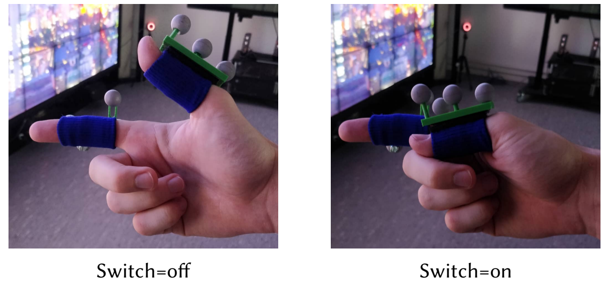


Figure 6: Bare-hand input is implemented as a raycast that follows the direction of the user's index. (Left) Typical posture for Switch=off. (Right) SideTrigger posture for Switch=on.

main axis of the controller, which likely minimizes unintentional deviations of the controller when activating the switch. This *Torch* design has thus the advantage of being orientation-independent. In particular, it affords two types of grip. In the air, it seems natural to hold the controller as a torch but, when on the surface, the long handle can also afford a pen-like grip.

Our second design, *Handle*, affords a door-handle-like grip (Figure 4-Right). Here again, the thumb is not necessarily involved in the grip. We can take advantage of its independence from the other fingers [19, 52] to push a simple spring-mounted button that is located on one end of the handle. Contrary to the *Torch* design, the *Handle* design is orientation-dependent, and affords the same grip for both on-surface and mid-air interaction.

4.3 Tracking and Recognition

We tested our *SurfAirs* on a wall display equipped with a multi-touch PQLabs© frame. The room also features a Vicon© motion-capture system with 20 cameras. We chose this motion-capture system not only because it is optimized for our experimental setup (room and display) but also and mainly because it is able to track both *SurfAirs* and bare-hand input, eliminating the tracking technology as a confounding factor in the studies that we report in the following sections.

Like several tangibles for on-surface interaction (*e.g.*, [12, 62, 68]), a *SurfAir* generates a multi-touch pattern when in contact with the tactile surface. Each *SurfAir* is mounted on three feet whose spatial configuration is specific so that a *SurfAir* can be recognized when in contact with a multi-touch surface using the simple pattern-matching algorithm described in [12, 46].

As illustrated in Figure 5, *SurfAirs* are equipped with two constellations of infrared markers that we can track with the motion-capture system in the air. The controller's base features a rail where infrared markers can be attached in flexible configurations to define the first constellation (orange-colored markers in Figure 5). We rely on this constellation to track the 3D position and orientation of a *SurfAir*, from which we can obtain a raycast. The second constellation of four markers is attached to a movable part of the *SurfAir* (blue-colored markers in Figure 5). This constellation's relative position to the base depends on whether the user activates the switch or not. This allows us to recognize the state of the switch (on or off) based on the distance between the two constellations of markers.

5 *SURFAIRS* VS BARE HAND FOR MID-AIR INPUT

Before evaluating hybrid input, we test the performance of our *SurfAirs* for performing elementary pointing and docking tasks from a distance. We run a comparative study between our two *SurfAir* prototypes and assess their performance using bare-hand input as a baseline. Although this first study does not involve input on the surface itself, our ultimate goal, which we test in the following study, is to support hybrid input. Bare-hand input is therefore the most relevant baseline to consider as it is the only single-handed input technique in the literature that allows for both surface and mid-air interaction while also enabling smooth transitions between those two [54].

In this experiment, participants perform pointing and docking tasks using a controller, which can be one of the two *SurfAirs* or their bare hand, depending on the condition. In all cases, users can point thanks to a raycast originating from the controller/hand, and rotate by adjusting the orientation of the controller/hand. They can select by clicking using either the *SurfAir*'s switch or by changing their hand posture. In the bare-hand condition, we use the *SideTrigger* gesture [4, 31] for clicking (Figure 6). Our general hypothesis is that *users will perform better with SurfAirs than with bare-hand input* ($H_{general}$). Although the *SideTrigger* gesture [4, 31] has proven good enough in recent studies to achieve remote selections, we hypothesize that participants will feel *more confident* when activating a mechanical switch than when switching between hand postures ($H_{confidence}$). This is not only because a button constrains the possibilities to two states only, but also because it is less subject to variations across users. In comparison, the space of hand postures is much less constrained and more user-dependent. As a result, the system's recognition and tracking performance will likely be better with *SurfAirs* than with bare-hand input. Second, a *SurfAir* provides haptic support which should help users be more stable when clicking and thus enable *more precision* in selection actions ($H_{precision}$). Finally, we expect the haptic feedback provided by a *SurfAir* to also reduce the sensation of fatigue in comparison with maintaining hand postures ($H_{fatigue}$).

5.1 Design and Procedure

Participants. Nine volunteers (8 men and 1 woman), all right-handed, aged 24 to 44 year-old (average 29, median 26), participated in the experiment.

Apparatus. The experiment runs in full screen mode on a cluster-driven wall-sized display (75 ultra-thin bezel screens tiled in a 15 × 5 grid, resulting in a total surface of 5m90 × 1m95 for a resolution of 14 400 × 4 800 pixels). The experimental software was developed using Unity 3D (version 2018.3).

As illustrated in Figure 6, we track hand postures using the exact same hardware and approach as the one we use for *SurfAirs*. In bare-hand input conditions, participants wear two finger sleeves equipped with constellations of infrared markers. We use the distance between these two constellations to recognize the switch between the two hand postures.

General Procedure and Design. We follow a within-subject design for primary factor INPUT = {*Hand*, *Handle*, *Torch*}. The experiment consists of two phases: *Point&Click* and *Docking*, always presented in this order and separated by at least 24, and at most 48, hours. In both phases, trials are blocked by INPUT, and the presentation order of these INPUT blocks is counterbalanced across participants using a Latin Square.

Before starting the experiment, participants have to sign a consent form after the operator has explained the general procedure and goal of the experiment.

At the beginning of each INPUT-block, participants have to put their right foot on a marker on the ground placed at a distance of 2 meters from the center of the wall. The operator then introduces the input technique to be used in this block. In the specific case of *Hand*, the block starts with a calibration procedure in order to account for the variability across different hand anatomies. The operator then explains the task before completing a series of sub-blocks. The first sub-block is for training purposes. As detailed below, the number of measured sub-blocks depends on the phase.

Between each INPUT-block, participants must sit and rest their arm until any feeling of fatigue disappears. During this break, they fill in a questionnaire where they have to rate the INPUT condition that they have just used along the following aspects, with 5-point Likert scales: easiness, confidence, physical demand and mental demand. In the *Point&Click* phase, they rate the click gesture and the pointing action separately before giving a final performance rating for the entire task.

At the end of each phase, the participants fill in a global questionnaire where they rank the three different INPUT techniques along the following aspects: physical demand, mental demand, cumbersomeness and preference. The operator also collects participants' informal feedback. Each phase lasts about 1 hour.

Phase 1: Point&Click. The first phase is a classic pointing experiment. It consists of clicking a series of 8 circular targets of the same size. The distance between two successive targets in a series is constant. A click outside a target is counted as an error but participants have to continue the task until they successfully select the target. Each INPUT-block consists of 3 sub-blocks, each featuring 6 series of pointing tasks: 2 DIST × 3 WIDTH. Following Jota *et al.*'s recommendations for raycast-based techniques [36], we use angular size and angular distance. Each participant experiences two values for DIST ({20° and 90°})[1] and three for WIDTH ({1.8°, 3.6° and 5.4°}).[2] The presentation order of the 6 series within a sub-block is random. The first sub-block is used for practice.

[1]86cm and 490cm on screen.

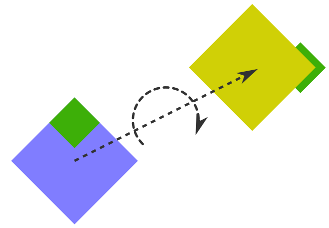


Figure 7: Experimental task in the *Docking* phase. Participants have to drag and rotate the *Modulus*, a blue square of 40 *cm* **side, into the *Stimulus*, an orange square positioned at** 90 *cm* **of the *Modulus*. A square's orientation is indicated by a green mark in one of the corners. In this example:** DIRECTION **= *North-East* and** ROTATION **= +90°.**

This design results in 2268 measured pointing tasks in total: 9 participants × 3 INPUT × 3 WIDTH × 2 DIST × 2 sub-blocks × 7 pointing tasks (the first pointing task is ignored as the cursor's initial location is not controlled).

Phase 2: Docking. The second sub-experiment is a docking experiment. As illustrated in Figure 7, participants have to manipulate a virtual object (the *Modulus*) to make its position and orientation match that of a target placeholder (the *Stimulus*). A trial starts with the two objects displayed on screen: the *Modulus* as a blue square in the middle of the screen, and the *Stimulus* as an orange square. The *Stimulus* is placed at constant distance of the *Modulus* (90 *cm*) in one DIRECTION among the following: *North-West*, *North-East*, *South-West* or *South-East*. Participants have to drag and rotate[3] the *Modulus* over the *Stimulus*. As soon as the position and orientation conditions are met, a blue ring starts to fill up. The ring is full when both conditions have been maintained for 1000 ms (dwell), effectively ending the trial. The experiment software has some tolerance in both orientation and position: 10° in orientation and 5cm (about 1°) in distance. The initial difference in orientation between the *Modulus* and *Stimulus* (ROTATION factor) is either 0°, -90° (counter-clockwise), 90° (clockwise) or 180°. Each INPUT-block consists of a training sub-block and three measured sub-blocks, each featuring 16 trials (4 ROTATION × 4 DIRECTION) presented in a random order.

This design results in 1296 measured docking tasks in total: 9 participants × 3 INPUT × 4 ROTATION × 4 DIRECTION × 3 sub-blocks.

5.2 Results: Point & Click

Pointing Time. We remove 17 points from our collection of 2268 data points: 10 outliers (6 *Hand*, 3 *Torch*, 1 *Handle*) and 7 trials in which participants experienced tracking issues (5 *Torch*, 2 *Hand*). We then run a repeated-measures factorial ANOVA with Greenhouse-Geisser correction for sphericity, and Bonferroni-Holm corrected paired post-hoc t-tests. Figure 8 illustrates our results. In all our bar plots, an error bar represents the 95% confidence interval relative to all the data points collected in the corresponding condition.

[2]For the 20° angular distance, the 1.8° target is 8cm wide, the 3.6° target is 16cm wide and the 5.6° target is 24cm wide. For the 90° angular distance, the 1.8° target is 15cm wide, the 3.6° target is 30cm wide and the 5.6° target is 45cm wide.

[3]To enable wide-angle rotations while remaining within a reasonable range of motion, the *Modulus* rotates twice as fast as the hand or *SurfAir*.

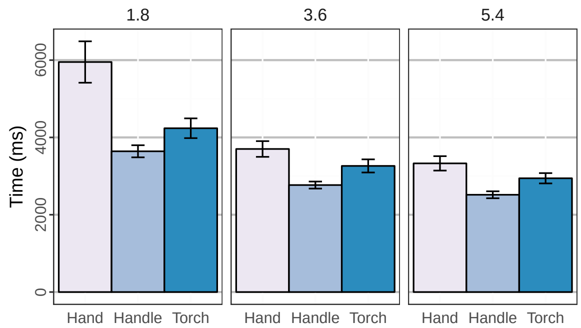


Figure 8: Pointing time by INPUT × WIDTH.

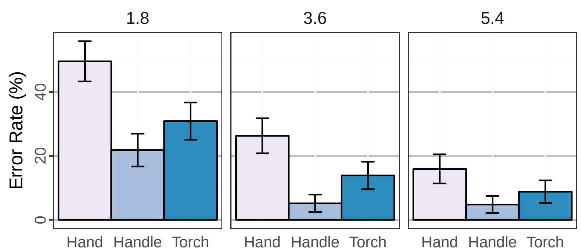


Figure 9: Error rate by INPUT × WIDTH.

Our primary INPUT factor has a significant effect on pointing time ($F_{1.2,9.7} = 17.5$, $p = 0.001$, $\eta^2_G = 0.22$), and post-hoc tests show that *Handle* is faster than both *Hand* ($p = 0.006$, $d = 1.48$) and *Torch* ($p = 0.007$, $d = 0.84$), and that *Torch* is faster than *Hand* ($p = 0.007$, $d = 0.81$).

As expected, the ANOVA test also reveals significant effects of both WIDTH ($F_{1.1,8.5} = 33.4$, $p < 0.001$, $\eta^2_G = 0.33$) and DIST ($F_{1,8} = 33.7$, $p < 0.001$, $\eta^2_G = 0.12$) on pointing time, with participants being significantly faster when WIDTH increases (p's < 0.002) or when DIST decreases ($p < 0.001$). Moreover, we have a significant INPUT × WIDTH interaction effect[4] ($F_{1.3,10.0} = 8.22$, $p = 0.013$, $\eta^2_G = 0.08$): *Handle* is faster than *Hand* and *Torch* for each WIDTH, but *Torch* is significantly faster than *Hand* only when WIDTH is small.

Error Rate. We notice an unexpectedly high error rate for *Torch* even for the larger targets (26% for 5.4°-large targets). Looking at the event-level logs, many of these errors with *Torch* occur either at the beginning or in the middle of the pointing movement. We believe that these errors actually reflect tracking issues. Indeed, the switch mechanism of a *Torch* is along the direction of movement and might have been accidentally triggered because of the movement's acceleration. Such errors should not have an impact on pointing performance. In order to focus on errors that can actually impact pointing performance, we filter out errors to consider only those that are close enough to the target (> $\frac{2}{3}$ of the distance) to be actual selection errors. This filtering operation significantly decreases the error rate for *Torch*, and marginally reduces it for *Handle* and *Hand*.

Figure 9 shows this corrected error rate. As participants had to continue with the current pointing task in case of an error, it is not surprising to observe effects that are similar to the ones we observed on *Pointing Time*. We observe a significant effect of INPUT ($F_{2,16} = 38$, $p < 0.001$, $\eta^2_G = 0.28$), with *Handle* having a lower error rate than both *Hand* ($p < 0.001$, $d = 1.95$) and *Torch* ($p = 0.013$, $d = 0.75$), and *Torch* having a lower error rate than *Hand* ($p = 0.001$, $d = 1.05$). Unsurprisingly, the target's WIDTH ($F_{2,16} = 28.6$, $p < 0.001$, $\eta^2_G = 0.38$) has a significant effect as well: the error rate decreases as the WIDTH increases ($p < 0.001$ and $p = 0.051$). Finally, we observe a significant INPUT × WIDTH interaction effect but with a small effect size ($F_{4,32} = 3.77$, $p = 0.013$, $\eta^2_G = 0.04$).

These results support $H_{precision}$: *SurfAirs* enable higher selection precision than bare-hand input does. Looking at the movement deviation during a click action, we observe that participants were actually more stable when clicking with a *SurfAir* than when switching between two free-hand postures. The average absolute angle deviation between press and release events of a successful click is 0.45°±0.03° for *Handle*, 0.58°±0.03° for *Torch*, and 0.73°±0.04° for *Hand*, all pairs being significantly different (p's < 0.025).

Subjective Feedback. There is no significant difference between the INPUT conditions for questions related to the pointing phase (we use paired Wilcoxon signed-rank tests). Overall, participants performed this part of the task with confidence (4.30 ± 0.67), and found it easy to perform (4.22 ± 0.70) with low mental and low physical demand (1.22 ± 0.42 and 2.07 ± 0.92). However, participants found it easier to click with *Handle* and *Torch* than with *Hand* (4.56 ± 1.01 and 4.78 ± 0.44 vs. 2.89 ± 1.17, $p = 0.031$ and $p = 0.023$). They were also more confident with *Handle* and *Torch* than with *Hand* (4.56 ± 0.73 and 4.44 ± 0.53 vs. 2.33 ± 1, p's $= 0.012$). The differences regarding physical and mental demand were not significant (1.85 ± 1.2 and 1.33 ± 0.73). This resulted in participants feeling that they were performing better with *Handle* and *Torch* than with *Hand* (4.56 ± 0.73 and 4.56 ± 0.53 vs.2.89 ± 1.17, p's $= 0.012$).

Regarding global rankings, all participants ranked either *Handle* (5 participants) or *Torch* (6 participants, 2 *ex-aequo* ranking) as their preferred technique, and all participants ranked *Hand* last (differences in ranking are significant: p's $= 0.012$). Differences in ranking regarding physical demand and cumbersomeness are not significant, but they are significant regarding mental load, with *Handle* and *Torch* ranked better than *Hand* (p's $= 0.012$).

5.3 Results: Docking

Docking Time. We remove 20 points from our collection of 1296 data points: 6 outliers (3 *Hand*, 2 *Torch*, 1 *Handle*) and 14 trials where participants experienced tracking issues (8 *Torch*, 3 *Hand*, 3 *Handle*). We then run a repeated-measures factorial ANOVA. Figure 10 illustrates our results.

The ANOVA reveals a significant effect of INPUT ($F_{2,16} = 21.4$, $p < 0.001$, $\eta^2_G = 0.28$) on *Docking Time*. *Handle* is significantly faster than

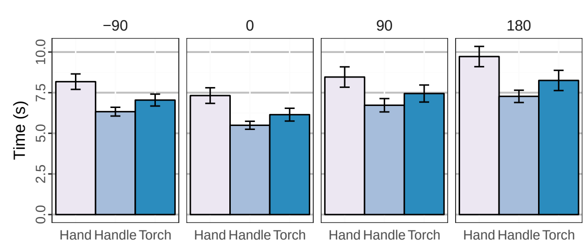


Figure 10: Docking time by INPUT × ROTATION.

[4]Other interaction effects are not significant.

both *Hand* ($p < 0.001$, $d = 1.6$) and *Torch* ($p = 0.037$, $d = 0.77$), and *Torch* is significantly faster than *Hand* ($p = 0.037$, $d = 0.93$). Unsurprisingly, ROTATION has a significant effect on time as well ($F_{3,24} = 21.8$, $p = 0.25$, $\eta^2_G = $), with participants being faster with 0° than with all other ROTATION angles (p's < 0.008). They were also faster with -90° than with 180° ($p = 0.007$). Since the timer starts as soon as the *Modulus* and *Stimulus* appear, *Docking Time* includes both the preparation time (the time taken to grab the modulus) and the manipulation time. Analyses on either the preparation time or the handling time in isolation lead to the same conclusions.

Clutching, Integrality & Simultaneity. About 92% of the trials have been performed without clutching, *i.e.*, in "one movement" without releasing the switch since the initial press to grab the *Modulus*. This percentage is high in all three conditions, without any significant differences between them. This suggests that the three techniques are adapted to control two dimensions (translation and rotation) in a single movement.

To better understand how both dimensions are manipulated by the participants, we computed the movement's *integrality* [27] for trials where ROTATION is not zero. For this purpose, we divide the drag motion until the instant all docking conditions are met for the first time into a series of 50ms intervals. For each of these intervals, we compute the difference in position and orientation, Δ_{pos} and Δ_{orient} (normalized in [0, 1]), these differences being positive if the differences in position or orientation between the *Stimulus* and the *Modulus* decrease. We classify each interval as either (i) *integral* if both Δ_{pos} and $\Delta_{orient} \geq th$; (ii) *stable* if $-th < \Delta_{pos} < th$ and $-th < \Delta_{orient} < th$; (iii) *separate* if neither integral nor stable. Then, we remove the stable intervals, and compute the percentage of the intervals that are integral.

With $th = 0.005$ (0.5% of the movement amplitude), the average integrality score is 44.6% ± 1.9 for *Handle*, 43.5% ± 1.9 for *Torch*, and 36.4% ± 1.7 for *Hand*. The difference between *Handle* and *Torch* is not significant, but both *Handle* and *Torch* have a significantly higher integrality score than *Hand* (p's < 0.002, $d \sim 1$).

Precision. The above results suggest that participants had better control with *Handle* and *Torch* than with *Hand* for both position and rotation in a single movement. Regarding the precision of that movement, we focus on the end of the task and analyze the number of overshoot errors, *i.e.*, the number of times the *Modulus* leaves its docking position after having met the conditions. On average, the number of overshoot errors is 0.47 ± 0.07 for *Torch*, 0.48 ± 0.07 for *Handle*, and 0.70 ± 0.09 for *Hand*, with *Hand* leading to significantly more overshoot errors than both *Torch* and *Handle* (p's < 0.008, $d \sim 1$). This suggests that participants have better stabilization and finer control abilities with *Handle* and *Torch* than with *Hand*.

Subjective Feedback. For this Docking phase, participants' subjective feedback is not significantly different across INPUT conditions. Overall, participants performed the task with confidence (4.5 ± 0.59). They found it easy to perform (4.19 ± 0.52), requiring low physical (2.11 ± 0.87) and mental (1.67 ± 0.76) demand. Regarding the global ranking between techniques, six participants ranked *Handle* first, 5 participants ranked *Torch* first, and 1 participant ranked *Hand* first (*ex-aequo* rankings were allowed).

5.4 Summary of Results

Overall results of this experiment support our main hypothesis ($H_{general}$): *SurfAirs* perform better and are preferred over bare-hand input. For both *Point&Click* and *Docking*, participants were more accurate with *SurfAirs* than with hand gestures ($H_{precision}$). In particular, participants were more stable when clicking with the mechanical switches of the *SurfAirs* than with hand gestures, which likely contributed positively to the confidence they had when interacting with a physical controller ($H_{confidence}$). Observations in the *Docking* phase also suggest that participants are better at controlling two dimensions concurrently with a physical controller than with their bare hand. Although both *SurfAir* prototypes outperformed bare-hand input, *Handle* seems to be a more promising design than *Torch*. Participants are very stable when clicking in the air with *Handle*, allowing for precise selection. However, while participants' informal feedback suggested more fatigue with bare-hand input than with *SurfAir*-based input, quantitative answers in the questionnaire does not support $H_{fatigue}$: participants did not find bare-hand input more tiring than *SurfAirs*.

6 *SURFAIRS* VS BARE HAND FOR BOTH ON SURFACE AND MID-AIR INPUT

In this second experiment, we evaluate *SurfAirs*' performance for tasks that involve both precise manipulations on screen and coarse manipulations in the air. We use a docking task to operationalize manipulations at these two levels of precision, as well as transitions between both. We conduct a comparative evaluation of our *Handle* prototype against bare-hand input (*Hand*). We do not include the *Torch* prototype in this second study for several reasons. First, we wanted to keep our experiment reasonably short for participants. Second, *Handle* performed slightly better than *Torch* in our first experiment. Finally, as opposed to *Torch*, the grip of a *Handle* is the same whether held on a surface or in the air.

The docking task consists of adjusting the orientation, scale and position of a virtual square (*Modulus*) to make it match the spatial configuration of another square (*Stimulus*). The translation to perform is fairly large (*i.e.*, 160cm), and participants are instructed to interact from afar (*mid-air*) to perform it. On the opposite, they are instructed to interact directly on screen (*surface*) for rotating and scaling the modulus. In order to operationalize the case where users have to do both types of interactions in a single and continuous chunk, participants are also instructed to switch from the surface to the air (or vice versa) during the translation manipulation without releasing control (*hybrid*).

6.1 Interaction

Mid-air interaction. As in the first experiment, participants can point with a raycast that departs either from the index finger (*Hand*) or from the controller (*Handle*). Participants can grab an object by adopting a SideTrigger posture (*Hand*) or by pressing the button (*Handle*) and then drag the object.

Surface interaction. In the *Hand* condition, participants use standard multi-touch gestures: one finger slides the *Modulus* (whether the hand adopts a SideTigger or a released posture), and two fingers both rotate (according to the orientation of the segment defined

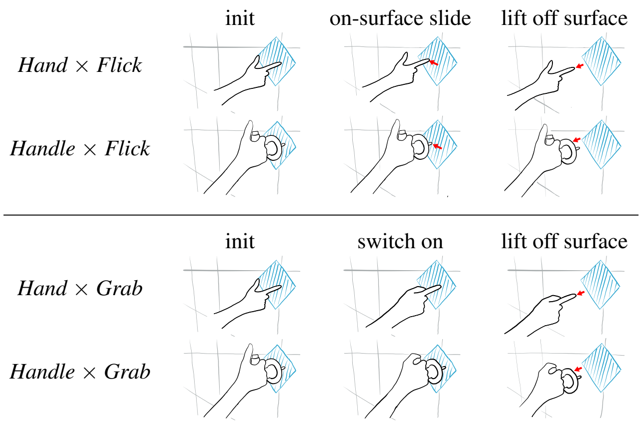


Figure 11: Surface to mid-air transition techniques: *Flick* and *Grab*. With the *Flick* technique, users slide on the surface before lifting their hand/controller off the surface. With the *Grab* technique, users activate the switch (SideTrigger posture or button press) before lifting their hand/controller off the surface.

by the two contact points) and scale (pinch). However, when two fingers are in contact, the translation is disabled. Pilot tests showed that it was very difficult for participants to adjust the rotation and scale without unintentionally moving the object. Participants can still translate on surface by simply lifting one of their two fingers off the surface. Similarly, in the *Handle* condition, the translation is disabled when the button is released. Participants can adjust the *Modulus* orientation by rotating the controller and its scale by sliding the controller up or down. To adjust the *Modulus* position, they must explicitly enter the dragging state by pressing the controller's button.

Mid-air/Surface transition. Users simply put either the controller (*Handle*) or their finger (*Hand*) on the surface to transition from mid-air to surface control. Participants can drag an object mid-air using ray-casting, then come in contact with the surface where they can still drag the object until they release the controller's button (*Handle*) or put a second finger on the surface (*Hand*). They can then continue interacting (rotate and scale).

Surface/Mid-air transition. This type of transition is more elaborate as the system must distinguish between releasing control and continuing the ongoing interaction. We consider two techniques, *Grab* and *Flick* (Figure 11), to leave the surface without releasing control. With *Grab*, participants have to activate the switch (SideTrigger posture or button press) while being in contact with the surface and maintain the switch on when they leave the surface. They then release control by releasing the switch. With *Flick*, participants must perform a flick gesture (*i.e.*, accelerating the movement as in [54]) when they leave the surface to continue with the ongoing interaction in the air. The flick can be performed with the switch either on or off. They then release control when the switch state changes (on → off or off → on). This offers more flexibility and avoids relying on clicks, which caused precision issues with *Hand* in our first experiment.

6.2 Tasks

Participants had to perform two types of tasks, either transitioning from the air to the surface or the opposite. In both cases, participants are instructed not to release control during the transition. For *AirToSurface*, participants must hold (drag) the *Modulus* when they come in contact with the surface. For example, they cannot drop the *Modulus* over the *Stimulus* using mid-air interaction and then come close to the wall for docking it. For *SurfaceToAir*, once the *Modulus* is docked, participants must use either the *Grab* or *Flick* technique before stepping back.

AirToSurface. The task starts with the *Stimulus* displayed in the center of the wall, and the *Modulus* (a square of 30 *cm* side) located at 160 *cm* on the left or the right of the *Stimulus*. Participants have to face the center of the wall, and be at a distance of at least 150 *cm* from it. Participants have to grab the *Modulus* and drag it over the *Stimulus* using mid-air interaction, while at the same time coming close to the wall to eventually come in contact with it. When in contact, they can precisely adjust the *Modulus* position (tolerance = 7cm) and dock the *Modulus* in the *Stimulus*: rotate (tolerance = 10°) and scale (tolerance = 10%). The task is validated once the docking conditions have been maintained for 1s.

SurfaceToAir. The task starts with both the *Modulus* and the *Stimulus* displayed in the center of the wall. Participants have to first dock the *Modulus* into the *Stimulus* on surface (like they do in the second part of an *AirToSurface* task). They then leave the surface without releasing control (using either *Flick* or *Grab* depending on the condition as detailed in the design below). This makes a second *Stimulus* appear on the right or the left at 160 *cm* from the center of the wall. They must drop the *Modulus* over this second *Stimulus* with a tolerance of 7*cm*. The second *Stimulus* appears only when the controller or hand is at 150 *cm* from the wall in order to operationalize the case where users decide on where to position an object only after they have a wider view angle.

6.3 Hypotheses

We formulate the following two hypotheses:

$H_{general}$: *SurfAir*-based input outperforms bare-hand input overall. Our first experiment suggests that *SurfAirs* perform better for mid-air input. In addition, studies reported in [12] suggest that tangibles act as better surface controllers than multi-touch gestures. As for the transition, we do not expect to observe strong differences between the two types of input. As a result, we expect a *SurfAir* to perform better than bare-hand input overall.

$H_{transition}$: For transitioning from the surface to the air, we hypothesize that *Flick* is a better technique for *Hand*, while *Grab* is better for *Handle*. This is because 1) state switching is more costly for *Hand* than it is for *Handle* (as suggested by our first experiment) and 2) a *SurfAir* is more rigid and has a larger contact surface than a finger, which could make the *Flick*'s accelerating gesture more difficult to perform.

6.4 Experimental Design and Procedure

Participants & Apparatus. Twelve volunteers (5 men and 7 women), all right-handed, aged 21 to 32 year-old (average 25.33, median 25), participated in the experiment. We use the same apparatus as in the

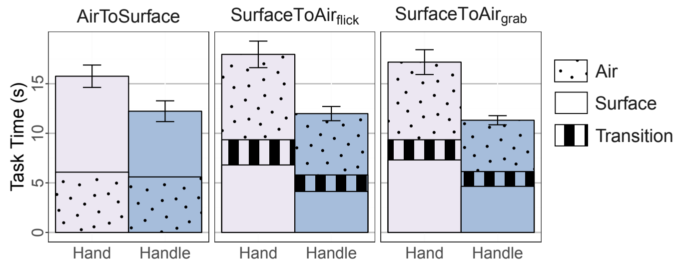


Figure 12: Task time by INPUT for each task.

	AirToSurface	*SurfaceToAir*
INPUT	$F_{1,11} = 22.7$, $p < 0.001$, $\eta^2_G = 0.23$	$F_{1,11} = 70.1$, $p < 0.001$, $\eta^2_G = 0.49$
TECH	-	$F_{1,11} = 1.01$, $p = 0.336$, $\eta^2_G = 0.01$
INPUT × TECH	-	$F_{1,11} = 0.01$, $p = 0.918$, $\eta^2_G = 0.00$

Table 1: Results of ANOVA tests for task time.

first experiment. The wall display is equipped with a multi-touch PQLabs© frame, which we rely upon for on-surface interaction.

Design and Procedure. We follow a within-subject design with primary factor INPUT = {*Hand*, *Handle*}. Trials are blocked by INPUT. With each INPUT, participants had to perform both *AirToSurface* and *SurfaceToAir* types of transition. We also test the two different techniques for transitioning from the surface to the air (TECH = {*Grab*, *Flick* }). We thus have three types of tasks: *AirToSurface*, $SurfaceToAir_{flick}$ and $SurfaceToAir_{grab}$. Each INPUT-block is a series of three sub-blocks, one per task. The presentation order of blocks and sub-blocks is counterbalanced across participants.

At the beginning of the experiment, participants sign a consent form after having read the general procedure and goal of the experience. Each task sub-block starts with the operator explaining how to perform the task. Participants then perform 16 trials = 2 repetitions (one training, one measured) × 2 rotations (-90°, 90°) × 2 scales (-50%, +50%) × 2 directions (right, left). The presentation order of the 8 trials within a repetition is random.

Between two task conditions, participants must sit and rest. During that break, the operator asks them to rate on a 5-point Likert scale the condition that they have just experienced regarding easiness, confidence, physical demand, mental demand and performance. They rate not only the task in general but also the transition specifically. Moreover, at the end of the experiment, participants rank the two INPUT conditions and the transition techniques.

The whole experiment lasts about 75 minutes. Answering the questionnaire represents about half of this time.

6.5 Results

Task Time. Among the 576 measured trials, we remove 9 data points where we experienced logging issues and 7 outliers based on a linear analysis (all from *Hand*). After checking the normality of our data, we run two ANOVA tests: 1) INPUT ∼ *time* for *AirToSurface* tasks and 2) INPUT × TECH ∼ *time* for *SurfaceToAir* tasks. Figure 12 illustrates the different effects that we observe, and Table 1 details the test results. First, *Handle* significantly outperforms *Hand* with a large effect size, and this overall difference is not impacted by the transition technique (TECH) for *SurfaceToAir* tasks.

Figure 12 shows a breakdown of the total Task *time* into: the time on the surface, the time in the air, and the transition time for *SurfaceToAir* tasks (*i.e.*, the interval between the moment the docking task is completed and the moment the *Hand*/*Handle* leaves the surface with a successful use of either *Flick* or *Grab*). We analyze each phase of the task below.

Transition. An ANOVA INPUT × TECH on the transition time reveals: (i) a significant effect of INPUT ($F_{1,11} = 17.9$, $p = 0.001$, $\eta^2_G = 0.28$), with *Handle* being faster than *Hand* (1578 ± 126 *ms vs.* 2291 ± 158 *ms*); (ii) a marginal effect of TECH ($F_{1,11} = 4.70$, $p = 0.054$, $\eta^2_G = 0.13$), with *Grab* being faster than *Flick* (1764 ± 118 *ms vs.* 2109 ± 177 *ms*); and (iii) no INPUT × TECH interaction effect ($F_{1,11} = 3.60$, $p = 0.084$, $\eta^2_G = 0.02$).

An analysis of errors can partially explain these differences. A transition error happens when participants fail at transitioning from the surface to the air at their first attempt with either *Flick* or *Grab*. The following table reports the percentage of trials where such errors occur:

Flick		Grab	
Hand	*Handle*	*Hand*	*Handle*
17.0% ± 7.6%	7.6% ± 5.4%	5.4% ± 4.6%	2.2% ± 3.0%

First, we observe that we have more transition errors with *Flick* than with *Grab* ($p = 0.045$, $d = 0.9$). This might be specific to the case that we operationalize in our experiment, with participants needing a wide viewing angle to decide on where to place the *Modulus*. As they do not know the direction of their future movement when they leave the surface, they make an arbitrary choice regarding the direction of their flick gesture. This could have played against the *Flick* technique. Second, we observe more transition errors with *Hand* than with *Handle* ($p = 0.037$, $d = 0.8$) for both transition techniques. Contrary to our hypothesis about *Flick* being better suited for *Hand* and *Grab* for *Handle* ($H_{transition}$), participants consistently performed better transitions with *Handle* than with *Hand* whatever the transition technique considered.

Surface. For the time spent on the surface (*i.e.*, time for docking), the comparison between *Hand* and *Handle* is very similar to what it is for the total task time. This supports results from previous studies [12], where tangibles were better than multi-touch gestures when used as surface controllers. We looked at some specific lower-level events to better interpret those observations. The table below reports the percentage of trials where participants (i) had to perform at least one *Clutch* action during the docking on surface; and (ii) entered and then left the target docking position (*enter/leave*):

	AirToSurface		*SurfaceToAir*	
	Hand	*Handle*	*Hand*	*Handle*
Clutch	89.1% ± 6.4%	19.8% ± 8.1%	67.4% ± 6.8%	3.2% ± 2.6%
enter/leave	55.4% ± 10.3%	39.6% ± 9.9%	59.3% ± 7.1%	24.9% ± 6.3%

There are significantly fewer clutch actions with *Handle* than with *Hand*, suggesting a more continuous control with *Handle*.for *AirToSurface* tasks than with *SurfaceToAir* tasks (going down to 3.2% of trials with *Handle*). This is probably because participants came in contact while maintaining a SideTrigger posture. Putting the thumb down from this posture led to uncomfortable positions. Participants then tended to lift off their finger and reposition their hand in order to make future on-surface manipulations more comfortable. Finally, in the *Handle* condition, participants performed less enter/leave actions than in the *Hand* condition, suggesting a better precision control and stability with *Handle*.

Mid-air. For the time spent in the air, *Handle* is significantly faster than *Hand* for the *SurfaceToAir* task ($F_{1,11} = 27.4$, $p < 0.001$, $\eta^2_G = 0.28$). However, for the *AirToSurface* task, the difference between *Hand* and *Handle* is not significant ($F_{1,11} = 0.92$, $p = 0.358$, $\eta^2_G = 0.01$). These different results can be easily explained by the nature of the tasks and their difficulty:

- In *AirToSurface* tasks, participants had no difficulty with either *Hand* or *Handle* to put the *Modulus* over the *Stimulus* with the required tolerance of 7 *cm* when coming in contact with the wall (participants almost never adjusted the position of the *Modulus* once on the surface).

- In *SurfaceToAir* tasks, participants had to release the *Modulus* over the *Stimulus* from afar with the same 7 *cm* tolerance. Consistently with our first experiment, it was more difficult with *Hand* (error rate 30%) than with *Handle* (error rate 10%). A larger tolerance would certainly lead to different results.

Subjective Feedback. Overall, participants were more confident with *Handle* (4.8 *vs.* 3.9, $p = 0.002$) and found that *Handle* was easier to use (4.5 *vs.* 3.5, $p = 0.002$), was less physically demanding (1.3 *vs.* 2.2, $p = 0.004$), and performed better (4.5 *vs.* 3.9, $p = 0.012$). The difference regarding mental demand is not significant (1.3 *vs.* 1.7, $p = 0.203$).

For the transition-specific questions, there are significant differences neither between the INPUT conditions nor between *Flick* and *Grab*. Overall, the participants performed transitions with confidence (4.6), found the transition techniques easy to use (4.4), with low physical and mental demands (1.4 and 1.3), and perceived their performance as good (4.5).

Regarding rankings, *Handle* was better than *Hand* for all participants. For *SurfaceToAir* tasks, *Handle* with *Grab* is always ranked first (7 participants) or second (5 participants). *Handle* with *Flick* has also very good rankings: first for 5 participants, and second for 5 other participants. Looking at *Hand* only, a majority of participants (8) ranked *Grab* before *Flick*. For rankings related to the transition action only, 8 participants ranked first *Handle* with *Grab*, 3 participants ranked *Handle* with *Flick* first, and 3 participants ranked *Hand* with *Flick* first (2 *ex-aequo*).

Overall, *Handle* obtained better subjective scores than *Hand*, and was preferred by the participants. However, we could not observe clear differences between the transition techniques.

6.6 Summary of results

Our results support ($H_{general}$): *SurfAirs* performed better and were preferred over bare-hand input for hybrid interactions involving both surface and mid-air control as well as transitions between both. Moreover, participants were able (with a low error rate) to transition from the surface to the air with both the *Grab* and *Flick* techniques. Although *Grab* performed slightly better than *Flick*, our observations do not support ($H_{transition}$): there is statistical evidence neither that *Grab* is better for *Handle* nor that *Flick* is better for *Hand*. Finally, contrary to our first experiment, participants found *SurfAir*-based input significantly less tiring than bare-hand input.

7 LIMITATIONS

As opposed to off-the-shelf technologies, experimental setups like our wall room are unique. Observations are thus dependent on the specificities of the setup. In particular, tracking accuracy depends on the motion-capture system, the number of cameras and their positioning in the room. In our case, we optimized tracking accuracy for the volume effectively used during the studies. However, we chose to implement all our conditions using the exact same tracking setup so that it cannot be a confounding factor in comparisons across conditions. This means that even if absolute numbers are likely to be different in another setup, the comparison between conditions should be the same.

The tracking accuracy might also be impacted by the specific PQLab© frame that we use for capturing multi-touch input. Such a frame is based on optics, meaning that the fingers that are in contact with the surface can be occluded by other parts of the hand. In addition, fingers that are very close to the screen may even be considered to be in contact with it. Bare-hand input might have been impacted more than *SurfAirs* by these issues. However, whatever the technology considered, the rigid structure of a controller reduces variability. Controller-based input is thus usually more resistant to technological imperfections.

We have tested a specific implementation of bare-hand input, using the SideTrigger gesture for clicking and the flick gesture for transitioning. We chose the SideTrigger gesture because both our personal experience and the literature indicate that it reaches good performance in terms of speed, stability and precision [4, 31]. We chose the flick gesture to transition for multiple reasons. Firstly, a flick gesture can be performed independently from the click state for each input technique. Secondly, a flick starts on the surface and finishes in the air, strengthening the metaphor of moving control from the surface to the air. However, alternative gestures within the very large space of multi-touch gestures could be considered and tested. Similarly, alternative handle and switch designs could be considered for a physical controller. But, a high-level property that remains independent from the design choices is that a gesture requires training or per-user calibration while a mechanical action on a physical controller does not.

Finally, participants in the two experiments were researchers, engineers, or graduate students in Computer Science. They were all familiar with multi-touch gestures from their experience with personal devices such as smartphones and tablets. However, except for two participants, they had no experience interacting with a wall display, and none of them had ever used multi-touch gestures on a wall display. Although post-experiment questionnaires suggest that participants were comfortable in all the experimental conditions, replicating the experiment with expert users may decrease the difference between bare-hand gestures and *SurfAirs* as bare-hand input might more benefit from learning effects. We also observe some anecdotal evidence that bare-hand input conditions may have been impacted by hand anatomy, with participants with large hands being more confortable with multi-touch gestures than participants with small hands. We did not observe such a tendency in *SurfAir* conditions, which reinforces the hypothesis that *SurfAirs*' performance is less user-dependent.

8 CONCLUSION AND FUTURE WORK

SurfAirs are physical controllers that can be tracked both on a surface and in the air. They enable controlling multiple degrees of freedom: translation and rotation on 2D surfaces, and full 6-DoF manipulation in the air. They are equipped with a switch that not only enables selection from afar but also enriches surface interaction with an additional state. For example, in our second experiment, we took advantage of this additional state to support translation, rotation and scaling on surfaces with a single controller. *SurfAirs* also feature mechanism based on a spring and suction cup [12] that makes them particularly well suited for interaction with vertical surfaces, as users can attach and detach them at will.

Our empirical studies compare the performance of *SurfAir*-based input against bare-hand input. Across the two studies, participants had to perform pointing and docking tasks in the air, on a surface, and across the air and a surface. *SurfAirs* performed better than, and were preferred to, bare-hand input. They enable steadier selections in the air, and more precise control both in the air and on-surface. In comparison with bare-hand input, they also have strong advantages by design. First, their manipulation is user-independent. Activating a switch with a change of hand posture is subject to both intra-user and inter-user variability as postures may vary over time and between users. In comparison, *SurfAirs* do not require any per-user calibration or pre-training. Second, *SurfAirs* do not require instrumenting the user, which is often cumbersome. For instance, even light instrumentation such as the finger sleeves we used (Figure 6) made it difficult for participants to fill out questionnaires during the study.

The fabrication of a *SurfAir* is relatively simple, relying on passive components, an optics-based multi-touch frame, and an optical tracking system that can accurately capture motion in 3D. Relying on modular components without any electronic connection was particularly convenient for testing alternatives during the design phase. For example, it allowed us to design the *Torch SurfAir* that features a switch that is orientation-independent. Such a switch would have been challenging to design with electronic components.

In our studies, we consider basic tasks with a single *SurfAir* in a specific technological setup. As future work, we would like to replicate our second experiment with other technologies such as a capacitive display or an electronic button for the *Handle SurfAir*. Future work should also evaluate *SurfAirs* with more ecological, high-level tasks that involve *e.g.*, a higher cognitive demand, multiple controllers and multiple users. Finally, it would be worth studying *SurfAirs* in the context of multi-display environments that can feature both vertical and horizontal displays such as tabletops.

SUPPLEMENTAL MATERIAL

Details on *SurfAirs* fabrication (*e.g.*, SDF files for the different components) and experimental data are available both as supplemental material and online at https://surfairs.lisn.upsaclay.fr.

ACKNOWLEDGMENTS

We wish to thank Emmanuel Pietriga and all the members of the Inria-LISN ILDA team for their feedback, Romain Di Vozzo and Jonah Ross-Marrs for their support at the Digiscope Fablab, and Olivier Gladin, Raphaël James and Eugenie Brasier for their help.

This research was supported by French government funding managed by the National Research Agency (ANR) under: (i) the Investments for the Future program (PIA) grant ANR-21-ESRE-0030 (CONTINUUM); and (ii) the Collaborative Research Project (PRC) grant ANR-21-CE33-0022 (InterPlay).

REFERENCES

[1] Benjamin Bach, Ronell Sicat, Johanna Beyer, Maxime Cordeil, and Hanspeter Pfister. 2018. The Hologram in My Hand: How Effective is Interactive Exploration of 3D Visualizations in Immersive Tangible Augmented Reality? *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (2018), 457–467. https://doi.org/10.1109/TVCG.2017.2745941

[2] Robert Ball and Chris North. 2008. The Effects of Peripheral Vision and Physical Navigation on Large Scale Visualization. In *Proceedings of Graphics Interface 2008 (GI '08)*. CIPS, 9–16. https://dl.acm.org/doi/abs/10.5555/1375714.1375717

[3] Robert Ball, Chris North, and Doug A. Bowman. 2007. Move to Improve: Promoting Physical Navigation to Increase User Performance with Large Displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. ACM, 191–200. https://doi.org/10.1145/1240624.1240656

[4] Amartya Banerjee, Jesse Burstyn, Audrey Girouard, and Roel Vertegaal. 2011. Pointable: An in-Air Pointing Technique to Manipulate out-of-Reach Targets on Tabletops. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS '11)*. ACM, 11–20. https://doi.org/10.1145/2076354.2076357

[5] Mark Billinghurst, Ivan Poupyrev, Hirokazu Kato, and Richard May. 2000. Mixing realities in Shared Space: an augmented reality interface for collaborative computing. In *International Conference on Multimedia and Expo. Proceedings. Latest Advances in the Fast Changing World of Multimedia (ICME '00, Vol. 3)*. IEEE, 1641–1644. https://doi.org/10.1109/ICME.2000.871085

[6] Eugenie Brasier, Emmanuel Pietriga, and Caroline Appert. 2021. AR-Enhanced Widgets for Smartphone-Centric Interaction. In *Proceedings of the 23rd International Conference on Mobile Human-Computer Interaction (MobileHCI '21)*. ACM, Article 32, 12 pages. https://doi.org/10.1145/3447526.3472019

[7] Alex Butler, Shahram Izadi, and Steve Hodges. 2008. SideSight: Multi-"touch" Interaction around Small Devices. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology (UIST '08)*. ACM, 201–204. https://doi.org/10.1145/1449715.1449746

[8] William Buxton. 1990. A Three-State Model of Graphical Input. In *Proceedings of the IFIP TC13 Third Interational Conference on Human-Computer Interaction (INTERACT '90)*. North-Holland, 449–456. https://www.dgp.toronto.edu/OTP/papers/bill.buxton/3state.html

[9] Olivier Chapuis, Anastasia Bezerianos, and Stelios Frantzeskakis. 2014. Smarties: An Input System for Wall Display Development. In *Proceedings of the SIGCHI*

Conference on Human Factors in Computing Systems (CHI '14). ACM, 2763–2772. https://doi.org/10.1145/2556288.2556956

[10] Christian Cherek, David Asselborn, Simon Voelker, and Jan Borchers. 2019. Off-Surface Tangibles: Exploring the Design Space of Midair Tangible Interaction. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems (CHI EA '19)*. ACM, Article LBW2115, 6 pages. https://doi.org/10.1145/3290607.3312966

[11] Maxime Cordeil, Benjamin Bach, Andrew Cunningham, Bastian Montoya, Ross T. Smith, Bruce H. Thomas, and Tim Dwyer. 2020. Embodied Axes: Tangible, Actuated Interaction for 3D Augmented Reality Data Spaces. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20)*. ACM, 1–12. https://doi.org/10.1145/3313831.3376613

[12] Emmanuel Courtoux, Caroline Appert, and Olivier Chapuis. 2021. WallTokens: Surface Tangibles for Vertical Displays. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)*. ACM, Article 421, 13 pages. https://doi.org/10.1145/3411764.3445404

[13] D. Englmeier, J. Dörner, A. Butz, and T. Höllerer. 2020. A Tangible Spherical Proxy for Object Manipulation in Augmented Reality. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 221–229. https://doi.org/10.1109/VR46266.2020.00041

[14] Barrett Ens, Sarah Goodwin, Arnaud Prouzeau, Fraser Anderson, Florence Y. Wang, Samuel Gratzl, Zac Lucarelli, Brendan Moyle, Jim Smiley, and Tim Dwyer. 2021. Uplift: A Tangible and Immersive Tabletop System for Casual Collaborative Visual Analytics. *IEEE Transactions on Visualization and Computer Graphics* 27, 2 (2021), 1193–1203. https://doi.org/10.1109/TVCG.2020.3030334

[15] George W. Fitzmaurice and William Buxton. 1997. An Empirical Evaluation of Graspable User Interfaces: Towards Specialized, Space-Multiplexed Input. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '97)*. ACM, 43–50. https://doi.org/10.1145/258549.258578

[16] Mike Fraser, Michael R. McCarthy, Muneeb Shaukat, and Phillip Smith. 2007. Seconds Matter: Improving Distributed Coordination Bytracking and Visualizing Display Trajectories. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. ACM, 1303–1312. https://doi.org/10.1145/1240624.1240822

[17] Bernd Fröhlich and John Plate. 2000. The Cubic Mouse: A New Device for Three-Dimensional Input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '00)*. ACM, 526–531. https://doi.org/10.1145/332040.332491

[18] Tovi Grossman, Daniel Wigdor, and Ravin Balakrishnan. 2004. Multi-Finger Gestural Interaction with 3d Volumetric Displays. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology (UIST '04)*. ACM, 61–70. https://doi.org/10.1145/1029632.1029644

[19] Charlotte Häger-Ross and Marc H. Schieber. 2000. Quantifying the Independence of Human Finger Movements: Comparisons of Digits, Hands, and Movement Frequencies. *Journal of Neuroscience* 20, 22 (2000), 8542–8550. https://doi.org/10.1523/JNEUROSCI.20-22-08542.2000

[20] Mark Hancock, Otmar Hilliges, Christopher Collins, Dominikus Baur, and Sheelagh Carpendale. 2009. Exploring Tangible and Direct Touch Interfaces for Manipulating 2D and 3D Information on a Digital Table. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS '09)*. ACM, 77–84. https://doi.org/10.1145/1731903.1731921

[21] Khalad Hasan, David Ahlström, Junhyeok Kim, and Pourang Irani. 2017. AirPanes: Two-Handed Around-Device Interaction for Pane Switching on Smartphones. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, 679–691. https://doi.org/10.1145/3025453.3026029

[22] Kirstie Hawkey, Melanie Kellar, Derek Reilly, Tara Whalen, and Kori M. Inkpen. 2005. The Proximity Factor: Impact of Distance on Co-Located Collaboration. In *Proceedings of the 2005 International ACM SIGGROUP Conference on Supporting Group Work (GROUP '05)*. ACM, 31–40. https://doi.org/10.1145/1099203.1099209

[23] Fabian Hennecke, Raphael Wimmer, Eduard Vodicka, and Andreas Butz. 2012. Vertibles: Using Vacuum Self-Adhesion to Create a Tangible User Interface for Arbitrary Interactive Surfaces. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction (TEI '12)*. ACM, 303–306. https://doi.org/10.1145/2148131.2148194

[24] Otmar Hilliges, Shahram Izadi, Andrew D. Wilson, Steve Hodges, Armando Garcia-Mendoza, and Andreas Butz. 2009. Interactions in the Air: Adding Further Depth to Interactive Tabletops. In *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology (UIST '09)*. ACM, 139–148. https://doi.org/10.1145/1622176.1622203

[25] Hiroshi Ishii and Brygg Ullmer. 1997. Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '97)*. ACM, 234–241. https://doi.org/10.1145/258549.258715

[26] Shahram Izadi, Steve Hodges, Stuart Taylor, Dan Rosenfeld, Nicolas Villar, Alex Butler, and Jonathan Westhues. 2008. Going beyond the Display: A Surface Technology with an Electronically Switchable Diffuser. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology (UIST '08)*. ACM, 269–278. https://doi.org/10.1145/1449715.1449760

[27] Robert J. K. Jacob, Linda E. Sibert, Daniel C. McFarlane, and M. Preston Mullen. 1994. Integrality and Separability of Input Devices. *ACM Trans. Comput.-Hum. Interact.* 1, 1 (March 1994), 3–26. https://doi.org/10.1145/174630.174631

[28] Mikkel R. Jakobsen and Kasper HornbÆk. 2014. Up Close and Personal: Collaborative Work on a High-Resolution Multitouch Wall Display. *ACM Trans. Comput.-Hum. Interact.* 21, 2, Article 11 (feb 2014), 34 pages. https://doi.org/10.1145/2576099

[29] Mikkel R. Jakobsen and Kasper Hornbæk. 2015. Is Moving Improving? Some Effects of Locomotion in Wall-Display Interaction. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, 4169–4178. https://doi.org/10.1145/2702123.2702312

[30] Mikkel R. Jakobsen and Kasper Hornbæk. 2016. Negotiating for Space? Collaborative Work Using a Wall Display with Mouse and Touch Input. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, 2050–2061. https://doi.org/10.1145/2858036.2858158

[31] Mikkel R. Jakobsen, Yvonne Jansen, Sebastian Boring, and Kasper Hornbæk. 2015. Should I Stay or Should I Go? Selecting Between Touch and Mid-Air Gestures for Large-Display Interaction. In *IFIP Conference on Human-Computer Interaction (INTERACT '15)*. Springer, 455–473. https://doi.org/10.1007/978-3-319-22698-9_31

[32] Mikkel R. Jakobsen, Yonas Sahlemariam Haile, Søren Knudsen, and Kasper Hornbæk. 2013. Information Visualization and Proxemics: Design Opportunities and Empirical Findings. *IEEE Transactions on Visualization and Computer Graphics* 19, 12 (2013), 2386–2395. https://doi.org/10.1109/TVCG.2013.166

[33] Yvonne Jansen, Pierre Dragicevic, and Jean-Daniel Fekete. 2012. Tangible Remote Controllers for Wall-Size Displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, 2865–2874. https://doi.org/10.1145/2207676.2208691

[34] Brett Jones, Rajinder Sodhi, David Forsyth, Brian Bailey, and Giuliano Maciocci. 2012. Around Device Interaction for Multiscale Navigation. In *Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '12)*. ACM, 83–92. https://doi.org/10.1145/2371574.2371589

[35] Sergi Jordà, Günter Geiger, Marcos Alonso, and Martin Kaltenbrunner. 2007. The ReacTable: Exploring the Synergy between Live Music Performance and Tabletop Tangible Interfaces. In *Proceedings of the 1st International Conference on Tangible and Embedded Interaction (TEI '07)*. ACM, 139–146. https://doi.org/10.1145/1226969.1226998

[36] Ricardo Jota, Miguel A. Nacenta, Joaquim A. Jorge, Sheelagh Carpendale, and Saul Greenberg. 2010. A Comparison of Ray Pointing Techniques for Very Large Displays. In *Proceedings of Graphics Interface 2010 (GI '10)*. CIPS, 269–276. https://doi.org/10.11575/PRISM/30630

[37] Hirokazu Kato, Mark Billinghurst, Ivan Poupyrev, K. Imamoto, and K. Tachibana. 2000. Virtual object manipulation on a table-top AR environment. In *Proceedings IEEE and ACM International Symposium on Augmented Reality (ISAR '00)*. IEEE, 111–119. https://doi.org/10.1109/ISAR.2000.880934

[38] Sven Kratz and Michael Rohs. 2009. HoverFlow: Expanding the Design Space of around-Device Interaction. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '09)*. ACM, Article 4, 8 pages. https://doi.org/10.1145/1613858.1613864

[39] Ricardo Langner, Ulrike Kister, and Raimund Dachselt. 2019. Multiple Coordinated Views at Large Displays for Multiple Users: Empirical Findings on User Behavior, Movements, and Distances. *IEEE Transactions on Visualization and Computer Graphics* 25, 1 (jan 2019), 608–618. https://doi.org/10.1109/TVCG.2018.2865235

[40] Jakob Leitner and Michael Haller. 2011. Geckos: Combining Magnets and Pressure Images to Enable New Tangible-Object Design and Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, 2985–2994. https://doi.org/10.1145/1978942.1979385

[41] Can Liu, Olivier Chapuis, Michel Beaudouin-Lafon, and Eric Lecolinet. 2017. CoReach: Cooperative Gestures for Data Manipulation on Wall-Sized Displays. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, 6730–6741. https://doi.org/10.1145/3025453.3025594

[42] Can Liu, Olivier Chapuis, Michel Beaudouin-Lafon, Eric Lecolinet, and Wendy E. Mackay. 2014. Effects of Display Size and Navigation Type on a Classification Task. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, 4147–4156. https://doi.org/10.1145/2556288.2557020

[43] Nicolai Marquardt, Ricardo Jota, Saul Greenberg, and Joaquim A. Jorge. 2011. The Continuous Interaction Space: Interaction Techniques Unifying Touch and Gesture on and above a Digital Surface. In *IFIP Conference on Human-Computer Interaction (INTERACT '11)*. Springer, 461–476. https://doi.org/10.1007/978-3-642-23765-2_32

[44] Fabrice Matulic and Daniel Vogel. 2018. Multiray: Multi-Finger Raycasting for Large Displays. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, 1–13. https://doi.org/10.1145/3173574.3173819

[45] David C. McCallum and Pourang Irani. 2009. ARC-Pad: Absolute+relative Cursor Positioning for Large Displays with a Mobile Touchscreen. In *Proceedings of the*

22nd Annual ACM Symposium on User Interface Software and Technology (UIST '09). ACM, 153–156. https://doi.org/10.1145/1622176.1622205

[46] Rafael Morales González, Caroline Appert, Gilles Bailly, and Emmanuel Pietriga. 2016. TouchTokens: Guiding Touch Patterns with Passive Tokens. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, 4189–4202. https://doi.org/10.1145/2858036.2858041

[47] Jörg Müller, Gilles Bailly, Thor Bossuyt, and Niklas Hillgren. 2014. MirrorTouch: Combining Touch and Mid-Air Gestures for Public Displays. In *Proceedings of the 16th International Conference on Human-Computer Interaction with Mobile Devices & Services (MobileHCI '14)*. ACM, 319–328. https://doi.org/10.1145/2628363.2628379

[48] Miguel A. Nacenta, Carl Gutwin, Dzmitry Aliakseyeu, and Sriram Subramanian. 2009. There and Back Again: Cross-Display Object Movement in Multi-Display Environments. *Human–Computer Interaction* 24, 1-2 (2009), 170–229. https://doi.org/10.1080/07370020902819882

[49] Miguel A. Nacenta, David Pinelle, Dane Stuckel, and Carl Gutwin. 2007. The Effects of Interaction Technique on Coordination in Tabletop Groupware. In *Proceedings of Graphics Interface 2007 (GI '07)*. ACM, 191–198. https://doi.org/10.1145/1268517.1268550

[50] 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, 831–840. https://doi.org/10.1145/2470654.2470773

[51] 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, 177–186. https://doi.org/10.1145/1978942.1978969

[52] Halla Olafsdottir, Vladimir M Zatsiorsky, and Mark L Latash. 2005. Is the thumb a fifth finger? A study of digit interaction during force production tasks. *Experimental brain research* 160, 2 (2005), 203–213. https://doi.org/10.1007/s00221-004-2004-0

[53] Gary Perelman, Marcos Serrano, Christophe Bortolaso, Celia Picard, Mustapha Derras, and Emmanuel Dubois. 2019. Combining Tablets with Smartphones for Data Analytics. In *IFIP Conference on Human-Computer Interaction (INTERACT '19)*. Springer, 439–460. https://doi.org/10.1007/978-3-030-29390-1_24

[54] Hanae Rateau, Yosra Rekik, Laurent Grisoni, and Joaquim Jorge. 2016. Talaria: Continuous Drag & Drop on a Wall Display. In *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces (ISS '16)*. ACM, 199–204. https://doi.org/10.1145/2992154.2992164

[55] 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, 31–39. https://doi.org/10.1145/263407.263505

[56] Alexander Schick, Florian van de Camp, Joris Ijsselmuiden, and Rainer Stiefelhagen. 2009. Extending Touch: Towards Interaction with Large-Scale Surfaces. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces (ITS '09)*. ACM, 117–124. https://doi.org/10.1145/1731903.1731927

[57] Dominik Schmidt, Fadi Chehimi, Enrico Rukzio, and Hans Gellersen. 2010. PhoneTouch: A Technique for Direct Phone Interaction on Surfaces. In *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology (UIST '10)*. ACM, 13–16. https://doi.org/10.1145/1866029.1866034

[58] Garth Shoemaker, Anthony Tang, and Kellogg S. Booth. 2007. Shadow Reaching: A New Perspective on Interaction for Large Displays. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology (UIST '07)*. ACM, 53–56. https://doi.org/10.1145/1294211.1294221

[59] Hemant Bhaskar Surale, Fabrice Matulic, and Daniel Vogel. 2019. Experimental Analysis of Barehand Mid-Air Mode-Switching Techniques in Virtual Reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, 1–14. https://doi.org/10.1145/3290605.3300426

[60] Theophanis Tsandilas, Anastasia Bezerianos, and Thibaut Jacob. 2015. SketchSliders: Sketching Widgets for Visual Exploration on Wall Displays. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, 3255–3264. https://doi.org/10.1145/2702123.2702129

[61] Philip Tuddenham, David Kirk, and Shahram Izadi. 2010. Graspables Revisited: Multi-Touch vs. Tangible Input for Tabletop Displays in Acquisition and Manipulation Tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, 2223–2232. https://doi.org/10.1145/1753326.1753662

[62] Simon Voelker, Kosuke Nakajima, Christian Thoresen, Yuichi Itoh, Kjell Ivar Øvergård, and Jan Borchers. 2013. PUCs: Detecting Transparent, Passive Untouched Capacitive Widgets on Unmodified Multi-Touch Displays. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces (ITS '13)*. ACM, 101–104. https://doi.org/10.1145/2512349.2512791

[63] Simon Voelker, Kjell Ivar Øvergård, Chat Wacharamanotham, and Jan Borchers. 2015. Knobology Revisited: A Comparison of User Performance between Tangible and Virtual Rotary Knobs. In *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces (ITS '15)*. ACM, 35–38. https://doi.org/10.1145/2817721.2817725

[64] Daniel Vogel and Ravin Balakrishnan. 2004. Interactive Public Ambient Displays: Transitioning from Implicit to Explicit, Public to Personal, Interaction with Multiple Users. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology (UIST '04)*. ACM, 137–146. https://doi.org/10.1145/1029632.1029656

[65] Daniel Vogel and Ravin Balakrishnan. 2005. Distant Freehand Pointing and Clicking on Very Large, High Resolution Displays. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology (UIST '05)*. ACM, 33–42. https://doi.org/10.1145/1095034.1095041

[66] Ulrich von Zadow, Daniel Bösel, Duc Dung Dam, Anke Lehmann, Patrick Reipschläger, and Raimund Dachselt. 2016. Miners: Communication and Awareness in Collaborative Gaming at an Interactive Display Wall. In *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces (ISS '16)*. ACM, 235–240. https://doi.org/10.1145/2992154.2992174

[67] Ulrich von Zadow, Wolfgang Büschel, Ricardo Langner, and Raimund Dachselt. 2014. SleeD: Using a Sleeve Display to Interact with Touch-Sensitive Display Walls. In *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces (ITS '14)*. ACM, 129–138. https://doi.org/10.1145/2669485.2669507

[68] Malte Weiss, Julie Wagner, Yvonne Jansen, Roger Jennings, Ramsin Khoshabeh, James D. Hollan, and Jan Borchers. 2009. SLAP Widgets: Bridging the Gap between Virtual and Physical Controls on Tabletops. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, 481–490. https://doi.org/10.1145/1518701.1518779

[69] Pierre Wellner. 1991. The DigitalDesk Calculator: Tangible Manipulation on a Desk Top Display. In *Proceedings of the 4th Annual ACM Symposium on User Interface Software and Technology (UIST '91)*. ACM, 27–33. https://doi.org/10.1145/120782.120785

[70] Andrew D. Wilson and Hrvoje Benko. 2010. Combining Multiple Depth Cameras and Projectors for Interactions on, above and between Surfaces. In *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology (UIST '10)*. ACM, 273–282. https://doi.org/10.1145/1866029.1866073