

Understanding Finger Input Above Desktop Devices

Chat Wacharamanatham Kashyap Todi Marty Pye Jan Borchers

RWTH Aachen University
52062 Aachen, Germany

{chat, borchers}@cs.rwth-aachen.de, {kashyap.todi, marty.pye}@rwth-aachen.de

ABSTRACT

Using the space above desktop input devices adds a rich new input channel to desktop interaction. Input in this elevated layer has been previously used to modify the granularity of a 2D slider, navigate layers of a 3D body scan above a multitouch table and access vertically stacked menus. However, designing these interactions is challenging because the lack of haptic and direct visual feedback easily leads to input errors. For bare finger input, the user's fingers need to reliably enter and stay inside the interactive layer, and engagement techniques such as midair clicking have to be disambiguated from leaving the layer. These issues have been addressed for interactions in which users operate other devices in midair, but there is little guidance for the design of bare finger input in this space.

In this paper, we present the results of two user studies that inform the design of finger input above desktop devices. Our studies show that 2 cm is the minimum thickness of the above-surface volume that users can reliably remain within. We found that when accessing midair layers, users do not automatically move to the same height. To address this, we introduce a technique that dynamically determines the height at which the layer is placed, depending on the velocity profile of the user's initial finger movement into midair. Finally, we propose a technique that reliably distinguishes clicking from homing movements, based on the user's hand shape. We structure the presentation of our findings using Buxton's three-state input model, adding additional states and transitions for above-surface interactions.

Author Keywords

Finger input; midair; near-surface; height; thickness

ACM Classification Keywords

H.5.2. Information Interfaces and Presentation (e.g. HCI): User Interfaces

INTRODUCTION

Sensing midair input in horizontal near-surface space has been used to augment keyboards, mice, and touchscreens. The horizontal near-surface space provides better ergonomics

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CHI 2014, April 26–May 1, 2014, Toronto, Ontario, Canada.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2473-1/14/04...\$15.00.
<http://dx.doi.org/10.1145/2556288.2557151>

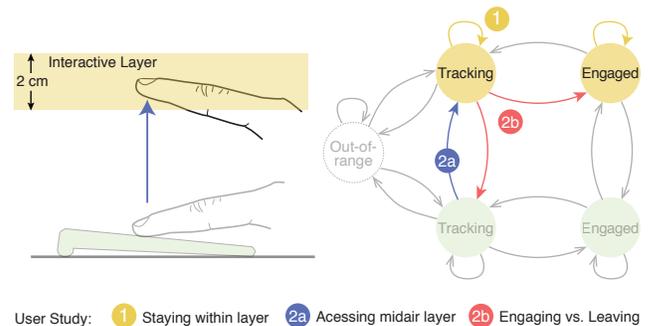


Figure 1: Interacting within an above-surface layer. We applied Buxton's three state model to explain the relevance of the user studies for such interactions.

than vertical midair space because users can rest their arms or elbows on the desk surface. Additionally, users can quickly switch to existing desktop devices for already-efficient tasks, e.g., using the keyboard for typing. To facilitate this switching between midair and desk-based devices, we focus on midair near-surface input with bare fingers.

Designing interactions for the near-surface space is difficult because the lack of haptic and direct visual feedback leads to input errors. Physiological tremor can cause the finger height to be unstable. Previous work partitioned the space into multiple interaction layers and determined appropriate layer thickness for using a stylus in midair. However, the movement behaviour of free hands differs from that of holding a stylus.

Another problem of midair input is distinguishing the intention of finger use. A finger in the near-surface volume may be intended for different kinds of interactions, such as tracking a cursor or activating an object, akin to clicking with the mouse. Alternatively, the presence of the finger in the volume may be unintentional while homing towards physical devices on the desk. We show that these states can be reliably classified, and brought into compliance with Buxton's three-state reference model for input [2].

This paper thus makes the following contributions:

- We determine an appropriate thickness for near-surface interaction layers through an empirical study.
- Based on an analysis of users' finger velocity during near-surface input, we propose a method to dynamically place the layer, which prevents over- and undershoots upon entering the volume.

- We present a near-surface “engagement” technique (clicking) that minimizes inadvertent layer changes and unintended engagements upon leaving the near-surface space.

In the following section, we describe the challenges of designing an interaction technique in the near-surface volume, using Buxton’s model.

INTERACTION MODEL

Buxton’s three-state model can be applied to near-surface interaction (Fig. 1). Ideally, an input technique would allow three states, “out of range”, “tracking” and “engaged”, and the transitions between them. As we will see in the following paragraphs, certain characteristics of the midair volume make this hard to achieve.

First of all, we need to disambiguate the midair volume above the desk surface from desk-bound devices. This means that users should be able to switch between desk-bound devices and midair at will (Fig. 1 transition 2a). Therefore, the volume needs to be placed at a certain *height* above the surface. This height is defined as the orthogonal distance from the desk surface to the bottom of the volume, and requires careful consideration. A volume that is positioned too high leads to more exertion and makes the interaction more overt, whereas a volume that is placed too low can lead to ambiguous interaction with desk-bound devices. One method to avoid ambiguous interaction would be to use a toggle button or a quasimode, where the user holds down a designated key, thus “enabling” the midair space for interaction. However, these switches add an additional step, which can introduce mode errors. A press-and-hold mode would forbid bimanual input. Having a static starting height for the volume could cause the user to not move high enough (undershoot) when homing into the volume. In order to optimize for all these factors, we propose a dynamic solution for accessing the volume, derived from the movement behaviour of the human hand into midair space.

Once the user has reached the midair volume, he needs to be able to remain in this volume (Fig. 1 transition 1). This means that the volume needs to have a minimum *thickness*. Thickness is defined as the distance from the bottom of the layer to the top of the layer, and also needs careful consideration. On the one hand, a thin volume is more susceptible to drifting, which can cause erroneous, unintended input. The user is required to carefully stay within the layer, which slows down their interaction and may lead to frustration. On the other hand, a thick volume, increases the distance the user has to move the hand when clutching out of the interaction area. This leads to slower input, fatigue and again, potential user frustration. Applications that use multiple layers for different semantics of interaction will have fewer possible layers, as the range of the human hand is limited. Increasing the thickness of the total volume requires tracking hardware that covers a larger area, and at the same time is more prone to catch unintended hand movement as an input signal. If we knew the minimum thickness in which a user can reliably remain without visual feedback, we would not be dependent on that visual feedback, which can clutter the screen and increase cognitive load. Therefore, we conducted a study in order to determine

the smallest possible layer thickness that still allows the user to reliably interact with it.

Assuming access to the midair-layer has been achieved, and the user is able to stay comfortably within it, what remains is an *engagement method* that allows entering the engaged state, similar to depressing a mouse button (Fig. 1 transition 2b). Such an engagement method should be an explicit action to reduce the likelihood of accidental engagement. The engagement method should allow each hand to be in the engaged state separately, enabling bimanual input if necessary. It should occupy a minimal amount of fingers, so that ideally, multiple fingers could maintain the engaged state separately, comparable to multitouch interaction. Therefore, we study the characteristics of the human hand’s movement in midair space and derive a suitable engagement technique.

In the next section, we review the literature of midair interactions in order to draw lessons relevant to the design considerations of height, thickness and engagement method.

RELATED WORK

The third dimension, added through the near-surface space, has previously been exploited for input techniques. Marquardt et al. investigated continuing touch input when the hand is raised above the surface to reduce content occlusion [10]. Hilliges et al. tracked the shadow of the hands when hovering over the surface of a tabletop to use it as a proxy to interact with objects [5]. The height of the hand can also be used as a parameter for a transfer function which can be additionally applied to the 2D input, e.g., modifying the granularity of a slider on a touchscreen [10]. Yu et al. used midair layers parallel to a trackpad to select different C/D-ratios [22]. Near-surface space can provide an additional layer of information associated with the two dimensional input. This additional layer has, for example, been used for a stylus over the tabletop [17], or for magic lenses [16]. Subramanian visualized the information on the existing device (tabletop), whereas Spindler visualized it directly in the midair layer using a projection on the magic lens. The additional dimension above the keyboard can also be used purely for control, decoupled from any horizontal movement. For example, Benko visualized the output directly on the hand, so that the user could navigate through a stack of menus vertically [20].

Previous work has discovered the need for a suitable thickness when placing multiple layers in near-surface space, and recommendations exist. Spindler et al. studied how accurately people are able to maintain a paper lens in a specific layer with two hands while hovering, and while performing a horizontal search task [15]. They found that the minimum layer thickness for hovering is 1 cm, while the minimum thickness for horizontal movement is about 4 cm. Subramanian investigated different layer thicknesses for stylus interaction with arm support on the desk in an informal study and recommended 4 cm [17]. A follow-up with a formal steering study in 1D, 2D and 3D showed that 2 cm sufficed to minimize movement time [8]. However, only movement along one axis was investigated, and continuous height feedback of the stylus within the layer was provided.

Although free finger interaction seems to require a similar layer setup as previous work, the finger movement differs from the aforementioned techniques in some crucial aspects, therefore jeopardizing the applicability of their guidelines. While a minimum 4 cm thickness may apply for the bimanual usage of magic lenses while standing at tablespots [15], this thickness may not apply in a seated position, where only one hand is used. Although it is plausible that the thickness acquired by Kattinakere et al. [8] would apply to free finger movement, this assumption cannot be made, as a finger moves along a different trajectory than a stylus. The movement path of a finger is more curved than that of a stylus, making the path longer, and therefore more prone to drifts [4].

Additionally to the inherent differences of free finger interaction, several other factors differ from previous work. While Spindler’s setup increases the stability of the object by spreading the load over two hands [16], the stability is potentially reduced by the user having to hold the object in midair without allowing arms to be rested. The continuous visual feedback provided in Kattinakere’s study allows closed-loop adjustment of the height, potentially resulting in a more precise movement [8].

Another important factor of near-surface input is the engagement technique, analogous to clicking. Existing engagement techniques that have been proposed for midair input allow at most one input state per hand, and can be classified as follows. First, a hardware button can be used. This can be either on a separate device, e.g., Mysliwiec used one hand to point, the other to press a clicking-key [12], or be on the pointing device, e.g., Subramanian used a button on the stylus for clicking [17]. Second, the movement can be used to change state, like crossing in and out of targets (horizontal movement), or crossing midair layers (vertical movement) [17]. Finally, hand-shape gestures can be used to decouple the movement from the engagement. Wilson used a pinching gesture of index finger and thumb to click [19]. In order to improve the stability of the tracked cursor, Kato and Yanagihara tracked the knuckle position instead of the fingertip, making the pinching engagement independent from the movement [7]. Vogel used the striking of the thumb on the index finger to detect a click [18]. This was improved by Banerjee et al. to striking the middle finger instead of index finger to enhance stability of the cursor movement [1]. Perhaps the most intuitive of all engagement techniques is to emulate tapping by using the same finger motion in the air [18]. This was ranked best by users because of the familiarity with mouse clicking [3]. Pyryeskin et al. performed a gesture elicitation for above multitouch-surface selection. Air-tapping (“push with a finger”) was the second most frequent gesture (26.6%), slightly less frequent than grabbing (35.2%), which occupies the whole hand [14]. Our paper takes a closer look at this index finger tapping, and addresses the shift of the finger that can cause erroneous input.

Before tackling the challenges of layer access and engagement, we will determine the required layer thickness for maintaining the tracking state in the next section.

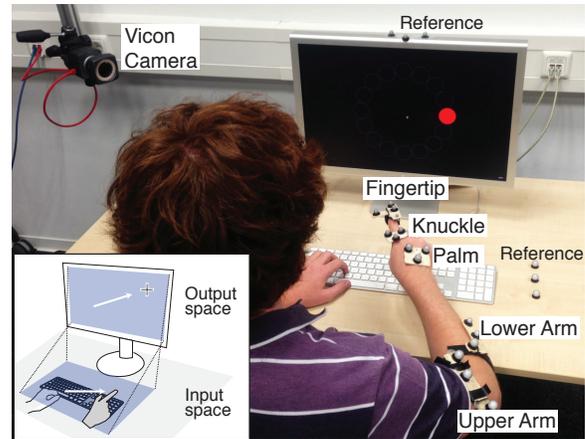


Figure 2: The physical setup of the user studies. Reflective markers were attached to the participants to record data for various sections of the right arm. Indirect mapping was used to prevent effect of hand occlusion.

STUDY 1: THICKNESS OF ABOVE-SURFACE LAYERS

In layered mid-air interactions, having thin layers allows for a large number of such layers within a given interaction volume. However, we need to take into consideration the ergonomics of the human arm and hand. Factors like hand tremor and drifting, and lack of haptic feedback in the near-surface space, make it difficult for users to maintain their hands at a constant level. This makes thinner layers harder for users to stay inside. This study aims to determine a suitable thickness of such near-surface layers while the finger remains in the tracking state (Fig. 1 transition 1).

Apparatus

The position of the users’ index fingertip was used as input to the application, to control a screen cursor. To obtain accurate positional coordinates, we used a Vicon motion-capture system to track passive infrared-reflective markers, which provided three-dimensional data with sub-millimeter accuracy at 100 Hz. Markers were attached to the user’s finger with lightweight patches (<8g each). Before the test, users flexed their fingers to ensure that the patches did not inhibit their movements. In addition, we recorded the position of the user’s index fingertip, wrist and elbow for analysis purposes (Fig. 2). The position of knuckle and palm was only recorded in the second study. The experiment tasks were displayed on an Apple Cinema Display (49.5 cm × 30.5 cm; 1920 × 1200 pixels). To prevent the influence of hand occlusion, the user’s finger was mapped from the orthogonal projection onto the desk surface, to the vertical screen. The participants sat in front of a desk, approximately 50 cm away from the screen, on an adjustable chair with armrests. The height of the chair was adjusted by each user according to preference.

Participants

Eight right-handed volunteers (two female) of computer science background were recruited (mean age of 24). Participants did not exhibit any known hand disabilities (severe tremor, etc.). Any visual impairments were noted down prior

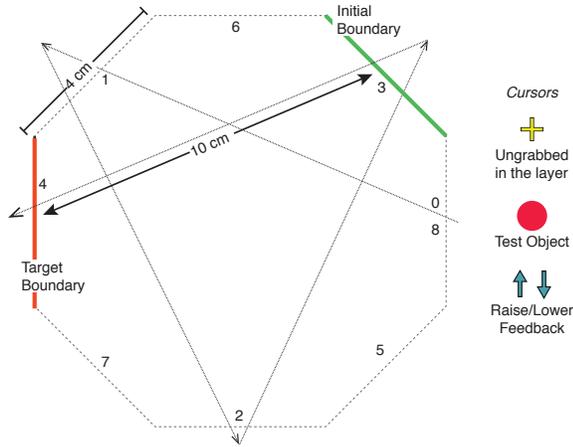


Figure 3: The steering tasks performed by the users. The cursor was used to grab a test object, and drag it from the initial to the target boundary. Different cursors provided feedback about the z-direction in which the finger needed to be moved to re-enter the layer.

to the test and it was confirmed that they had been corrected. All participants had a computer science background.

Task

The task users performed involved finger steering within a near-surface layer, similar to Kattinakere et al.'s [8]. Users controlled an on-screen cursor with their index finger to drag the object from an initial boundary across a target boundary (Fig. 3). After crossing each boundary, the next target boundary was shifted anti-clockwise. When the cursor enters the object, the object was automatically grabbed (no engaging gesture required). When the finger exited the layer, the object was released. Users could only pick the object back up again by readjusting their finger height.

Whenever the finger was outside the near-surface layer, the cursor changed to an arrow indicating whether the finger was too high or too low. Therefore, during the movement within the layer, the users needed to rely on proprioception to maintain the height of the finger.

To ensure constant initial velocity of hand movement, a short pause was programmed after crossing each goal. Since arm movements vary depending on the direction, the eight boundaries were evenly distributed, in order to make sure that the results were applicable to 2D movements in general.

Design

A within-subjects design was used for the user study. The independent variables used were the *Thickness*, *SurfaceSupport*, and *MovementDistance*. The different layer thickness values used were 1, 2, 3 and 4 cm. The two hand support configurations tested were (1) the users rested the wrist on the desk surface, and (2) the wrist had no support. Two types of movement sizes were used *SmallMovement* (1 cm) and *LargeMovement* (10 cm).

- *DriftCount*: Less drifts indicate that the user can reliably maintain the finger in the layer. This is the main measurement for determining optimal thickness. However, less drifts may result from the fact that the user carefully moves the finger during the test.
- *ManipulationTime*: Time taken for dragging the object from the initial boundary to the final boundary. This is measured only when the finger is within the given thickness and the object has been grabbed. This time excludes instances where the finger drifts outside the layer, causing the object to be released. Lower cursor manipulation time results from faster movement within the layer. This indicates that the user is more confident in moving the object in the layer.

In summary, the experimental design was: 8 participants \times 4 *Thickness* \times 2 *SurfaceSupport* \times 2 *MovementDistance* \times 8 boundaries \times 3 repetitions = 3072 total trials.

Data analysis

In order to make each row of the data represent the performance across all movement directions, we averaged the *ManipulationTime* across all movement directions in the same repetition. *DriftCount* was also summed across movement directions.

We used mixed-effect model analysis of variance (henceforth “ANOVA”) in which *Thickness*, *MovementDistance*, and *SurfaceSupport* are fixed effects and *UserID* is a random effect. We used Tukey HSD for post-hoc tests (henceforth “post-hoc test”). For all significant interaction effects, we performed the post-hoc tests for both main and interaction effects. The statistically significant results in the following section have the agreement of post-hoc tests up to the highest degree of significant interaction effects.

As expected, both *DriftCount* and *ManipulationTime* are not normally distributed. Appropriate transformations were applied before ANOVA and post-hoc tests. Neither Poisson nor Gamma-Poisson distribution fitted the *DriftCount*; therefore, it is aligned-rank transformed [21]. The *ManipulationTime* fits log-normal distributions, so we applied a $y = \log(x)$ transformation.

Due to the lack of normality, descriptive statistics (mean, median, and 95% confidence interval) derived directly from the data may not be an accurate representation, e.g., means of *ManipulationTime* are pulled higher by a long tail of the log-normal distribution. The resolution of the statistics could be too low, e.g., the median of many *DriftCount* conditions are 1 regardless of their distributions. Also, simple inverse-transforms of CIs may produce the interval that excludes the sample mean. Therefore, to provide useful estimates, we calculated the descriptive statistics by ordinary non-parametric bootstrapping (10,000 replicates). CIs were calculated with the bias-corrected and accelerated method (BCa).

We used the original scales for the charts and the descriptive statistics. Error bars denote 95% confidence intervals.

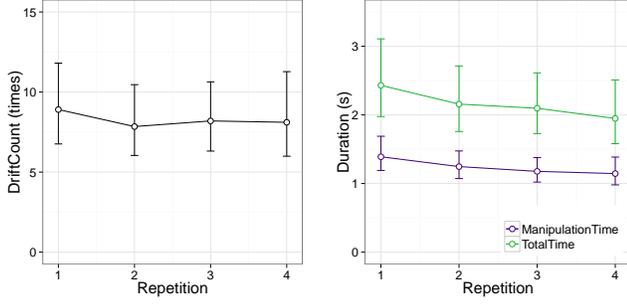


Figure 4: User’s performance in the first repetition differs from the others, indicating the learning effect. The learning curve stabilizes after the first repetition.

Table 1: The significant main- and interaction effects to the both *DriftCount* and *ManipulationTime*.

	df	F	p	
DriftCount				
Thickness	3	329	79.534	<.0001
Height	1	329	171.09	<.0001
MovementSize	1	329	162.01	<.0001
Thickness * Height	3	329	60.57	<.0001
MovementSize * Thickness	3	329	53.87	<.0001
MovementSize * Height	1	329	110.95	<.0001
MovementSize * Thickness * Height	3	329	28.88	<.0001
ManipulationTime				
Thickness	3	329	41.68	<.0001
Height	1	329	330.38	<.0001
MovementSize	1	329	215.10	<.0001
Thickness * Height	3	329	11.35	<.0001
MovementSize * Thickness	3	329	20.59	<.0001
MovementSize * Height	1	329	29.10	<.0001

Results and Discussion

Learning Effect

As expected, we found a statistically significant learning effect of repetitions on *ManipulationTime* ($F_{3,441} = 4.91, p = .0023$). The post-hoc test indicated that only the first repetition stood out from the rest (16% slower in *ManipulationTime*, Fig. 4). The faster movements suggest that the users were rapidly gaining confidence after one repetition whereas the drifts occurred independently of the users’ confidence ($F_{3,441} = 1.12, p = .3394$). The interaction effects between repetitions and other independent variables were not statistically significant. To rule out the influence of users’ confidence, we excluded the first repetition from the following analysis (resulting in 384 data rows).

Layer Thickness Analysis

The ANOVA results are shown in Table 1. As expected, we found significant main effects of *Thickness*, *SurfaceSupport*, and *MovementDistance* on both dependent variables. Due to significant interaction effects, we will take a closer look at each condition to determine suitable thickness.

Movement with surface support

When the hand was supported by the desk surface, post-hoc tests found no statistically significant differences in *DriftCount* and *ManipulationTime* among the 2, 3, and 4 cm thick-

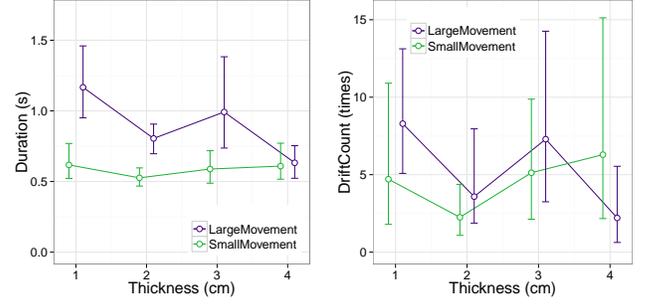


Figure 5: Drift occurrences and manipulation time for hand-on-desk conditions. Post-hoc tests shows that the 2 cm thickness yields similar performance to thicker layers.

nesses. The users drifted more in the 1 cm layer ($M = 6.50$ times) than in the others (4.46 times). This difference was independent of the movement sizes (Fig. 5). The users also moved 29% slower in the 1 cm layer than the others. Post-hoc comparison in the *SmallMovement* condition, which is the easiest condition, highlighted the worsened performance of the 1 cm compared to the 2 cm layer. When users achieved comparable *DriftCount*, the post-hoc test indicated that the *ManipulationTime* is significantly slower. In summary, when surface support is available, the near-surface layer can be as thin as 2 cm without worsening the user’s performance.

In contrast to [8], our visual feedback only indicates whether the finger is in the layer, too high, or too low. The similar result of the 2 cm thickness suggests that maintaining a finger in these conditions may not require continuous visual feedback of the finger height with respect to the layer.

Movement without surface support

When the hand and arm are not supported by the desk surface, the effect of *Thickness* was different between *SmallMovement* and *LargeMovement*.

SmallMovement: Post-hoc tests found no statistically significant differences in *DriftCount* and *ManipulationTime* among the 2, 3, and 4 cm thicknesses. Fig. 6 shows that drifts in the 1 cm thickness ($M_{1cm} = 10.79$ times) triples those of the others ($M_{2,3,4cm} = 3.25$ times). Longer *ManipulationTime* in the 1cm layer ($M_{1cm} = 1.17$ s vs. $M_{2,3,4cm} = 0.95$ s) suggests that users moved with significantly less confidence for this thickness.

Although the 2 cm result is similar to the hand-on-surface condition, the *SmallMovement* in midair is only comparable to the *LargeMovement* with surface support ($M_{DriftCount} = 4.33$ times and $M_{ManipulationTime} = 0.81$ s). *SmallMovement* with surface support in the 2 cm thickness is slightly better than both ($M_{DriftCount} = 2.25$ and $M_{ManipulationTime} = 0.53$).

LargeMovement: Post-hoc tests found no statistically significant difference in *DriftCount* between the 3 cm and 4 cm thickness ($M_{3cm} = 9.58$ times, $M_{4cm} = 4.15$ times, Fig. 7). However, users were faster in the 4 cm ($M_{3cm} = 1.91$ s, $M_{4cm} = 1.10$ s). The 4 cm thickness in midair also performed simi-

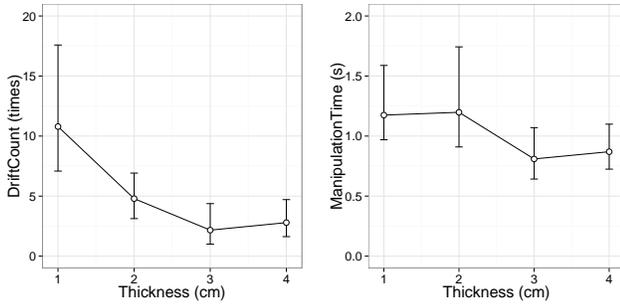


Figure 6: Drift count and manipulation time for *SmallMovement* in midair. The 2 cm thickness yields similar performance to thicker layers, similarly to the hand-on-surface condition.

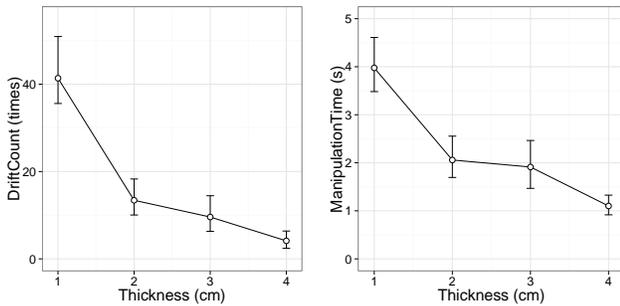


Figure 7: Drift occurrences and manipulation time for *LargeMovement* in midair. Although users drift by comparable amounts in the 3 cm and 4 cm layers, they were significantly faster in the 4 cm layer.

larly to the 2, 3, and 4 cm thickness on the surface ($M_{DriftCount} = 4.45$ times, $M_{ManipulationTime} = 0.69$ s). Therefore, it is possible to compensate the lack of surface support by increasing the thickness to 4 cm.

Will layers thicker than 4 cm be better? We repeated the study for 4 cm, 5 cm, 6 cm, and 7 cm thickness for large movements in midair. The test was done with a different group of participants (average age 25.84 years, 2 female). The increased thickness did not significantly improve *DriftCount*, $F_{3,77} = 0.27, p = .76$. However, the thicker layer allowed users to move more freely, therefore, significantly reducing the *ManipulationTime*, $F_{3,77} = 6.97, p = .0003$. However, each of the 1 cm thicknesses reduced the *ManipulationTime* by only 0.1s. This improvement is much smaller than the 0.8 s improvement from 3 cm to 4 cm.

In summary, for finger movement in the near-surface space without the arm touching the surface, we recommend 2 cm thickness for short movements (1 cm) and 4 cm thickness for long movements (10 cm).

STUDY 2: ELICITING USER'S MOTION DURING ACCESSING, ENGAGING, AND LEAVING THE MIDAIR LAYER

This elicitation study aims to determine how the user's finger moves when homing into a midair layer, and how the characteristics differ from those of moving the cursor. Also, we wish to investigate how the hand parts behave during engagement (clicking and dragging). Some characteristics of the motion may be consistent enough for real-time detection of engagement gestures.

Apparatus

We refer to the Apparatus section of the first study. In addition to the positions of fingertip, wrist and elbow, we recorded the positions of the major knuckle and the opisthenar.

Participants

Ten unpaid right-handed participants (three female), without any known hand disabilities (severe tremor, etc.), with a mean age of 25 were recruited for the study. Any visual impairments were noted down prior to the test and it was confirmed that they had been corrected. All but two of the participants had a computer science background.

Task

Users performed a two-dimensional point-and-write task, similar to Ortega and Nigay for AirMouse [13]. They alternated between typing and pointing, controlling an on-screen cursor with their right index finger. A trial consisted of first typing some text, and then pressing enter with the right hand. After that, the user had to raise their hand into midair and point to the target, which they then clicked by tapping their index finger down and back up again, thus clicking the target. They then had to drag the target by pressing the index finger down and moving it to the next goal, while keeping the index finger down. Once the goal had been reached, they had to release the target by bringing their index finger back up into a normal pointing state. Once the target had been released, a new text field appeared, and the user homed back down to the keyboard in order to start typing again, and repeat the cycle.

No engagement or midair-access recognition algorithm was used in this study, so as not to taint the user's data by a specific algorithm. Instead, a Wizard-of-Oz technique was used to trigger the state of the application based on the user's finger motion. The user's finger was mapped with a 1:1 absolute mapping.

Pilot studies showed that the users' performance in term of overall duration was stable during the second to fourth repetition. Therefore, the test was conducted with 4 repetitions, with the first repetition marked as a training run. In summary, this resulted in 10 participants \times 16 tasks \times 4 repetitions = 640 total trials.

Results and Discussion

The results and discussion are divided into two parts. We begin by discussing the layer height, followed by a discussion about distinguishing between homing and engagement.

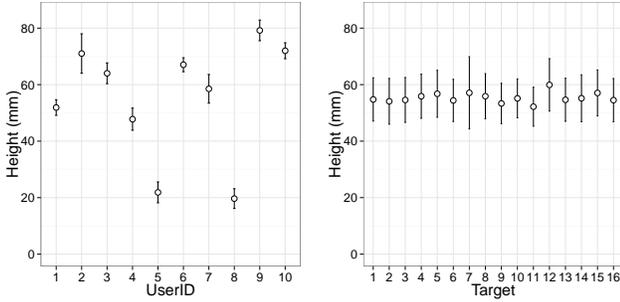


Figure 8: The initial homing stroke height greatly varies across users, whereas it is consistent across target locations. (User 5, 9, and 10 were female.)

Layer Height

After having removed 13 erroneous trials from the data, a mixed-model analysis of variance with the user as a random effect resulted in no significant effect of repetition and target number. This implies that no learning effect was to be observed within the four repetitions. A three-way analysis of variance showed a significant effect of the user ($F_{9,617} = 102, p < .0001$). Levene’s test indicated unequal variances ($F_{9,617} = 1.93, p = .0452$) across users (Fig. 8).

The substantial variation among users suggests that an initial static height would require practice, as different users prefer different heights. There are two possible solutions to deal with this user preference. The initial layer height could be either personalized or dynamically calculated for every homing stroke. The effectiveness of the personalization method would depend on whether individual users stably home to the same height, with little deviation. The effectiveness of the dynamic method would require some characteristic measure, which could be used to reliably calculate the height every time. Next, we analyze the data for hints whether personalization or dynamic calculation is more suitable.

In order to judge whether personalization of the layer height would be a suitable technique, we analyzed the average standard deviation of individual users, and found it to be 1.5 cm. Even with the highly conservative estimation that users will reliably home the finger with a maximal deviation of 1.5 cm, the fluctuation exceeds more than half of the 2 cm layer thickness learned from the first study. Therefore, having a personalized static height is unlikely to result in a satisfactory user experience. Instead, determining the initial height dynamically for every homing stroke seems to be more suitable.

It remains to be analyzed which parameter is a suitable measure to determine the initial layer height. A strong correlation of the overall velocity to both the horizontal (Pearson’s $r(627) = .93, p < .0001$) and vertical peak velocity ($r(627) = .74, p < .0001$) indicates that users move in both the horizontal and vertical direction while raising the finger. Therefore, using the motion vector alone would not suffice to detect the moving-to-midair gesture. However, a non-significant correlation between the horizontal and vertical valley velocity ($r(627) = .06, p = .13$) suggests that once the user stops rais-

Table 2: The significant main effect of participants, suggesting the potential to personalize the initial height.

	df	F	p
Vz (Peak-Valley)			
UserID	9	9	2.27 .0167
RepetitionNo	3	554.2	0.45 .7189
TargetNo	15	554.2	0.51 .9367
RepetitionNo * TargetNo	45	554.2	0.48 .9983
Vz (Peak)			
UserID	9	9	2.58 .0063
RepetitionNo	3	554.2	0.50 .6804
TargetNo	15	554.2	0.49 .9472
RepetitionNo * TargetNo	45	554.2	0.50 .9975
Vz (Valley)			
UserID	9	9	10.03 <.0001
RepetitionNo	3	554.2	2.28 .0785
TargetNo	15	554.2	1.05 .3986
RepetitionNo * TargetNo	45	554.2	0.98 .5030

ing his finger, the movement in horizontal direction does not necessarily stop. Therefore, the vertical finger velocity is a good candidate for detecting lifting-to-midair.

Further mixed-model ANOVA of the vertical peak velocity, valley velocity and the difference between both revealed no significant main effect of repetition and target number. Three-way ANOVA revealed a significant main effect of the user for all variables (Table 2). This suggests the possibility of personalizing the velocity threshold for transition to the layer by detecting either the peak velocity or velocity difference. Analysis showed radical difference between the magnitude of the valley velocity ($M = 5.13$ cm/s, $SD = 1.79$ cm/s) and the peak velocity ($M = 138.37$ cm/s, $SD = 41$ cm/s). Therefore, it is possible to use a cut-off by velocity method to detect the transition to the midair layer. Based on the empirical rule, a threshold of 21.25 cm/s is suitable.

Distinguishing Engagement Gesture from Leaving Midair

We manually filtered erroneous movement (e.g., starting to drag without clicking) in the first test repetition for each user based on the 3D coordinate of the fingertip. In total, we have reliable data from 157 clicks, 153 Mouse-downs, and 148 Mouse-ups. For this analysis, the data is classified into the following movement phases:

- **Pre-depressing:** The movement between the click and the height (during the cursor wait). We used all points in this phase before the user initialized the high velocity to depress.
- **DragDepressing:** The high velocity downward stroke followed by a large movement amplitude along the planar direction. All the points within this movement phase with the overall velocity greater than above the 97.5% quantile (200 cm/s) are used.
- **ClickDepressing:** The high velocity downward stroke followed by the upward stroke during the phase that the user is assigned to click in midair. We used the same velocity criteria as in *DragDepressing*.

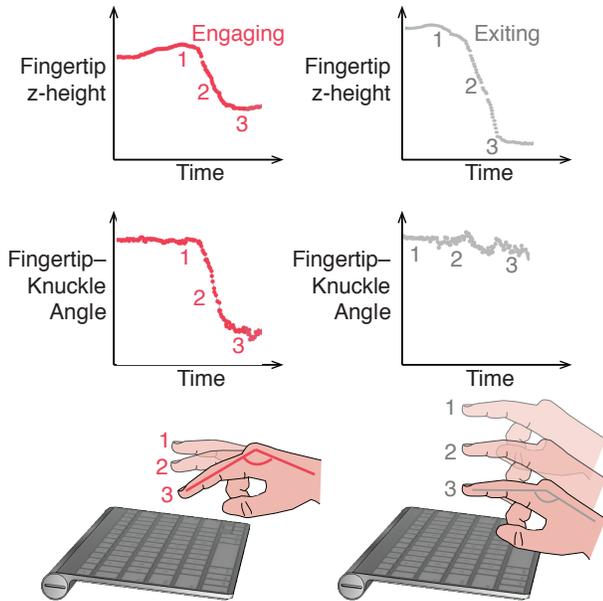


Figure 9: During engagement gesture (left), users flex their finger, causing the knuckle angle to change over time. During exiting movements (right), users did not change their knuckle angle. This difference in the movement profile can be used to identify the user’s intention.

- *Pre-leaving*: This is the point after the user lifts the finger from the dragging movement. It is manually annotated during the study.
- *Leaving*: The stroke after Pre-leaving point, but before the user reaches the keyboard. Again, we use all data points with velocity greater than 200 cm/s.

We considered positional and velocity features of the fingertip, knuckle and hand in order to help distinguish the engagement gesture by tapping from the leaving action, e.g., homing the hand back to the keyboard for typing.

As expected, we found that knuckle angles (Fig. 9) are significantly different between *Pre-depressing* and *DragDepressing* ($t_{87} = 5.87, p < .0001$) along the movement direction. However, the differences of the knuckle angle in *Pre-leaving* and *Leaving* are not statistically significant ($t_{157} = -.1632, p = .1047$). This suggests that the change in the knuckle angle could be used to distinguish the engagement from the leaving gesture. Additionally, the difference of the knuckle angles between *ClickDepressing* and *DragDepressing* is also not statistically significant, ($t_{87} = 1.88, p = .0639$). This indicates that the knuckle angle can be used reliably, regardless of the duration that the finger dwells in the engaged state.

These results indicate that the angle between the fingertip, knuckle and palm do not significantly change when the users leaves midair space to the desktop devices. This feature allows us to distinguish the engagement from the leaving midair action.

Limitation

Our study used a keyboard as a representation of desk-based devices. Prehension research indicates that grip forms only after 75% of the movement [6]. While this suggests that our finding should generalize to device of different shapes, the influence of the affordance of each desktop device during leaving the midair space warrants further studies. Sensing the shape of the desktop devices may allow more accurate classification of the engagement gesture from the leaving midair action.

The small number of participants biased towards male may influence the results. However, we did not find statistically significant effect of gender or lower arm length. We surmised that the difference across users were greater (Fig. 8).

DYNAMIC ENTRY HEIGHT

Analysing these three different behaviour patterns of users, while switching from on-surface to near-surface interactions, has helped us in devising an algorithm which does the following:

1. *Intention detection*: We can estimate whether the user wishes to either simply start near-surface interactions, browse through the stack of mid-air layers, or if the movement is unintentional, and the user wishes to perform other activities.
2. *Layer Positioning*: In the case of the user wishing to start near-surface interactions, we can intelligently position the first interaction layer dynamically. In the scenario where users browse through a stack of layers, the layers are positioned statically.

The algorithm is formulated in Fig. 10. For the velocity threshold v_{min} , we recommend the result from Study 2 (21.25 cm/s). From iteratively tuning the parameter, we found that 0.1 s is a suitable minimum duration d_{min} . Figure 11 illustrates the stages of this algorithm.

We performed 10-fold cross-validation of the algorithm with the data from Study 2. The results show 95.16% accuracy with stable v_{min} thresholds among iterations ($M=19.41$ cm/s, $SD=3.46$).

Bimanual Hand Usage

While using dynamic layer positioning, special care needs to be taken while designing for bimanual interactions. The implementation of the technique can vary depending on the usage scenarios:

Symmetric Hand Movements: When the application involves bimanual interactions, where both hands need to be coordinated, the positioning of the layers should be the same for both hands. Hence, to determine the exact placement of the layer, using dynamic positioning, the point of entry of the first hand into the interaction volume is considered, and the layer position is fixed with reference to this point.

Asymmetric Hand Movements: When the application involves asymmetric bimanual interactions, each hand has its own reference frame. In such a case, it is possible to position the near-surface layer independently for each hand. Hence,

The **DynamicHeight** function is called to determine the height of the interactive layer while the finger is out of the interactive layer.

Constants:

h_1 : default height of the lowest layer.

t : layer thickness

v_{min} : minimum velocity for a fast intended stroke.

d_{min} : minimum duration in for a slow intentional stroke.

Variables:

C : current frame number

$Z[0, C]$: array of buffered z-coordinates

$V[0, C]$: array of buffered z-velocities

Z and V are cleared everytime the finger leaves the tracking range

Helpers:

FireEvent(i) fires above-surface interaction event using data at frame i

FindPeak(M) returns the index of the highest peak in array M

FindValley(M) returns the index of the lowest valley in array M

DynamicHeight(C, Z, V):

```

1. P = FindMax(V[0, C])
2. if (P == INVALID)
3.     return INVALID
4. if V[P] > vmin (Detects fast strokes)
5.     L = FindValley(V[P, C])
6.     if (L != INVALID)
7.         for j from P to C (Fire events retrospectively)
8.             FireEvent(j)
9.     return Z[P] - (t / 2) (Detects slow strokes)
10. else if C > dmin
11.     FireEvent(C)
12.     return h1
13. else
14.     return INVALID

```

Figure 10: Dynamic entry height algorithm

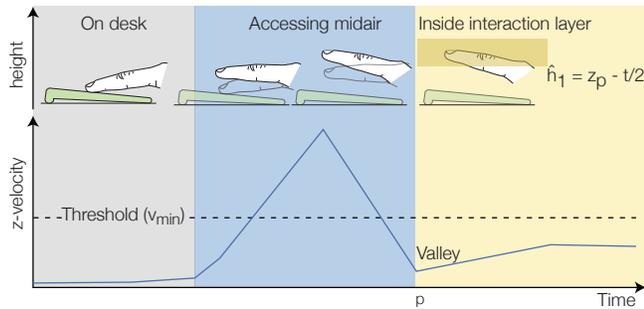


Figure 11: User's action while accessing a midair layer and the corresponding velocity profile. Dynamic height (\hat{h}_1) of the layer is set at the velocity valley, at time p .

the two hands can work in different interaction spaces, and this can avoid the occurrence of collisions.

IMPLICATIONS

The understanding of elementary operations presented in our paper (entering, staying, engaging, and leaving) is a basis for further above-surface interaction designs. We summarize these implications to the previously discussed interaction model.

Switching in and out of above-surface interaction without explicit switch or gestures: Using our dynamic height algorithm and leaving/engaging classification criteria could lead to more fluent switching between above-surface and desktop device input.

Enabling above-surface interaction without continuous height feedback: Using the recommended layer thicknesses reduces

the need of visual feedback for the relative position of the fingertip to the surface. This reduces accidental layer switching and reduces users' cognitive load in close-loop maintenance of finger height.

Natural engagement with air-tapping: Tracking knuckle angle allows air-tapping as above-surface engagement technique without being ambiguous to the leaving motion.

These implications could be applied to previous works.

Example 1: Hover Paint [14] lets users lift their finger above a tabletop to select brush size (z) and color (x, y). It could benefit from our work in 3 ways: (a) It uses a quick upward flick to activate the brush UI. However, such sudden movements can lead to fatigue. Our dynamic height algorithm would allow for a less tiring activation: the user can move their finger up more naturally, and our algorithm establishes the interaction layer dynamically and reliably based on this initial movement. (b) Hover Paint could also improve the stability of its brush size setting while picking a color by using a minimum layer thickness that does not trigger size changes. We provide the suitable thickness of this layer. (c) Finally, Hover Paint does not allow canceling a selection, and when users confirm their settings by flicking down, the Hover Paint video actually shows that they accidentally start painting. Air-tapping would have avoided this, and the knuckle-angle method we contribute could distinguish moving down (to cancel) from air-tapping (to confirm).

Example 2: SpaceTop [9] lets users interact with a virtual 3D world behind a transparent screen in midair above a keyboard. To avoid hitting virtual objects while typing, objects are placed into a layer whose height and thickness are fixed, and which is substantially high, preventing users from resting the hand on the surface. However, with extended use, this height leads to fatigue. We provide the findings to place SpaceTop's object layer closer to the surface, establish its height dynamically, and use the right layer thickness for interaction. Our knuckle-angle technique could also be applied to detect when users air-tap to activate vs. when they leave the midair area.

CONCLUSION AND FUTURE WORK

In our studies, we wished to determine the parameters necessary for enabling three-state input in the near-surface space. Our results show that 2 cm is the minimum thickness necessary for maintaining the tracking state. Furthermore, we found that users do not automatically use a consistent height for accessing the midair volume. Therefore, we designed a dynamic algorithm that reliably detects access to the midair volume, based on the empirical data gathered from our second user study. Motions of the human hand when engaging the midair layer are consistent enough to differentiate between engagement and homing to the desk device. We therefore proposed an engagement technique that reliably detects engagement of the midair layer, based on the empirical data from our second study.

Clearly, our approach has some limitations. The above results are applicable when the user interacts with the midair layer in the middle of the desk surface. Other placements

of the midair volume are possible, e.g., above the mouse to the right hand side of the user. The movement in this area uses different joint angles, therefore, the result we propose in this paper may have limited generalizability. For example, the thickness recommendations for maintaining the tracking state may not apply when the arm is almost fully outstretched to the far right.

As a next step, we will design and evaluate an interaction technique that uses bare finger input in near-surface space, containing our methods for midair access and engagement. Furthermore, we will investigate whether our techniques hold for multiple stacks of midair layers, and whether our techniques for layer engagement are not only able to differentiate between engagement and homing to the keyboard, but also between engagement and accessing higher and lower layers.

Further future work could involve investigating slanted layers, e.g., caused by a keyboard that is slanted for ergonomic reasons, and how they influence performance. However, the performance influence should be during layer access, which is already compensated by our dynamic height algorithm. The thickness should not be affected. Whether the keyboard frame is used as a frame of reference during orientation will be interesting to investigate. Also, there is research that uses a slanted touchscreen as a digital sketching board [11]. Investigating the effect of orientation will be useful to such applications.

ACKNOWLEDGMENTS

This work was funded in part by the German B-IT Foundation. We thank Thorsten Karrer for his valuable feedback, Philipp Wacker and Ignacio Avellino for their helps in the user studies.

REFERENCES

1. A. Banerjee, J. Burstyn, A. Girouard, and R. Vertegaal. Pointable: an in-air pointing technique to manipulate out-of-reach targets on tabletops. In *ITS '11*, 11–20. 2011.
2. W. Buxton. A three-state model of graphical input. In *INTERACT '90*, 449–456. 1990.
3. F. Camp, A. Schick, and R. Stiefelhagen. How to click in mid-air. In *Distributed, Ambient, and Pervasive Interactions*, volume 8028 of *LCNS*, 78–86. 2013.
4. M. Desmurget, M. Jordan, C. Prablanc, and M. Jeannerod. Constrained and unconstrained movements involve different control strategies. *J NEUROPHYSIOL*, 77(3):1644–1650, 1997.
5. O. Hilliges, S. Izadi, A. D. Wilson, S. Hodges, A. Garcia-Mendoza, and A. Butz. Interactions in the air: adding further depth to interactive tabletops. In *UIST '09*, 139–148. 2009.
6. M. Jeannerod. The timing of natural prehension movements. *J MOTOR BEHAV*, 16(3):235, 1984.
7. H. Kato and H. Yanagihara. Pacman ui: vision-based finger detection for positioning and clicking manipulations. In *MobileHCI '13*, 464–467. 2013.
8. R. S. Kattinakere, T. Grossman, and S. Subramanian. Modeling steering within above-the-surface interaction layers. In *CHI '07*, 317–326. 2007.
9. J. Lee, A. Olwal, H. Ishii, and C. Boulanger. SpaceTop: integrating 2D and spatial 3D interactions in a see-through desktop environment. In *CHI '13*, 189–192. 2013.
10. N. Marquardt, R. Jota, S. Greenberg, and J. Jorge. The continuous interaction space: Interaction techniques unifying touch and gesture on and above a digital surface. In *INTERACT '11*, 461–476. 2011.
11. C. Mueller-Tomfelde, A. Wessels, and C. Schremmer. Tilted tabletops: In between horizontal and vertical workspaces. In *Wksp on Horizontal Interactive Human Computer Systems, TABLETOP '08*, 49–56. 2008.
12. T. Mysliwiec. Fingermouse: A freehand computer pointing interface. In *Int'l Conference on Automatic Face and Gesture Recognition*, 372–377. 1994.
13. M. Ortega and L. Nigay. Airmouse: Finger gesture for 2d and 3d interaction. In *INTERACT '09*, 214–227. 2009.
14. D. Pyryeskin, M. Hancock, and J. Hoey. Comparing Elicited Gestures to Designer-created Gestures for Selection Above a Multitouch Surface. In *ITS '12*, 1–10. 2012.
15. M. Spindler, M. Martsch, and R. Dachsel. Going beyond the surface: studying multi-layer interaction above the tabletop. In *CHI '12*, 1277–1286. 2012.
16. M. Spindler, S. Stellmach, and R. Dachsel. Paperlens: advanced magic lens interaction above the tabletop. In *ITS '09*, 69–76. 2009.
17. S. Subramanian, D. Aliakseyeu, and A. Lucero. Multi-layer interaction for digital tables. In *UIST '06*, 269–272. 2006.
18. D. Vogel and R. Balakrishnan. Distant freehand pointing and clicking on very large, high resolution displays. In *UIST '05*, 33–42. 2005.
19. A. D. Wilson. Robust computer vision-based detection of pinching for one and two-handed gesture input. In *UIST '06*, 255–258. 2006.
20. A. D. Wilson and H. Benko. Combining multiple depth cameras and projectors for interactions on, above and between surfaces. In *UIST '10*, 273–282. 2010.
21. J. O. Wobbrock, L. Findlater, D. Gergle, and J. J. Higgins. The aligned rank transform for nonparametric factorial analyses using only anova procedures. In *CHI '11*, 143–146. 2011.
22. C. Yu, X. Tan, Y. Shi, and Y. Shi. Air finger: enabling multi-scale navigation by finger height above the surface. In *UbiComp '11*, 495–496. 2011.