RESEARCH-ARTICLE

AirPanes: Two-Handed Around-Device Interaction for Pane Switching on Smartphones

KHALAD HASAN, University of Manitoba, Winnipeg, MB, Canada

DAVID AHLSTRÖM, Alpen-Adria-Universität Klagenfurt, Klagenfurt, Carinthia, Austria

JUNHYEOK KIM, University of Manitoba, Winnipeg, MB, Canada

POURANG POLAD IRANI, University of Manitoba, Winnipeg, MB, Canada

Open Access Support provided by:

University of Manitoba

Alpen-Adria-Universität Klagenfurt

Published: 02 May 2017

Citation in BibTeX format

CHI '17: CHI Conference on Human Factors in Computing Systems
May 6 - 11, 2017
Colorado, Denver, USA

Conference Sponsors:
SIGCHI

AirPanes: Two-Handed Around-Device Interaction for Pane Switching on Smartphones

Khalad Hasan[1], David Ahlström[2], Junhyeok Kim[1], Pourang Irani[1]
[1]University of Manitoba
Winnipeg, Manitoba, Canada
{Khalad, kimj3415, Irani}@cs.umanitoba.ca
[2]Alpen-Adria-Universität Klagenfurt
Klagenfurt, Austria
david.ahlstroem@aau.at

ABSTRACT
In recent years, around device input has emerged as a complement to standard touch input, albeit in limited tasks and contexts, such as for item selection or map navigation. We push the boundaries for around device interactions to facilitate an entire smartphone application: browsing through large information lists to make a decision. To this end, we present AirPanes, a novel technique that allows two-handed in-air interactions, conjointly with touch input to perform analytic tasks, such as making a purchase decision. AirPanes resolves the inefficiencies of having to switch between multiple views or panes in common smartphone applications. We explore the design factors that make AirPanes efficient. In a controlled study, we find that AirPanes is on average 50% more efficient that standard touch input for an analytic task. We offer recommendations for implementing AirPanes in a broad range of applications.

Author Keywords
Around-Device Interaction; In-air input; Two-handed Mobile Interaction; Analytic Interfaces; m-commerce interfaces.

ACM Classification Keywords
H.5.2. Information interfaces and presentation (e.g., HCI): User Interfaces – Interaction styles.

INTRODUCTION
With the rapid advances in optical sensing and finger tracking technologies [9, 37], researchers have explored the use of the in-air space around a mobile device for input. Prior studies have demonstrated the use of in-air space for fundamental on-screen interactions, such as selecting on-screen [17] and off-screen items [18], switching modes [41], text-entry [29], zooming and panning [25]. While such prior work has laid the foundation for around-device in-air input, a complete mobile application that deploys and benefits from such an input modality has yet to be demonstrated. Accordingly, our goal is to step beyond the design and study of isolated in-air alternatives for standard on-screen interactions and to explore how in-air input can enhance user performance in more complete mobile scenarios. In particular, we focus on the ability to facilitate a complex (high-level) goal, such as making a decision through information exploration and interaction.

In the context of smartphone applications, GUI designers generally organize large information spaces into multiple views, or panes. Typically, each pane serves a specific function, such as providing interactive controls to query a large dataset, showing the query results, or showing details of a selected list item. This use of a multi-pane UI structure (analogous to Tabs or Windows in desktop applications) forces users to frequently switch back and forth between views, which quickly becomes tedious for even common tasks such as looking at details of items in a list.

We propose *AirPanes*, a novel strategy to structure a mobile interface, using panes located in mid-air around the device. As an example scenario we pick mobile commerce (also known as m-commerce) applications as these are used to purchase products through mobile devices by millions of users [35] and require browsing and interacting with large information content before arriving at a decision. We demonstrate the benefits of AirPanes in a scenario where the user browses products, applies filters, inspects result lists, and bookmarks interesting items before making a final purchase. Our AirPanes interface relies on two-handed in-air interaction and takes full advantage of the space around the device. It uses both the spatial region accessible by the thumb of the hand holding the device, as shown in Figure 1a, and the in-air space reachable by fingers of the non-holding hand, as shown in Figure 1b and c. Ways to leverage multiple in-

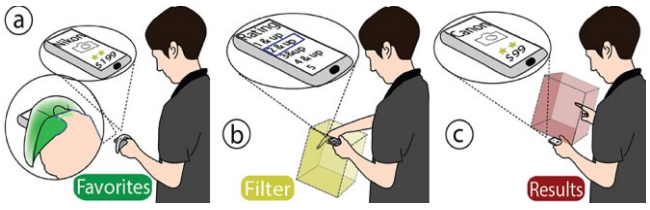

Figure 1. Two-handed around-device interaction in an m-commerce application using AirPanes. a) Previously tagged favorite products are access in-air with the thumb of the hand holding the phone. b) Query result lists and c) filter options are accessed using in-air panes reachable with the other hand.

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

DOI: http://dx.doi.org/10.1145/3025453.3026029

air spaces to provide access to functionality in mobile applications has never been explored before.

We first investigated how the mobile application interfaces are structured to support information exploration on such small screen. Our initial investigation indicates that application interfaces use multiple panes and users required to frequently switch between the panes to access necessary information. We use this knowledge to identify how best to organize panes in mid-air around the device. We optimize design parameters for AirPanes interactions necessary for a complete application, i.e., to select items, their details, to interact with filter controls, and to switch between views. Our results reveal promising item organization and selection strategies as well as layouts to facilitate in-air pane switching. We then designed and implemented AirPanes for an m-commerce app interfaces that works with a Vicon tracking system [37]. We evaluate AirPanes with standard touch interfaces and find that AirPanes is on average 50% more efficient than the common touch interaction.

Our findings came as a result of a systematic design process leading to the following contributions: 1) AirPanes, a strategy to structure mobile interfaces using content panes located in the air around the device that leverages two-handed around-device input; 2) two studies that explore and reveal insights on key AirPanes design factors; 3) a demonstration using AirPanes in an m-commerce interface to support an analytic decision-making task that requires browsing through large item sets and frequent switching between interface views.

RELATED WORK

We review previous work that inspired our AirPanes approach, including work on around-device interactions, in-air sensing techniques, and two-handed interactions.

Around-Device Interactions

Prior work on around-device interaction has shown that the in-air space surrounding a mobile device can be used as an alternative to touch input. Researchers have explored the use of around-device space for many fundamental operations, such as widget manipulation [17, 30], multi-touch interaction [4], digit entry [29], and item selection [17, 18]. These operations facilitate limited activities within a complete application, such as controlling the volume of a music player, switching between images, answering and rejecting phone calls, drawing shapes, or entering text. Research also shows that mid-air space can be used for more complex tasks, such as map navigation [39], content browsing [18, 30], mode switching, zooming [25, 46], and to manipulate 3D objects that are displayed on the screen [31].

Several earlier projects have focused on around-device sensing mechanisms and the associated factors, such as hardware, software, localization and classification of detected in-air movements [30, 41, 45], but without considering realistic use cases for around-device input and how to integrate this type of interactions in a complete application.

In this paper, we explore the use of around-device space for a complete smartphone application. We consider an advanced analytic task in our application, which requires users to perform many operations, such as item selection, item browsing, and switching between multiple interface views, or panes. Some of these operations are best suited for on-screen touch, but we identify which of them can be facilitated through in-air input.

People often operate their smartphone applications with one hand while their second hand serves other activities, such as holding a shopping bag while clinging onto a handle bar in a bus [24, 26, 27]. In such one-handed situations, the thumb belonging to the hand holding the device is the digit most readily available for input purposes [24]. However, given the bio-mechanical limits of the thumb, most users cannot reach all parts of the touchscreen with the thumb without readjusting the grip of the phone. Researchers have proposed different solutions to overcome this limitation, such as using a virtual on-screen thumb [32], fisheye and zoom techniques [28], pressure sensors along the side of the device [24], and tilt [8] or back-of-device input [21]. Although the thumb can easily be moved 'in-air' while holding the device, in-air thumb interaction has received very little attention. To our knowledge, our work is the first to investigate the use of this in-air space for a smartphone application. Additionally, our AirPanes strategy considers using two different in-air spaces: one controlled by the thumb holding the device, the second controlled via the index finger on the hand not holding the device. To the best of our knowledge, the combined use of two in-air spaces (and operating digits) has not been previously explored.

Around-Device Sensing Techniques

There has been substantial previous work from industry and academia investigating different sensing mechanisms to track movements around various devices. Several prior projects used industry solutions such as the Vicon [18, 19] and OptiTrack [15, 25, 42] systems, or the Microsoft Kinect [10, 22] to emulated an around-device sensing environment. Though such solutions provide precise motion capture data, miniaturizing to making them portable for mobile devices is not yet possible. Accordingly, numerous research projects have mounted external sensors to a mobile device to detect in-air hand movement in its vicinity. For example, IR sensors [4, 30, 45], depth cameras [9, 31, 44], and the use of GSM signals [47] have been promising solutions to detect in-air input. Also on-board sensors (magnetometers and cameras) in combination with external widgets, such as magnetic objects [17, 29] or an omnidirectional mirror [38, 46], have been shown to be useful for in-air tracking. Recently, Song et al. [41] showed that a smartphone's RGB camera can be used to recognize hand movement behind and in front of the camera without relying on external attachments. These

efforts inspire us to count on future smartphones with advanced 3D around-device sensing technologies.

Two-Handed Interactions

Previous work has demonstrated advantages of using two-handed input for symmetric and asymmetric bimanual tasks over one-handed input [3, 5, 7, 34]. Symmetric assignments, where both hands have a similar role and are used in parallel, have been shown to improved performance [3, 7, 34]. However, Balakrishnan and Hinckley [3] revealed that due to the lack of visual integration, symmetric role assignments could become asymmetric. In asymmetric bimanual interaction, as defined by Guiard's [13] Kinematic Chain model, the non-dominant hand provides the frame of reference and the dominant hand regulates the finer level of interaction details. AirPanes leverages asymmetric bimanual interaction: the non-dominant hand holding the device creates a reference point which the dominant hand navigates around. Additionally, the dominant hand performs fine-grained operations such as exploring items and panes, whereas the non-dominant hand (thumb) issues course-grained operations, such as making selections.

BACKGROUND ON MULTI-PANE INTERFACES

Our AirPanes strategy to structure a mobile interface in content panes located in mid-air around the device is motivated by the frequent (and often time consuming) pane switching imposed by the interface organization used in current mobile applications.

We started our exploration on the structure and organization used in mobile applications guided by Carrascal et al. [6] who found that people use their smartphones for the following most frequent purposes: 1) phone and audio, 2) SMS/texting, 3) social networking, 4) searching and browsing, and 5) email communication. We studied two applications in each purpose-category: 1) Phone dialer and Skype, 2) Messaging and Hangout, 3) Facebook and Twitter, 4) Google and Yahoo, and 5) Gmail and Yahoo mail. Our informal study revealed that all studied interfaces use multiple panes to organize functionality and information. The number of panes that is used depends on the features supported and the amount and type of information the application contains. However, generally the contained information is broken down into several smaller units which are presented and accessed on separate panes. A master pane (i.e., the home/start screen of the application) shows snippets, or links, to these panes in a vertical scrollable list (some applications use a 'horizontal' carousel widget). A tap on a snippet/link opens a new pane, which shows the full and detailed information. For example, the home screen of the Gmail app lists a snippet of each received email. The user accesses the content of the desired email by taping on its snippet. To check the next email, the user switches back to the home screen and taps the next snippet. Few apps provide a 'Next/Previous' button. Application functionality, settings and options are organized in the same way, with access through taps on snippets/links in an overview pane.

This common multi-pane structure often requires extensive scrolling in the master pane and forces the user to frequently switch back and forth between the different panes.

We iterated through several cycles of design-test-change activities to inform our AirPanes strategy for pane switching. In the following we first present the two most central user studies and then we present our final study where we evaluated the performance of the AirPane strategy against a touch-based interfaces.

TWO DESIGN STUDIES

In our work, we assume that robust finger tracking in 3D-space will be possible with future smartphones (as indicated by strong efforts in both industry, e.g., [33, 40, 43] and academia, e.g., [4, 17, 30]). We use a Vicon MX system to emulate a device with around-device tracking capabilities. The system tracks IR markers attached to a Google Nexus 5 phone (4.95-inch screen, 1080×1920 pixels) and to the user's left and right thumbs and right hand index finger. The four markers for each thumb and for the finger are positioned on 3D printed finger rings and the six markers for the phone are positioned on a 3D printed holder, which is attached to the top of the phone, as depicted in Figure 2a. With this arrangement of markers and with eight tracking cameras, the system provides noise- and dropout-free tracking of finger and device movements within a 3×3×2 meters large volume. Position data is sent from the Vicon system to an application (Unity+Android) on the smartphone, which interprets the data as user input and reacts with corresponding output.

People hold their smartphones and provide screen input in many ways, depending on the current task (e.g., texting vs. surfing), interaction (e.g., pinch-zooming vs. scrolling), and context (e.g., sitting vs. walking and carrying a bag). We focus on the common situation where the user holds the device in the non-preferred hand and uses the index finger on the preferred hand for input. In this constellation, the thumb of the non-preferred hand, which holds the device, is mostly passive and rests along the side of the device. However, our earlier work [20] demonstrates that an average user can easily and comfortably move the thumb of the hand that is holding the device at least 130mm sideways and 57mm up above the smartphone, as visualized in Figure 2b and c. Encouraged by this finding, we envision this accessible in-air thumb-space as being useful for triggering commands and for storing, selecting and browsing information items (much like Hasan et al.'s [18] AD-Binning concept).

Accordingly, we explored ways to arrange items (i.e., 'in-air buttons') within the accessible thumb-space and methods to select such in-air items. After pilot testing with different numbers of items and different ways to arrange items inside the thumb-space we found that a horizontal arrangement of 10 to 12 wedge-shaped items, in either two or three layers (Figure 2d and e), seems reasonable. Through pilots we also arrived at three promising selection methods: Dwell, Tilt, and Touch (Figure 2f, g, and h). To trigger a selection with Dwell, the user keeps the thumb still for 600ms within the

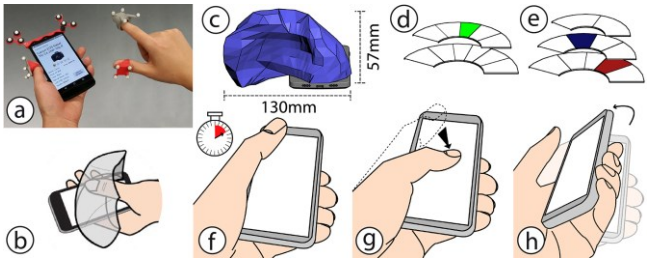

Figure 2. a) Tracking markers. b) In-air swipe motions. c) Spatial in-air region accessible by the thumb of the hand holding the device. d) On screen visualization with a colored cursor to indicate the position of the thumb inside the in-air region in relation to the items, arranged in two and e) three layers. f) Dwell selection. g) Touch selection. h) Tilt selection.

desired item's region. To select with Tilt, the user positions the thumb within the item's desired region and then quickly tilts (or rolls) the device sideways (we use a relative 30° threshold). The Touch method involves moving the thumb to the desired in-air region and then quickly tapping down with the thumb, anywhere, on the screen. We use a threshold to resolve situations where the thumb passes through undesired items on its way to the screen (e.g., when selecting an item in an upper layer): we ignore any items which the thumb visited for less than 250ms and treat the most recent item with a visit time greater than 250ms as the selected target.

Study 1: Selecting In-Air Regions with the Thumb

To inform our design choices for the AirPanes design, we studied users' performance with the three selection methods and the two different item arrangements in a 3×2 within-subjects experiment. Twelve right-handed smart-phone owners participated (3 female; no participant had previously participated in any pilot or study within our AirPanes project; mean age 25.2 years, s.d. 5.5).

Task

A start button in the middle and a 1.5×1.5cm visualization of the thumb-space (cf. Figure 2d and e) in the top-right corner of the screen are displayed at the trial start. A tap with the tracked left thumb on the button starts timing. The target item for the trial is colored red in the visualization and a blue position cursor indicates the current location of the thumb within the thumb-space. The cursor turns green when the thumb enters the target item (Figure 2d). Timing for the trial ends and the start screen for the next trial is loaded when the system detects a correctly issued selection from within the target item. An error is recorded (timing continues) if the participant performs a selection action from a non-target item.

All participants performed two series of six blocks of trials with each of the three selection methods, one series with the 2-Layer and one series with the 3-Layer arrangement. Each block contained one trial for each of the 11 item positions (presented in random order within a block). Participants were divided in six pairs, one pair for each of the six possible presentation orders of selection methods. One participant in each pair always started with the series for the 2-Layer arrangement, one started with the 3-Layer arrangement. Each participant performed a total of 396 timed trials: 3 methods × 2 arrangements × 6 blocks × 11 item positions. A study session lasted 45 min (incl. instructions, practice and breaks).

Results

Error trials: In 253 of the 4,752 trials (5.32%) participants issued one or more correct selection actions in a non-target item before correctly selecting the target item. In total, 5,115 correct selection actions were registered, 363 selections were on a non-target item: 99 with Touch, 130 with Dwell, and 134 with Tilt. A Friedman Test showed no significant difference between the three methods. The number of errors for each item position in the two arrangements are shown in Figure 3a and b. In total, 192 and 171 errors were registered with the 2-Layer and 171 3-Layer arrangements, respectively. A Wilcoxon Signed-Rank test showed no difference between the two arrangements.

Although the item size was slightly smaller in the middle of the layers (cf. Figure 2d and e), we see a trend with more erroneous selections for the slightly larger items at the right and left side. A possible reason for this is that the outer items require stretching the thumb more. Possible solutions to resolve this issue could be to either use a fisheye space discretization [18] that provides more space to the currently 'touched' item or to use a more extreme angle dependent discretization to assign even more space to outer items.

Trial time: We only use error free trials (4,499) to analyze selection time and take the median trial time for each method × arrangement × position combination for each participant. This median time data was right skewed and we performed a logarithmic transformation (which resulted in a distribution close to normal) before analyzing the data.

The geometric mean selection time (i.e., the antilog of the mean of the log-transformed data) was similar for the three methods: Touch 1.53s, Tilt 1.56s, and Dwell 1.60s. There was also no marked difference between the two arrangements: 2-Layer 1.58s, 3-Layer 1.56s. A two-way RM-ANOVA (method × arrangement) showed no significant main or interaction effect.

Figure 3c and d show the geometric mean selection time for each item position in the two arrangements. Two separate one-way ANOVAs, one for each arrangement, showed that there was a statistically significant effect for item position on selection time (2-Layer: $F_{10,121} = 9.31$, $p < 0.0001$, $\eta^2 = 0.44$;

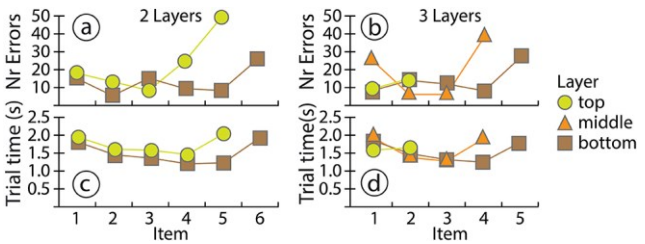

Figure 3. a) and b) Number of errors and c) and d) geometric mean trial time for each item position (from left to right in each layer, cf. Figure 3c and d).

3-Layer: $F_{10,121} = 6.67$, $p < 0.0001$, $\eta^2 = 0.36$). Again, as with error trials, we see a trend with more difficulties with outer items. Bonferroni adjusted post-hoc pairwise comparisons confirmed: the inner items, Item 3, 4, and 5, in the bottom layer in the 2-Layer arrangement were faster to select than the two outmost items in either layer; Item 4 in the bottom layer in the 3-Layer arrangement was faster to select than the two outmost items in either layer (all p's < 0.0009).

Summary: We conclude that regarding both errors and selection time, the three selection methods and the two arrangements score about equally well. The somewhat more error and slightly longer selection times for item positions at the outer ends in a layer indicate a trend suggesting that item positions at the far ends tend to be slightly more troublesome to select than items in the middle. The results regarding the three selection methods are promising and suggest that all of our three methods are suitable for in-air thumb selections, which gives the designer the flexibility to choose method as appropriate according to other factors, such as usage context or application type.

With insights about viable item arrangements and selection methods for the in-air thumb space, we move on and present a second central study we performed to inform our AirPanes apprroach.

Study 2: AirPane Layout and Pane Switching

As we noted in our survey on mobile application interfaces, the small display sizes used on smartphones forces designers to structure their interfaces in a multiple-pane. With such a structure the user has to repeatedly tap on small UI buttons (or on other interface elements, such as entries in scrollable lists) to switch views. Our AirPane approach is based on the idea to off-load panes into around-device space. With panes residing in-air, the user can quickly switch between panes by simply moving the in-air input finger inside the desired pane's in-air area – and the pane's content is displayed on the screen. We also envision that the user interacts with pane content directly from in-air space using finger gestures.

We explored five different ways to arrange four panes in around-device space and the use of an in-air pinch-gesture to select pane content.

Stacked layout (Figure 4a): The four panes are stacked on top of each other to the right of the smartphone, to above the device, two below the device. Each pane's bottom-left x-y position aligns with the device's bottom-right x-y position. If the user moves the device, the panes follow to maintain their position relative to the device. With previous work [1] showing that social acceptability related concerns are raised for around-device gestures taking place beyond 30cm from the device, we limit our panes to a 30×30cm area. The screen's 1080×1776 pixels are mapped to 1080×1776 in-air 'pixels', each approximately 0.028×0.017cm large. The two middle panes are each 15cm high; one of our earlier studies showed this gives the user enough vertical space to perform in-air gestures inside a pane without accidentally entering (and so switching to) an adjacent pane. The top and the bottom panes extend infinitely in the upward respectively downward directions.

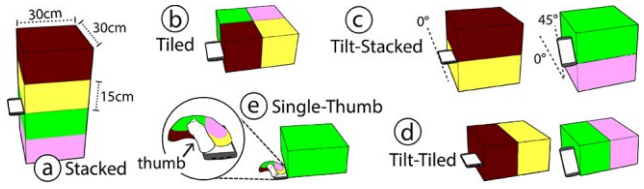

Figure 4. Five AirPane layouts.

Tiled layout (Figure 4b): The four panes share the space used for a single pane in the Stacked layout. This requires less in-air hand movements for pane switching, but more precise movements when interacting with pane content. Each pane measures 15×15cm and extends infinitely upward and downward. The corresponding in-air 'pixel' measures approximately 0.014×0.008cm.

Tilt-Stacked layout (Figure 4c): Similar to the Stacked layout, each pane uses a 30×30cm area (the top pane extends infinitely upwards, the bottom pane downwards, in-air 'pixels' measure approx. 0.028×0.017cm). The four panes are grouped in two pairs, only one pair is accessible at a time. When the device is tilted at an angle less than 45°, the user can switch between the first pair of panes by moving the finger vertically above or below the device. When the device is titled at an angle more than 45°, the other pair is accessible. This provides a larger area for each pane and requires less vertical movements than with the Stacked layout, but introduces a tilting action.

Tilt-Tiled layout (Figure 4d): Again, panes are grouped in pairs and a tilt angel is used to switch between pairs. Each pane uses a 15×30cm area (extended infinitely upward and downward, in-air 'pixels' measure approximately 0.014× 0.017cm). Horizontal in-air movements are used to switch between panes in a pair.

Single-Thumb layout (Figure 4e): Only one 30×30cm pane (extending infinitely upwards and downwards, in-air 'pixels' measure approximately 0.028×0.017cm) is accessible at a time beside the device. The user switches between panes with in-air thumb selections using the left thumb. The in-air thumb-space is divided into four approximately equally sized regions arranged in two layers (cf. inset Figure 4e). Moving the thumb into another area and selecting with the Touch method from Study 1 switches to the selected pane.

To inform our design choices regarding pane switching for our m-commerce demonstration interface, we studied 15 right-handed smartphone owners' performance with the five AirPane layouts (2 female, no participant had previously participated in any pilot or earlier study; mean age 21.1, s.d. 1.8). We used a low-level task where participants had to switch between four panes and select items in the panes.

Task

A trial consists of selecting first a blue and then a yellow 'target item', positioned in two different panes. The other

two panes contain one red distractor item each. The squared items are positioned at random positions inside the pane. The blue target is labelled "1", the yellow target is labelled "2". The panes are numbered 1 to 4 (to provide a clear visual identifier for each pane). A black circular cursor on the screen provides feedback about the position of the index finger inside the active in-air pane. Figure 5a and b visualize a selection of the yellow target item; when the cursor enters the target, the target is highlighted in green. With the cursor inside the target, the participant separates the pinched thumb and index finger by at least 1cm to invoke a selection.

Figure 5. a) A participant moves the 'pinched' in-air fingers to steer the on-screen cursor over the target. b) Releasing the pinch with the cursor over the target selects it. c) Start screen.

After successfully selecting the first target the participant proceeds to find (i.e., switch to) the pane with the yellow target. Successfully selecting the yellow target ends timing and shows the start screen for the next trial (Figure 5c). An on-screen tap on the "start" button starts timing for the next trial. Selections in distractor items are ignored. Selections outside items or a selection of the yellow target before the blue target are also ignored, but the trial is marked as an error trial. Marked trials are re-queued at a random position among unfinished trials within a block of trials.

The in-air space available for an item inside a pane depends on the used pane layout. This suggests a possible trade-off between the ease and speed with which a user can switch panes and the ease and speed with which in-air items can be selected. Accordingly, we used three different on-screen item sizes to investigate such a possible trade-off: 100×100, 200×200, and 300×300 pixels.

Each participant completed a block of five (error free) trials for each of the 5×3 layout-size combinations. Participants had six practice trials with each layout before they started with the first block of timed trials. The presentation order of the five layouts were counterbalanced between participants and the three item sizes were presented in a random order within each layout. We instructed participants to finish each trial as quickly and accurately as possible. Participation lasted 45 min (incl. instruction, practice trials, and breaks).

Results

Error trials: 135 trials were marked with an error. These trials were about equally distributed across participants, layouts and target sizes.

Trial time: For the error free trials we use the median trial time for each layout × target size combination for each participant (i.e., 15 trials). This data was right skewed and we performed a logarithmic transformation (which resulted in a distribution close to normal) before analyzing the data.

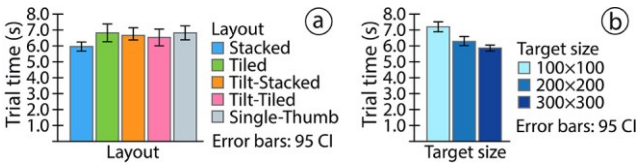

Figure 6. Geometric mean trial times, a) layout, b) target size.

Figure 6 shows the geometric mean selection time for the five layouts and for the three target sizes. A two-way RM-ANOVA showed significant main effects for both layout ($F_{4,56} = 2.70$, $p < 0.05$, $\eta^2 = 0.16$) and target size ($F_{2,28} = 50.7$, $p < 0.0001$, $\eta^2 = 0.78$) but not a significant interaction effect. Bonferroni corrected post-hoc pairwise comparisons of layouts showed that the Stacked layout (5.96s) was significantly faster than all other layouts (all p's < 0.005), which did not differ and were 10.7% slower (or more): Tiled 6.85s, Tilt-Stacked 6.64s, Tilt-Tiled 6.60s, Single-Thumb 6.68s.

As expected, large items are faster to select than smaller (Figure 6b, 100×100 7.40s, 200×200 6.44s, and 300×300 5.97s). Bonferroni corrected post-hoc pairwise comparisons of the three target sizes showed that each pairwise comparison was significant (all p's < 0.016) with the larger target in each comparison being faster than the smaller.

Summary: Our results suggest that the Stacked layout is fast since no secondary activity is needed to switch to another pane, such as tilting the device or moving a second hand, as with Single-Thumb. A vertical hand movement is enough with Stacked. Many participants commented on the convenience and ease with moving the in-air hand in vertical directions, i.e., up and down for switching panes, compared to moving the hand toward or away from the body. A few participants mentioned that they had slight difficulties getting used to tilting the device, especially in early trials, which may explain the somewhat inferior performance with the tilt-based layouts. Our results also indicate that the trial time is strongly influenced by target size. Naturally, large items are easier and faster to select than small items. The non-significant layout×size interaction effect indicates a consistency between all layouts, with no layout suffering more (or less) due to too small in-air targets. Next, we demonstrate how we used our insights about effective switching mechanisms and around-device pane layouts to design an m-commerce interface that utilizes AirPanes and two-handed interaction.

USING AIRPANES IN AN M-COMMERCE INTERFACE

With millions of users purchasing products online from their smartphones (the number of online purchases from smartphones has steadily increased over the past years and the growing trend is expected to continue [35, 36]), an m-commerce scenario is a realistic scenario to showcase the use of AirPanes. We first studied the responsive websites and mobile apps (versions installed from Google Play [12]) from 15 popular online retail services [2] (listed in Figure 7a) to learn about common m-commerce interface structures and functionality. We found that all studied interfaces use similar

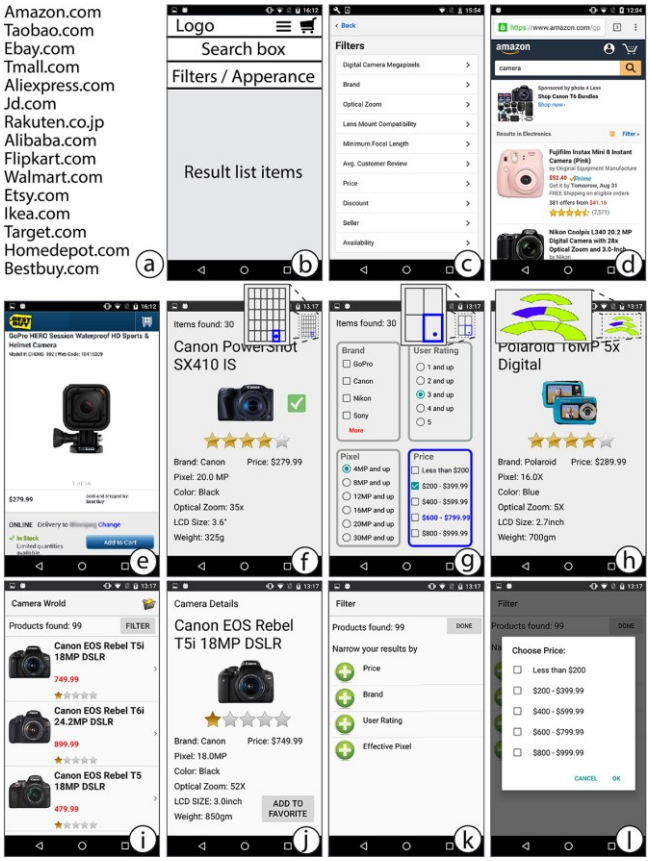

Figure 7. a) Studied m-commerce interfaces. b) General UI structure of the surveyed interfaces. c) Filter selection pane in a mobile app. d) A responsive website's overview list. d) Detail item view in a mobile app. AirPanes interface: f) product details, g) filter options, and h) product detail view for a bookmarked item. Classic touch interface: i) scrollable overview list of search results, j) product detail view, k) filter view, and l) modal view with filter options.

interface components to provide the necessary functionality, as summarized in Figure 7b. A text field is used for product search. Filter functionality is accessed through a "filter" button (most often positioned close to the search functionality) which opens a separate pane that displays the provided filter options (Figure 7c). Search results (filtered or unfiltered) are displayed below the search box in a scrollable overview list (Figure 7d), or in a grid. A thumbnail image and general product information are provided for each item in the list. A tap on an item opens up a new pane which shows further details about the product (Figure 7e). A tap on a "back" button displays the overview list again. From the detail pane, the user can add the item to the shopping cart (and/or favorite list) with a tap on an "add" button.

Our AirPanes interface includes the same functionality as found in conventional m-commerce interfaces. We present the design based on a scenario where the user wants to buy a camera. To issue a product query, the user enters a search text in a text field, as in conventional m-commerce interfaces. Instead of displaying overview information for each camera in the result set in a scrollable list on the screen, for the AirPane interface we off-load the result items onto a 30×30cm in-air pane to the right of the smartphone. We use the best performing Stacked layout from Study 2, and we provide access to filter functionality in a second 30×30cm pane below the results pane. The user can switch between these two panes by moving the in-air hand up and down, crossing an imaginary horizontal plane (as defined by the smartphone's touch-surface). As soon as the in-air finger enters the results pane with camera items, the user sees full information about one of the camera items on the screen, as depicted in Figure 7f. What camera is shown depends on the position of the in-air finger. The camera items in the results pane are arranged in a n×m matrix with equally sized cells filling out the pane. The size of the cells depends on the number of items in the result set. An on-screen visualization with a blue cursor (inset Figure 7f) provides feedback about the in-air finger's current position within the pane.

When the in-air finger enters the filter pane, below the results pane, the user sees filter options on the screen, as depicted in Figure 7g. The filter pane uses a Tiled layout that divides the pane in four 15×15cm large sub-panes, one for four different filter categories (Brand, User ranking, Pixels, Price). Again, the finger's position within the pane is visualized in an on-screen overview (inset Figure 7g) and the current sub-pane is marked with a blue border. The available filter options in each sub-pane are arranged top-to-bottom in the in-air sub-pane (as on the screen). Each option in a category spans horizontally across the whole in-air sub-pane (15cm), the number of options determine the 'height' of the in-air area used for each option (e.g., with five options, each in-air area measures 15×3cm). If more than five filter options are available within a filter category, as with the "Brand" category in Figure 7g, the user can move the in-air finger below the filter pane into a new 30×30 pane where the additional options are available.

The user issues a pinch gesture (as used in Study 2, Figure 5) to select filter options. The pinch gesture is also used when browsing result items inside the topmost in-air pane. With a pinch inside a camera item, the item is added to a list of favorites for later quick access and further inspection and a check mark is shown on the screen (a second pinch inside the same item removes it from the list, and the check mark from the screen). The list of bookmarked items can be inspected using the in-air thumb-space (as in Study 1, cf. Figure 2). To activate the thumb-space the user moves the right hand outside any in-air pane (e.g., moving the hand to the right thigh) and lifts the left thumb into thumb-space. Favorite items are arranged in layers inside thumb-space, as Studied in Study 1. The number of layers and the size of the items are dynamically adjusted according to the number of items that has been added to the list. When the thumb enters an item the corresponding camera information is displayed on the screen. An on-screen visualization informs about the current position within thumb-space, as shown in Figure 7h. The thumb-space is deactivated when the user moves the right

hand into any in-air pane to the right of the smartphone. The results pane which is the topmost pane extends infinitely upwards. The filter pane (with its sub-panes) in the middle is 15cm high and the bottom most pane with additional filter options extends infinitely downwards.

Study 3: AirPanes vs. Conventional Interface Design

We evaluated the efficiency of our AirPanes interface against a touch-based m-commerce interface. The touch interface provides the same functionality as the AirPanes version. The functionality is organised in multiple views, as typical for popular m-commerce interface and shown in Figure 7. Search results are presented in scrollable list with overview information for each item (Figure 7i). A tap on an item shows its detail view (Figure 7j), where the user can add the item to the favorite list. A check mark signals if the item is already in the favourite list (as in Figure 7f) and the "add" button is substituted by a "remove" button. From the view with the result list the user can tap on a "Filter" button to narrow down the result by first selecting a filter category in the filter view (Figure 7k), and then select desired filter options in a second view (Figure 7l). Access to the list of bookmarked favorites is provided from the view with the result list by tapping the "favorite" button in the top-right corner of the screen. The favorites are presented in a scrollable list with overview information for each item, exactly as in the main view with search result. A tap on an item shows its detail view. From all views, a tap on the back button in Android's navigation bar (bottom of the screen) switches back to the previous view.

Twelve right-handed male smartphone owners (age 23.2 years, s.d. 3.2) participated. All were new to the concept of around-device interactions and none had participated in any previous pilot or earlier study within our AirPanes project.

Task

The study task covers typical steps taken when searching for a camera to buy online. First, a text prompt is displayed on an external monitor, e.g., "Apply the following filters: Price between $200 and $400, user rating 3 and higher, then find the lightest camera with a weight between 400g and 500g" and "Apply the following filters: Brand: Nikon and Sony, Pixels: 10MP and up, then find the camera with the greatest optical zoom between 25X and 40X". The task prompt always includes two filters in varying combinations, one criteria range (*between* value X and Y for either optical zoom, LCD size, or weight), and one superlative (smallest, greatest, lightest, or heaviest). After reading the prompt (which remains visible throughout the trial), the participant taps a "start" button on the smartphone screen, this starts the trial timer. We exclude the entering of a search query from the task since this is done in the same way in the two interfaces. Instead, the task starts with a default set of 99 cameras. Now the participant needs to apply the two requested filters. After applying the filters either 10, 20, or 30 cameras remain in the results set (the number varies randomly between trials). The participant can now start inspecting the product details of the remaining cameras and can bookmark candidate cameras to "favorites" that correspond to the selection criteria (e.g., weight between 400g and 500g, 4 to 7 cameras per trial). After this, the participant accesses the favourite functionality to find the one and only camera that matches the complete task prompt. When the participant believes having found the correct camera, the trial ends with a tap on a "buy" button (when using the touch interface) or with a thumb-tap from the corresponding item inside thumb-space (when using AirPanes, cf. Study 1 and Figure 2g). If it is the correct camera trial time stops and the task prompt for the next trial is displayed on the external display. If it was incorrect, the screen flashes in red and the participant can continue the search for the correct camera.

Each participant performed nine trials with each interface (three trials per result set size (10, 20, 30) in random order). Six participants started with AirPanes, six with the touch interface. We demonstrated the two interfaces and showed how the filter functionality and the bookmarking feature would help them to speed up their searches. We instructed participants to try to finish each trial as quickly as possible. Each participant had two practice trials with each interface before starting with the timed trials. Participation lasted about 1 hour (including instructions, practice, and breaks).

Results

We analyze the trial time and the time participants spent with each of the three sub-activities: applying filters (*filter time*), inspecting items and bookmarking items in the filtered set of cameras (*results time*), and inspecting and selecting from the bookmarked favorites (*favorite time*). For each separate time measure we use the median time for each interface × results size combination for each participant. The data was right skewed and we performed logarithmic transformations (which resulted in a distribution close to normal for all four time measures) before analyzing each time measure.

Trial time: The trial time is shown in Figure 8a. The geometric mean trial time was 57.3s with AirPanes and 116.1s with the touch interface. The difference corresponds to 50.6%. A RM-ANOVA (interface, results size) showed a significant effect for interface ($F_{1,11} = 222.60$, $p < 0.0001$, $\eta^2 = 0.95$) and for results size ($F_{2,22} = 48.64$, $p < 0.0001$, $\eta^2 = 0.82$), but no significant interaction effect. The trial time increased with increasing results size: 63.2s with 10 items, 86.3s with 20 items, and 99.7s with 30 items. Bonferroni corrected post-hoc pairwise comparisons between results sizes showed a significant difference for each pair (all p's < 0.016).

We turn to our analyses of the three separate sub-activities the participants performed to find the explanation for the great overall difference between the two interfaces.

Filter time: The time spent with the first sub-activity is shown in Figure 8b. A two-way RM ANOVA showed that across the three result sizes participants needed significantly more time ($F_{1,11} = 22.80$, $p < 0.001$, $\eta^2 = 0.68$) to apply the filters using AirPanes (15.1s) than with the touch interface (11.0s). As expected, there was no significant effect for

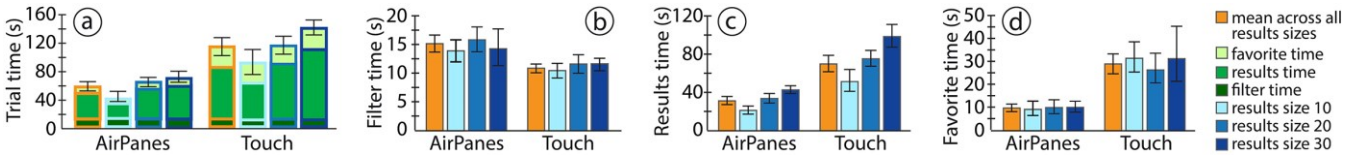

Figure 8. Geometric mean times. a) Trial time, b) filter time, c) results time, and d) favorite time. Error bars: 95 CI.

results size (results size 10: 11.1s, result size 20: 14.6s, results size 30: 13.0s) or a significant interaction effect. Recall that the results size was relevant only after that the filters had been applied. Accordingly, the time difference (3.9s) between the two interfaces mainly comes from selecting the filter controls, which was more challenging (and unfamiliar) with in-air pinches using AirPanes.

Results time: The time spent in the second sub-activity, inspecting items and bookmarking items in the filtered set of cameras, is shown in Figure 8c. Overall, across the three results sizes, participants needed significantly ($F_{1,11} = 157.4$, $p < 0.0001$, $\eta^2 = 0.94$) more time for this activity when using the touch interface (71.3s) than when using AirPanes (30.8s). There was also a significant effect for results size ($F_{2,22} = 50.8$, $p < 0.0001$, $\eta^2 = 0.82$). Comprehensibly, with more items to inspect the longer it takes (results size 10: 35.8s, results size 20: 53.7s, results size 30: 69.5s). Bonferroni adjusted post-hoc comparisons showed that all three results sizes differed (all p's < 0.016). The relative increase in results time with increasing results size was similar for both interfaces (i.e., the RM-ANOVA did not show a significant interface × results size interaction). We attribute the slow inspection and bookmarking time with the touch interface to the frequent (and tedious) switching between items in the result list and their corresponding detail views to find the necessary information. With AirPanes on the other hand, no switching is needed. Instead, the user does only need to move the in-air finger a short distance to enter a new in-air item and so call in its detail information on the screen.

Favorites time: The time spent in the third sub-activity, selecting from the bookmarked items, is shown in Figure 8d. Overall, across the three result sizes, participants were significantly faster ($F_{1,11} = 59.2$, $p < 0.0001$, $\eta^2 = 0.84$) selecting from the favorite set when using AirPanes (9.7s) than when using the touch interface (28.7s). As with the filtering activity, results size had no significant effect on how much time was needed to identify and select the target item from the favorites (results size 10: 17.4s, results size 20: 19.8s, results size 30: 20.8s). Again, as when involved with inspecting the filtered set of cameras, we attribute the disadvantage with the touch interface to the frequent view switching that is required to going back and forth between camera overviews and details. With AirPanes, only a small movement of the in-air thumb is necessary to view information about a new item.

Summary

Our results show that, in comparison to touch input, in-air interactions with AirPanes reduces browsing time by taking the advantage of an in-air 'hover' state. With AirPanes, detailed item information can be inspected without having to perform an action (i.e., tapping) to open the detail view. This makes AirPanes an efficient browsing interface, which does not require frequent switching between views, as in many touch-based interfaces (where the user has to tap on an item in a list, open its details and then tap again to switch back to the list). Our results also demonstrate that users can capitalize on this in-air 'hover' state when using their thumb.

Though promising, a few participants raised concerns about arm fatigue through prolonged use of in-air movements. Furthermore, some participants indicated the possible difficulty for persons with small hands to retain a stable grip of the device while using the in-air thumb space.

DISCUSSION

In the light of these promising results, we discuss design recommendations, potential issues regarding the integration of AirPanes in smartphone applications and present future directions to extend our work.

Design Recommendations

Our investigation offers the following recommendations regarding around-device interactions:

Thumb-space usage: When complemented with in-air index finger input, the in-air thumb-space is well suited for input. It works best for short interactions involving a limited number of items. We suggest using this space to provide access to ten or less items, such as frequently used phone contacts, or to quickly load recently visited websites.

Non-tilt based pane layout: Several participants in Study 2 reported having felt less comfortable with the tilt-based pane layout style as it required rotating the smartphone to switch between sets of panes. This movement also caused a disturbing change in viewing angle. Accordingly, we recommend designers to consider using the Stacked and Tiled layouts for placing panes in around-device space.

Nested pane layout: In our AirPanes design, we used nested in-air panes to provide access to the four filter categories and their options. This nesting strategy increases the number of panes that can be used in an application. We recommend designers to adopt nesting instead of stacking several planes on top of each other which requires larger vertical movements for switching between panes.

Considerations for Integration of AirPanes in Applications

AirPanes scalability: We tested AirPanes with a limited number items in a pane (e.g., a maximum of 30 cameras in the results pane). However, the design can be extended to accommodate a larger number of panes. For instance, a UI

scrolling strategy could be adopted where moving the index finger to a certain in-air location would show new panes. Moreover, an AirPanes application could incorporate other strategies, such as pagination, to divide large sets of in-air items into sub-sets which could be triggered in thumb-space.

Pane organization styles: Our AirPanes implementation considers a mixed pane organization, the panes with filter options and the results pane are stacked. The panes with filter categories were tiled. We envision that AirPanes can work with many alternative pane organizations. For example, AirPanes could be designed for one-handed use using only stacked or tiled layout, as shown in Figure 9a respectively 9b. In these ways, all the panes would be accessible with the index finger on the free hand. Alternatively, a pane organization could leverage two-handed usage by assigning the thumb-space for triggering panes that are accessed with the other hand, as shown in Figure 9c.

Generalization of AirPanes for other Applications: We showed the benefit of using AirPanes for an m-commerce application scenario with a complex analytic task where users had to frequently switch between multiple panes to reach a decision. Thus, the AirPanes design can be generalized to any applications that require navigate and interact with several panes before reaching a decision. Such applications include any shopping, retails or travel booking apps. In addition, other popular apps such as search engines (e.g., Google or Yahoo) are most often displayed with headlines and snippets and the user has to open the results, one by one, which involves frequent view switching. Since AirPanes reduces the need for frequent switching through its in-air 'hover' state, we believe most applications that require view switching would benefit from an AirPanes design. Further investigation is needed to identify a generic design for AirPanes accommodating a wide range of applications.

Usage Context: We explored AirPanes with a task where the user was standing. However, AirPanes could also be used while sitting, or when the smartphone is laying on a flat surface. In such cases, AirPanes could use the surrounding physical surface to leverage haptic feedback into mixed physical and in-air interactions. For instance, AirPanes could use the surrounding surface (e.g., use the user's thigh, a table or a wall as shown in Figure 9d) for applying filters, and then use the in-air space for browsing results.

Limitations and Future Work

AirPanes demands accurate and robust finger detection around the smartphone. Although current smartphones do not provide such features, in a recent study Song et al. [41] showed that a smartphone's camera can be used to recognize in-air hand movement, but in limited directions. Further such explorations are needed to make devices capable of precise around-device hand tracking.

We considered the smartphone to be in 'AirPanes' mode by default. However, further investigation is needed to develop implicit or explicit mode switching mechanisms to activate and deactivate around-device input to counter accidental in-air gesture events. For instance, air thumb input can be implicitly set into "in-air" mode by making it possible for the thumb to engage only when the in-air space is being used with the index finger. Explicit mode switching can be done, for example, with a physical button that the user presses to activate AirPanes mode.

We acknowledge that current mobile apps also support switching between views using right/left swipes, which we did not include in out last study. We believe that using such a swipe mechanism in the touch-based interface would further ascertain our claims as this approach allows only sequential access to items, which we expect to be slow.

In our AirPanes design, we consider in-air thumb-space as a complementary input region. In our future work we will focus on exploring this space for one-handed use cases such as supporting multi-touch interactions (e.g., zoom in/out a map) with thumb within this in-air space. Though the current smartphone design allows holding the device with one hand, the flat shape of the phone prohibits users from reaching the topmost regions of the screen with the thumb in a one-handed usage situation. We will also explore improved off-screen visualizations [14, 16] to manage two-handed around device input, as well as explore ways to mitigate thumb and arm fatigue [23].

CONCLUSION

We have presented AirPanes, a novel technique that utilizes in-air spaces to organize multiple panes that are commonly seen in smartphone applications. We demonstrated the value of AirPanes for carrying out a complex task in an m-commerce application that requires users to frequently switch between multiple in-air panes, such as filters, results, and favourites to make a purchase decision. Through two initial studies, we identified properties of various design factors relevant to AirPanes. Our final study confirmed that AirPanes facilitates complex analytic tasks by reducing task time by 50% compared to a standard touch screen interface. We believe that AirPanes is the first successful step in using in-air spaces for a complete mobile application and so pushes the boundary of current around-device interactions.

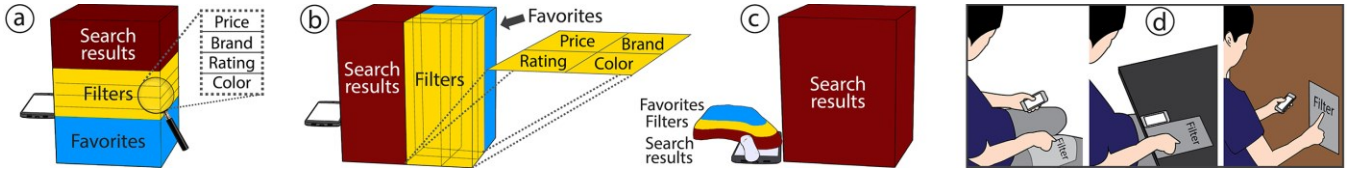

Figure 9. In-air pane organization: a) and b) for one-handed interaction with nested panes, c) for two-handed interaction using in-air thumb space for pane switching. d) AirPanes can leverage a nearby surface for haptic feedback, e.g., a knee, table, or wall.

REFERENCES

1. David Ahlström, Khalad Hasan, and Pourang Irani. 2014. Are you comfortable doing that?: acceptance studies of around-device gestures in and for public settings. In *Proceedings of the 16th International Conference on Human-Computer Interaction with Mobile Devices and Services* (MobileHCI '14), 193-202. https://doi.org/10.1145/2628363.2628381
2. Alexa Top 500 Global Sites. 2017. Retrieved January 04, 2017 from http://www.alexa.com/topsites.
3. Ravin Balakrishnan and Ken Hinckley. 2000. Symmetric bimanual interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '00), 33–40. https://doi.org/10.1145/332040.332404
4. 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), 201-204. http://dl.acm.org/citation.cfm?doid=1449715.1449746
5. W. Buxton and B. Myers. 1986. A study in two-handed input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '86), 321–326. https://doi.org/10.1145/22627.22390
6. Juan Pablo Carrascal and Karen Church. 2015. An in-situ study of mobile app & mobile search interactions. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (CHI '15), 2739-2748. https://doi.org/10.1145/2702123.2702486
7. Didier Casalta, Yves Guiard, and Michel Beaudouin-Lafon. 1999. Evaluating two-handed input techniques: Rectangle editing and navigation. In *Proceedings of the CHI '99 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '99), 236–237. https://doi.org/10.1145/632716.632862
8. Youli Chang, Sehi L'Yi, Kyle Koh, and Jinwook Seo. 2015. Understanding users' touch behavior on large mobile touch-screens and assisted targeting by tilting gesture. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (CHI '15), 1499–1508. https://doi.org/10.1145/2702123.2702425
9. Xiang 'Anthony' Chen, Julia Schwarz, Chris Harrison, Jennifer Mankoff, and Scott E. Hudson. 2014. Air+touch: interweaving touch & in-air gestures. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (UIST '14), 519-525. https://doi.org/10.1145/2642918.2647392
10. Euan Freeman, Stephen Brewster, and Vuokko Lantz. 2016. Do that, there: an interaction technique for addressing in-air gesture systems. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (CHI '16), 2319-2331. http://dx.doi.org/10.1145/2858036.2858308
11. Verena Giller, Rudolf Melcher, Johann Schrammel, Reinhard Sefelin, and Manfred Tscheligi. 2003. Usability evaluations for multi-device application development three example studies. In *Proceedings of Human-Computer Interaction with Mobile Devices and Services* (MobileHCI '03), 302–316. http://link.springer.com/chapter/10.1007/978-3-540-45233-1_22
12. Google Play. 2017. Retrieved January 04, 2017 from https://play.google.com/store?hl=en
13. Yves Guiard. 1987. Asymmetric division of labor in human skilled bimanual action. *Journal of Motor Behavior* 19, 4: 486–517. https://doi.org/10.1080/00222895.1987.10735426
14. Sean Gustafson, Patrick Baudisch, Carl Gutwin, and Pourang Irani. 2008. Wedge: clutter-free visualization of off-screen locations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '08)*, 787-796. https://doi.org/10.1145/1357054.1357179
15. Sean Gustafson, Daniel Bierwirth, and Patrick Baudisch. 2010. Imaginary interfaces: spatial interaction with empty hands and without visual feedback. In *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology* (UIST '10), 3-12. http://dl.acm.org/citation.cfm?doid=1866029.1866033
16. Sean G. Gustafson and Pourang P. Irani. 2007. Comparing visualizations for tracking off-screen moving targets. In *Proceedings of the CHI '07 Extended Abstracts on Human Factors in Computing Systems (CHI EA '07)*, 2399-2404. https://doi.org/10.1145/1240866.1241014
17. Chris Harrison and Scott E. Hudson. 2009. Abracadabra: wireless, high-precision, and unpowered finger input for very small mobile devices. In *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology* (UIST '09), 121-124. https://doi.org/10.1145/1622176.1622199
18. Khalad Hasan, David Ahlström, and Pourang Irani. 2013. Ad-binning: leveraging around device space for storing, browsing and retrieving mobile device content. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '13), 899-908. http://dl.acm.org/citation.cfm?doid=2470654.2466115
19. Khalad Hasan, David Ahlström, and Pourang Irani. 2015. SAMMI: a spatially-aware multi-mobile interface for analytic map navigation tasks. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and*

Services (MobileHCI '15), 36-45. https://doi.org/10.1145/2785830.2785850

20. Khalad Hasan, Junhyeok Kim, David Ahlström, and Pourang Irani. 2016. Thumbs-Up: 3d spatial thumb-reachable space for one-handed thumb interaction on smartphones. In *Proceedings of the 2016 Symposium on Spatial User Interaction* (SUI '16), 103-106. https://doi.org/10.1145/2983310.2985755
21. Khalad Hasan, Xing-Dong Yang, Hai-Ning Liang, and Pourang Irani. 2012. How to position the cursor?: an exploration of absolute and relative cursor positioning for back-of-device input. In *Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services* (MobileHCI '12), 103–112. https://doi.org/10.1145/2371574.2371591
22. Doris Hausen, Sebastian Boring, and Saul Greenberg. 2013. The unadorned desk: exploiting the physical space around a display as an input canvas. In *Proceedings of the 14th IFIP TC13 Conference on Human-Computer Interaction* (INTERACT '13), 140–158. http://link.springer.com/chapter/10.1007/978-3-642-40483-2_10
23. Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. 2014. Consumed endurance: a metric to quantify arm fatigue of mid-air interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*, 1063-1072. https://doi.org/10.1145/2556288.2557130
24. David Holman, Andreas Hollatz, Amartya Banerjee, and Roel Vertegaal. 2013. Unifone: designing for auxiliary finger input in one-handed mobile interactions. In *Proceedings of the 7th International Conference on Tangible, Embedded and Embodied Interaction* (TEI '13), 177-184. http://dx.doi.org/10.1145/2460625.2460653
25. 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), 83-92. https://doi.org/10.1145/2371574.2371589
26. Amy K. Karlson and Benjamin B. Bederson. 2007. ThumbSpace: generalized one-handed input for touchscreen-based mobile devices. In *Proceedings of the 11th IFIP TC 13 International Conference on Human-Computer Interaction* (INTERACT'07), 324-338. http://dx.doi.org/10.1007/978-3-540-74796-3_30
27. Amy K. Karlson, Benjamin B. Bederson, and Jose L. Contreras-Vidal. 2008. Understanding one handed use of mobile devices. In *Handbook of Research on User Interface Design and Evaluation for Mobile Technology,* 86-101.
28. Amy K. Karlson, Benjamin B. Bederson, and John SanGiovanni. 2005. AppLens and launchTile: two designs for one-handed thumb use on small devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '05), 201–210. https://doi.org/10.1145/1054972.1055001
29. Hamed Ketabdar, Mehran Roshandel, and Kamer Ali Yüksel. 2010. MagiWrite: towards touchless digit entry using 3D space around mobile devices. In *Proceedings of the 12th International Conference on Human-Computer Interaction with Mobile Devices and Services* (MobileHCI '10), 443-446. https://doi.org/10.1145/1851600.1851701
30. Sven Kratz and Michael Rohs. 2009. Hoverflow: exploring around-device interaction with IR distance sensors. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services* (MobileHCI '09), Article 42, 4 pages. https://doi.org/10.1145/1613858.1613912
31. Sven Kratz, Michael Rohs, Dennis Guse, Jörg Müller, Gilles Bailly, and Michael Nischt. 2012. PalmSpace: continuous around-device gestures vs. multitouch for 3D rotation tasks on mobile devices. In *Proceedings of the Working Conference on Advanced Visual Interfaces* (AVI '12), 181-188. https://doi.org/10.1145/2254556.2254590
32. Jianwei Lai and Dongsong Zhang. 2014. ExtendedThumb: a motion-based virtual thumb for improving one-handed target acquisition on touch-screen mobile devices. In *Proceedings of the Extended Abstracts of the 32nd Annual ACM Conference on Human Factors in Computing Systems* (CHI EA '14), 1825–1830. https://doi.org/10.1145/2559206.2581158
33. Leap motion. 2017. Retrieved January 04, 2017 from https://www.leapmotion.com/
34. Andrea Leganchuk, Shumin Zhai, and William Buxton. 1998. Manual and cognitive benefits of two-handed input: an experimental study. *ACM Transaction on Human-Computer Interaction*. 5, 4: 326–359. https://doi.org/10.1145/300520.300522
35. Mobile eCommerce Stats in 2016 and the future. 2017. Retrieved January 04, 2017 from http://www.outerboxdesign.com/web-design-articles/mobile-ecommerce-statistics
36. Mobile Marketing Statistics Compilation. 2017. Retrieved January 04, 2017 from http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/
37. Motion Capture System | VICON. 2017. Retrieved January 04, 2017 from https://www.vicon.com/.
38. Takehiro Niikura, Yuki Hirobe, Alvaro Cassinelli, Yoshihiro Watanabe, Takashi Komuro, and Masatoshi

Ishikawa. 2010. In-air typing interface for mobile devices with vibration feedback. In *ACM SIGGRAPH 2010 Emerging Technologies* (SIGGRAPH '10), Article 15, 1 pages. https://doi.org/10.1145/1836821.1836836

39. Michel Pahud, Ken Hinckley, Shamsi Iqbal, Abigail Sellen, and Bill Buxton. 2013. Toward compound navigation tasks on mobiles via spatial manipulation. In *Proceedings of the 15th International Conference on Human-Computer Interaction with Mobile Devices and Services* (MobileHCI '13), 113-122. https://doi.org/10.1145/2493190.2493210
40. Project Soli. 2017. Retrieved January 04, 2017 from https://atap.google.com/soli/
41. Jie Song, Gábor Sörös, Fabrizio Pece, Sean Ryan Fanello, Shahram Izadi, Cem Keskin, and Otmar Hilliges. 2014. In-air gestures around unmodified mobile devices. In *Proceedings of the 27th Annual ACM symposium on User Interface Software and Technology* (UIST '14), 319-329. https://doi.org/10.1145/2642918.2647373
42. Martin Spindler, Martin Schuessler, Marcel Martsch, and Raimund Dachselt. 2014. Pinch-drag-flick vs. spatial input: rethinking zoom & pan on mobile displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '14), 1113-1122. https://doi.org/10.1145/2556288.2557028
43. Structure sensor. 2017. Retrieved January 04, 2017 from http://structure.io/
44. Eduardo Velloso, Jayson Turner, Jason Alexander, Andreas Bulling and Hans Gellersen. 2015. An empirical investigation of gaze selection in mid-air gestural 3d manipulation. In *Proceedings of the 15th IFIP TC13 Conference on Human-Computer Interaction* (INTERACT '13), 315-330. http://dx.doi.org/10.1007/978-3-319-22668-2_25
45. Anusha Withana, Roshan Peiris, Nipuna Samarasekara, and Suranga Nanayakkara. 2015. zSense: enabling shallow depth gesture recognition for greater input expressivity on smart wearables. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (CHI '15), 3661-3670. http://dx.doi.org/10.1145/2702123.2702371
46. Xing-Dong Yang, Khalad Hasan, Neil Bruce, and Pourang Irani. 2013. Surround-see: enabling peripheral vision on smartphones during active use. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*, 291-300. https://doi.org/10.1145/2501988.2502049
47. Chen Zhao, Ke-Yu Chen, Md Tanvir Islam Aumi, Shwetak Patel, and Matthew S. Reynolds. 2014. SideSwipe: detecting in-air gestures around mobile devices using actual GSM signal. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (UIST '14), 527-534. https://doi.org/10.1145/2642918.2647380