Pull Gestures with Coordinated Graphics on Dual-Screen Devices

Vivian Shen
Carnegie Mellon University
Pittsburgh, PA, USA
vhshen@cmu.edu

Chris Harrison
Carnegie Mellon University
Pittsburgh, PA, USA
chris.harrison@cs.cmu.edu

Figure 1: In this example sequence, a user pulls an image from a web browser on the lower screen into the airspace between the two screens, and places it into a presentation on the upper screen.

ABSTRACT

A new class of dual-touchscreen device is beginning to emerge, either constructed as two screens hinged together, or as a single display that can fold. The interactive experience on these devices is simply that of two 2D touchscreens, with little to no synergy between the interactive areas. In this work, we consider how this unique, emerging form factor creates an interesting 3D niche, in which out-of-plane interactions on one screen can be supported with coordinated graphics in the other orthogonal screen. Following insights from an elicitation study, we focus on "pull gestures", a multimodal interaction combining on-screen touch input with in air movement. These naturally complement traditional multitouch gestures such as tap and pinch, and are an intriguing and useful way to take advantage of the unique geometry of dual-screen devices.

CCS CONCEPTS

• **Human-centered computing → Gestural input**; *Ubiquitous and mobile devices.*

KEYWORDS

Interaction Techniques; Hand Gestures; Dual-Screen Devices; In-Air Interaction

ACM Reference Format:

Vivian Shen and Chris Harrison. 2022. Pull Gestures with Coordinated Graphics on Dual-Screen Devices. In *INTERNATIONAL CONFERENCE ON MULTIMODAL INTERACTION (ICMI '22), November 7–11, 2022, Bengaluru, India.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3536221.3556620

ICMI '22, November 7–11, 2022, Bengaluru, India

ACM ISBN 978-1-4503-9390-4/22/11.
https://doi.org/10.1145/3536221.3556620

1 INTRODUCTION

Smartphone screens have increased in size over the years to accommodate consumer demand for high quality interactivity and media consumption on-the-go. To maintain "pocketability" while still increasing screen real estate, smartphone manufacturers are now releasing single-screened devices that can fold (such as the Samsung Flip and Fold [10, 11]) or dual-screened devices that hinge (such as the Microsoft Duo [9]). Simultaneously, we are also seeing dual-screened laptops emerging, like the ASUS Zenbook [49]. These example devices are shown in Figure 2.

Across all of these device categories, the interactive experience is simply that of two conventional touchscreens (i.e., 2D finger input). We believe this is a missed opportunity, and that this unique and emerging device form factor creates an opportunity for interesting interactions in the 3D niche between the two screens (Figure 1), which can work synergetically with conventional touch gestures such as tap and pinch. Moreover, the unique orthogonality of the two screens means that out-of-plane interactions on one screen can be naturally supported with coordinated graphics on the other.

Through an elicitation study, we identified a new multimodal interaction that takes advantage of the unique geometry of these angled, dual-screen devices. We call this gesture a "pull", and it is a straightforward way to couple a conventional tap or pinch gesture with an out-of-plane manipulation, either discrete or continuous. For example, a user could eject a USB drive by pulling it from the screen's surface, rather than long-pressing to access a menu. A drop down menu could be tapped as usual to open it, or pulled from the screen in order to open it on the orthogonal screen, affording users greater flexibility in layout. In both cases, the orthogonal screen is perfectly-placed to provide in-situ, coordinated visual

Figure 2: Top row, research prototypes: A) Codex [15], B) FoldMe [20], C) Foldable3D [5], and D) PaperFold [12]. Bottom row, commercial products: E) Microsoft Duo [9], F) Samsung Z Flip [10], G) Samsung Z Fold [11], and H) ASUS Zenbook [49].

feedback, facilitating user manipulation. We note the latter is what differentiates our work from previous above-single-screen (e.g., Air+Touch [7], Marquardt et al. [27]) and multi-device interactions (and thus multi-screen, but without coordinated visual feedback on orthogonal screen axes; e.g., Pick-and-Drop [37], Lyons et al. [25]).

The contributions of this paper are multifold. First and foremost, this work presents a small but novel interaction technique, building on ideas in prior work and extending them into a new multi-screen device context. Our proposed "pull" gestures were inspired and motivated by a 13 participant elicitation study, where this manipulation was the highest-rated new interaction modality. Then, to demonstrate that such sensing is feasible in reality, we created a proof-of-concept implementation, which forms an additional contribution. Finally, to help illustrate how pull gestures might work, we used our proof-of-concept platform to build working demos of ten interactions across three use categories.

2 RELATED WORK

In the HCI literature, Codex [16] was an early example of a dual-screen tablet computer that consisted of two hinged screens. Later Systems such as FoldMe [22], Foldable3D [6], and PaperFold [13] explored use cases of different folding configurations of multi-screen hinged devices. Flecto [21] even provides a framework and tools to prototype graphical user interfaces on such flexible/hinged screen devices. However, none of these papers considered input incorporating above-screen interactions, or how different screen orientations could support in-air manipulations with coordinated graphics. Figure 2, top row, shows some of these research devices. More recently, dual-screen devices have started to launch as consumer products. We now have smartphones that can fold, but with both halves maintaining their touchscreen capabilities, such as the Samsung Galaxy Z Flip and Z Fold series [11,12]. Microsoft released the Surface Duo [10], which is more akin to a small tablet when unfolded. Dual-screen laptops are also being released, such as the ASUS Zenbook [49], with the lower half of the device shared between a touchscreen and physical keyboard. In all cases, the regions created by any folds or hinges simply act as smaller conventional touchscreens. The bottom row of Figure 2 provides reference images for the above commercial devices.

A second and significant body of work exists for multi-computer (and thus multi-screen) interactions [4, 6, 8, 25, 31, 35, 37, 38], as well as multi-monitor (but not touchscreen) setups [18, 24, 26, 28]. Also, work such as TouchCuts/TouchZoom [46] has looked at superior integration of the multimodal inputs (distributed across orthogonal surfaces) available on touchscreen laptops (e.g., trackpad, keyboard, touchscreen). A review of this literature is beyond the scope of this paper. Instead we focus on the two most highly-related areas of prior work: 1) systems exploring or implementing in-air interactions above displays, and 2) systems combining in-air interactions with touchscreen inputs.

2.1 In-Air Interactions Above Displays

The idea of utilizing the space around devices has been well explored in literature, motivated by the small screen real estate that many smartphones and touchscreen devices have. Hinckley et al. [16] and DeepFishEye [36] both focus on interactions happening right above the screen, using capacitance and fisheye cameras respectively. These near-surface interactions are useful for awareness, but are not meant to disambiguate between many different hand gestures. Another application specific to in-air gesturing above displays is creating in-air typing interfaces for mobile devices [32, 33, 39, 47].

On the other side of the spectrum, large interactive surfaces such as tabletops can also be augmented with in-air gestures [2, 14], as it is often difficult to be able to traverse large surface areas with traditional 2D gestures. TouchLight [42] creates its own touchscreen surface and integrates depth cameras to allow gesture-based interaction with the device. There are also many systems that incorporate more general in-air gestures above displays. For example, Hinckley et al. [17] devised a prototype that augmented a touchscreen with IR proximity range sensing, side touch sensitivity and an accelerometer, to be able to capture more natural gestures above and around the device. Similarly, PalmSpace [22] and HoverFlow [21]

Figure 3: Three different participants interacting with the paper prototype used in our elicitation study.

performed coarse hand gesture / depth estimation around the device, while Finger in Air [23] and Wang et al. [41] focus on tracking fingers and finger gestures above the device. Finally, MagiTact [19] also performed around device (and above display) interactions, but rather than using a vision system, it uses a magnetometer sensor.

2.2 In-Air + Touchscreen Interactions

Only a handful of systems have explored the combination of touchscreen input and in-air user manipulation. Notable among these is Air + Touch [7], which uses in-air motion paths (before, during, and after touches) to add modal functionality to otherwise generic touchscreen inputs. The intriguing Digital Vibrons [40] project, extending ideas in Pick-and-Drop [37], allowed users to "pick up" digital content from a touchscreen using a pinching gesture and deposit the file into real-world physical containers for storage and later retrieval. The opposite workflow is explored in Slurp [50] and Slurp Revisited [45], where users can hold a physical eyedropper that "slurps" content from physical objects or digital screens and can "inject" them back out onto a screen or other interface (smart home device, AR glasses, etc.). This work proposes a similar concept as a pull gesture, but relies instead on a physical device rather than a free-hand gesture.

Finally, and perhaps most related to our work is Continuous Interaction Space [27], where the focus is on the continuity of the different interaction modalities, often integrating them into one motion or simultaneously processing state information from both screen and in-air inputs. Of note, two "lifting" interactions from the screen surface are described, conceptually similar to our pull gestures. A commonality of these projects is the use of a single screen (a tabletop SmartBoard in the case of [27]), precluding the ability to consider how an out-of-plane second display could serve not only as an alternate interaction end point, but also naturally and usefully provide coordinated graphics to aid in above-screen manipulation, the focus of this work.

3 EXPLORATORY STUDY

As noted in Related Work, there have been many previous investigations into single-screen devices employing both on- and above-screen interactions. However, there has been limited exploration of dual-screen single devices, especially those mimicking recently released mobile devices (i.e., prior work has generally focused on larger fixed setups). To explore this emerging design space and understand how users might utilize such devices with this form factor, we ran an elicitation study (inspired by [44]). This method excels at idea generation, and the resulting curated designs can exceed the quality of expert output [30].

3.1 Procedure

For our elicitation study, we constructed an acrylic prototype of a dual-screen device. We used a laptop-like arrangement of the screens (as opposed to book-like), mirroring the commercial examples in Figure 2. We also chose a screen size between the Microsoft Duo [9] and ASUS Zenbook [49]. This was introduced to participants as a "futuristic" computer that could interpret any input the user could imagine. To facilitate the discussion, participants were given a series of example digital tasks, delivered verbally, and simulated using graphics on post-it notes. This setup was purposely unconstrained and low-fidelity in hopes of reducing design bias [29, 44] and encourage personalization [34] and speculative design.

We recruited 13 participants (7 female, mean age 21, all undergraduate or graduate students from our institution) for the 30 minute study. All participants owned and regularly used touchscreen smartphones, but none had a multi-screen or folding touchscreen device. The study was conducted at a normal desk with the dual-screen, paper prototype computer placed in front of them, simulating a working environment. Before starting any tasks, participants were encouraged to think outside the box and ignore any technological restrictions they may think exist.

We chose five interactions representative of everyday digital tasks, and these tasks were ones that all smartphone/laptop users would be familiar with (and not have to learn). More specifically, these were: 1) moving icons between two screens, 2) zooming into an image, 3) ejecting a USB drive, 4) changing font size in a document, and finally, 5) previewing the contents of a folder. For each task, participants were asked to think of as many ways to complete the task as they could, speaking aloud and demonstrating actions physically (Figure 3), until they felt satisfied with their ideation. The facilitator did not comment on the designs, other than reminders to think aloud and think outside the box. After each task, participants were read back a list of ideas they generated, and were asked to rank them in terms of preference.

3.2 Results

Our 13 participants generated 22 unique interactions, which we named and categorized into four broad categories organized along two design axes (Figure 4). The first axis was whether one or two screens were utilized in the interaction, and the second axis was whether the participants' interaction remained on the screen, or involved input that lifted from the screen (i.e., above-screen interaction). We used a stacked bar chart in Figure 5 to illustrate the frequency a gesture was elicited, along with color coding for participant preference ranking.

As expected, all 13 participants enumerated interactions that were equivalent to conventional touchscreen gestures, including taps, double taps, and pinches (black-on-white axis labels in Figure 5). However, 12 of our 13 participants demonstrated interactions that utilized the space above a screen or moved between two screens, which would require in-air hand tracking. Three of our participants enumerated other modes of interaction, such as shaking the computer (two participants) or voice input (one participant), but these were surprisingly uncommon, with users preferring to employ their hands for manipulation (even if in new ways). Gesture agreement [43] for move icons, image zoom, USB eject, font size, and preview folder tasks was 28.6%, 19.6%, 22.9%, 17.8%, and 13.9%, respectively.

3.3 Pull Gestures with Coordinated Graphics

Of the interactions generated by our participants, one gesture stood out as both common and novel —- a selection action followed by a withdraw of the hand from one screen into the volume between

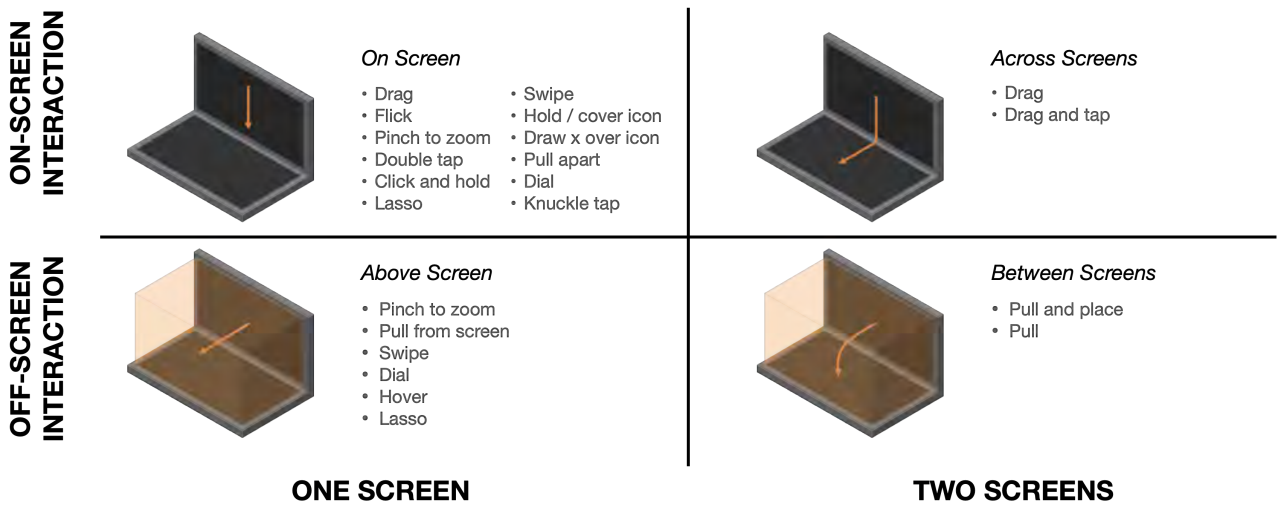

Figure 4: A gesture taxonomy created using post hoc categorization of participant data. The top row contains on-screen interactions, while the bottom row contains off-screen interactions. On the other axis, the left column contains interactions using just one screen, while the right column contains two-screen interactions.

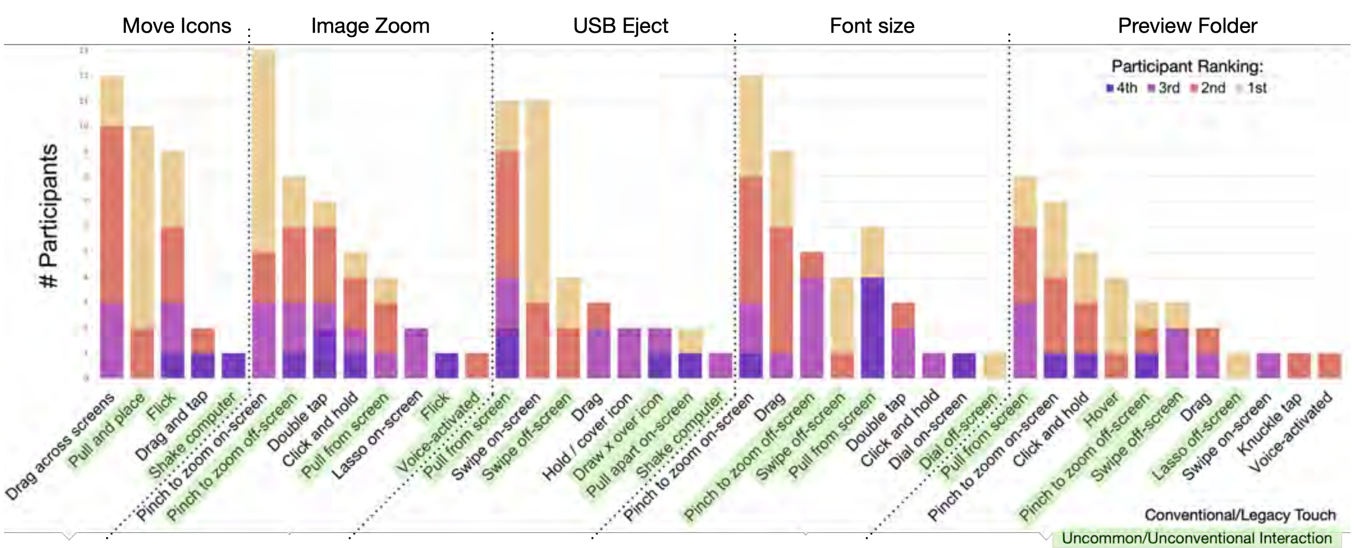

Figure 5: A chart of all gestures presented during the elicitation study. The Y axis is the number of participants who thought of the gesture, and the X axis is labels. The color stacks represent participants' ranking preference, as seen in the legend.

the two screens (i.e., normal to the touch surface). Indeed, this input modality was the highest rated non-conventional interaction (green labels in Figure 5) elicited by our participants overall. In other words, if we set aside conventional touch interactions (Figure 5) — taps, long presses, swipes, drags, pinches, etc. — pull interactions ranked first in three out of five of our example scenarios.

In almost all cases, participants initiated interactions with a pinch- or grasp-like action to select one or more onscreen elements (which acted as a clutch). One participant used a single-finger-lassoing action as the selection mechanism. Twelve of our participants used the term "pull" to describe their interaction idea –– a name we adopt. To terminate the interaction, participants explained they would release the pinch or grasp to drop, throw, place or lock an element or value. In seven cases, participants explained that supporting graphics would appear either in the air (i.e., holography, via an AR headset) or on the other screen to facilitate the action. Participants 8 and 11 noted this would counteract the "loss" of graphical feedback on the initiating screen and help bridge the interactive experience across the two screens. As in-air rendering is not currently practical, we chose to focus on coordinated graphics rendered on the orthogonal screen.

4 PROOF-OF-CONCEPT IMPLEMENTATION

To help convey the feasibility of our idea and illustrate potential use cases, we constructed a proof-of-concept, dual-screen device. While our apparatus is new, it is also built using off-the-shelf hardware and software components. For this reason, we do not claim it as a central contribution, nor do we evaluate its performance. Instead, we used our platform as a vehicle for exploration and illustration. It also demonstrates that such tracking is imminently possible with today's technology. To encourage replication and exploration by researchers and practitioners, we have open sourced our software and hardware at http://github.com/FIGLAB/pullgestures.

4.1 Hardware

To simulate the form factor of a dual-touchscreen device, we placed two 10.1" LCD touchscreens (1024x600, projected capacitive) into a laser cut acrylic frame (Figure 6, left). Mirroring both the commercial examples in Figure 2 and our "paper" prototype used in the elicitation study, we used a laptop-like arrangement of the screens. Our functional prototype features two fisheye cameras [1], one placed where a typical video conferencing camera would be located, and another centered just above the lower screen / keyboard (Figure 6A). As can be seen in the example images in Figure 6 (B & C), this allows the hands to be digitized anywhere in the interactive niche (please also see Video Figure). The displays are driven over HDMI, while the touchscreen and cameras operate over USB, which all connect to an Intel Core i7-10750H laptop running Windows 10. Obviously in a commercial implementation, these components would all be tightly integrated. We also note that very compact, ultra-wide-angle cameras are becoming more prevalent, and recent smartphone and tablet models by Samsung, Apple, Xiaomi and others have field of views of 120° or greater [48].

4.2 Hand Tracking

We track hand input in two distinct ways. First, we monitor for touches on either of the two capacitive touchscreens. For in-air hand tracking, we unwarp both fisheye cameras views and pass these frames to MediaPipe's BlazePose hand pose model [3]. Internally, the model selects the single best hand to report (generally the hand that is least occluded and most frontal) to produce the most reliable hand tracking output. Regressions are used to map Blazepose's 3D positions for 20 hand keypoints from camera space to screen space.

The above process runs at 30 FPS (i.e., camera frame rate) on our laptop. BlazePoze is also capable of running in realtime on smartphones, but continuous background operation would be power prohibitive on today's mobile devices, although we expect that to change with new generations of smartphones with more advanced machine learning hardware acceleration. Moreover, basic heuristics (tracking hands in-air only after a touch event, activating only when the proximity sensor is triggered) could eliminate the need to run continuously in the background. Finally, we also note that we do not currently fuse the touch and in-air data, and instead applications subscribe to whichever data feed they need to enable their particular interaction. However, in future work, the data could be combined in a multi-modal machine learning model that would likely offer superior hand tracking performance by leveraging the high-quality spatial ground truth provided by the touchscreens.

4.3 Front-End Software

We chose to develop our proof-of-concept demo applications in Unity. This allowed us to quickly prototype a variety of interactions

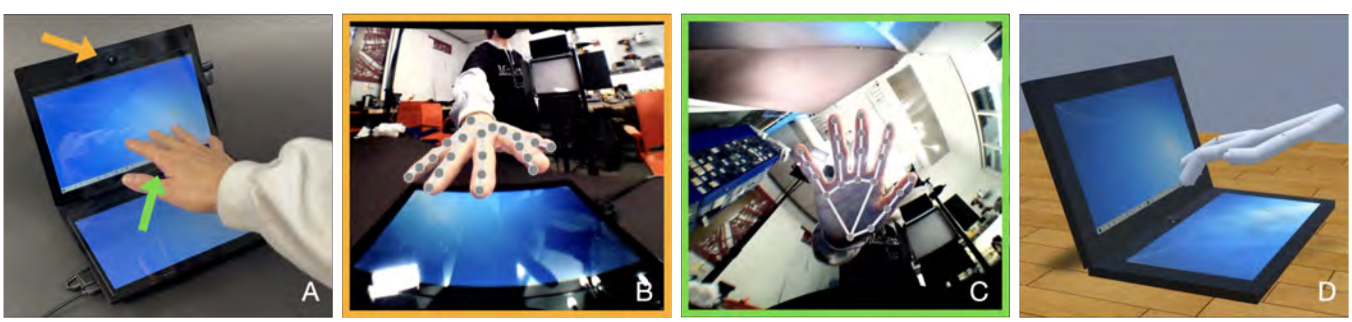

Figure 6: A) photo of our prototype device, with cameras indicated by arrows. B/C) frames of the unwarped video showing a user's hand and labeled key points, from top and bottom cameras respectively. D) our 3D hand model in Unity.

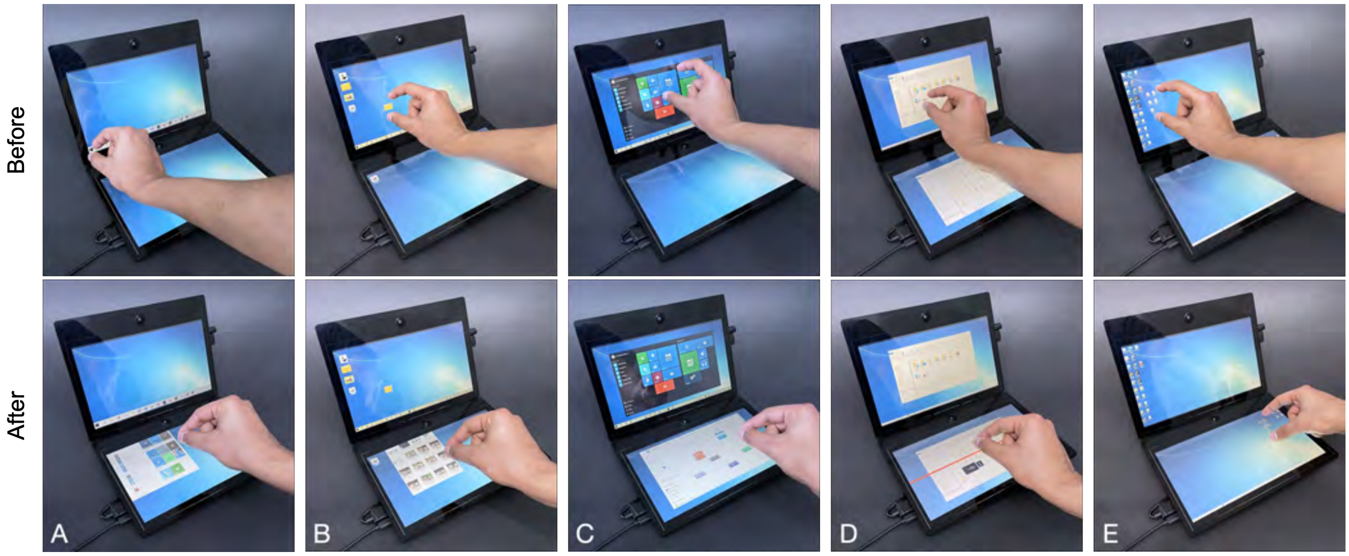

Figure 7: Five example use cases: the top row is before the interaction, and the bottom row is the after results. The examples depicted are: A) opening the start menu, B) previewing the contents of a folder, C) opening an application widget, D) ejecting a USB, and E) rearranging icons on a desktop.

in a unified 3D "desktop" scene. Two orthographic cameras view the scene, which capture each screen's content (Figure 6). Our real-time 3D hand pose is also embedded into this scene in order to trigger and interact with objects (e.g., collision physics), though it is invisible except when debugging (as is the case, for illustration, in Figure 6). Effects such as shadows (a la Interactive Shadows [13]) and animated easing are trivial in Unity, and we employ these effects in our demo examples, described next.

4.4 Example Pull Interactions

To illustrate the potential utility and feasibility of pull gestures on dual-screen devices, we built ten exemplary interactions on top of our proof-of-concept implementation. In all cases, the orthogonal screen is used for supporting coordinated graphics. We succinctly describe these demo applications below, but recommend that readers consult our Video Figure as a superior way to understand the inherently sequential and highly visual nature of our example interactions.

Our example interactions include a toolbar, which can be opened as usual by "tapping" on it, or alternatively, with a pinch and pull on the toolbar to drag open a drawer on the opposite screen (Figure 7, A). In a similar manner, a user can pull on a file or folder to preview the contents (Figure 7, B); releasing the pull causes the preview to snap shut. Users can pull on application icons, such as a calendar, to peek inside (Figure 7, C) without having to open the full application (similar to Mac OS X's "Quick Look" function, which is generally triggered from the keyboard).

It would also be convenient for users to "pluck" content from one document and drop it into another, such as an image from a webpage into a presentation (Figure 1). Similarly, users could pull selected content from a screen and copy it to one of several custom clipboards (Figure 8, A), or pick up icons and place them anywhere on either desktop (Figure 7, E), rather than having to drag across the touchscreen boundary. We also envision users able to swiftly eject e.g., external drives by pulling the icon from the screen until it crosses an ejection confirmation threshold (Figure 7, D).

All of the above examples utilize modal options and destination targets. However, the unconstrained nature of the 3D volume is also well-suited for the manipulation of continuous values. For example, a user could pull on selected text to adjust the font size. After passing a confirmatory threshold, a slider manipulates the font size (Figure 9, bottom). Similarly, pulling on setting icons can allow for continuous adjustment of screen brightness and speaker volume (Figure 9, top). In an HSB color picker widget, the user can manipulate hue (x-axis), saturation (y-axis) and brightness (z-axis) using the 3D volume (Figure 8, B).

5 LIMITATIONS & FUTURE WORK

Many of the limitations of our system stem from design choices rather than inherent problems. For example, our proof-of-concept software pipeline can only track one hand at a time and is thus not capable of multi-hand input. However, it would be interesting to consider how pull gestures could be extended to multi-hand, as well as existing and successful multi-touch gestures. Our focus on pull gestures was initiated by our elicitation study, but there are many other potential multi-plane, mid-air, dual-screen interactions that could be explored in future work.

When designing our hardware, our goal was to create a prototype that resembled a real world product. Readily available hand tracking cameras (e.g., Leap Motion, Microsoft Kinect, Intel RealSense) were too large to be elegantly incorporated, and so we used monocular fisheye RGB cameras instead. Although small, these cameras do not innately provide distance or depth, and so we estimate z-distance using the relative distance between hand keypoints. This is

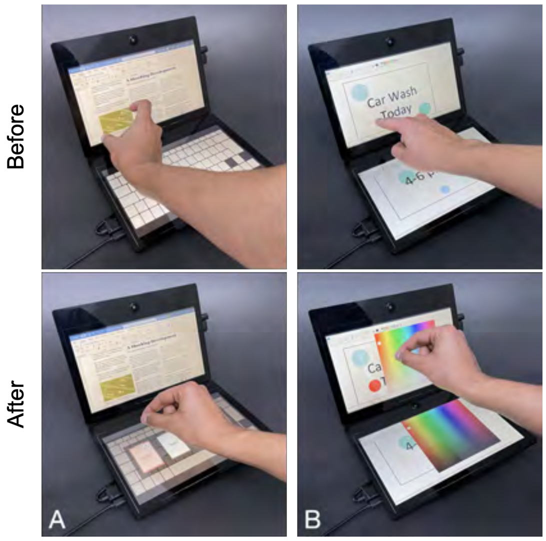

Figure 8: Two more example use cases: A) copying an image snippet to a clipboard, and B) picking a value from the HSB color space.

Figure 9: The final two example use cases, depicted in the sequences that they are performed. Top row: adjusting brightness by first pulling the icon past a confirmation threshold. Bottom row: adjusting font size by similarly pulling the text past a confirmation threshold.

inherently less accurate than true depth sensing. Fortunately, very small depth cameras do exist, such as the TrueDepth sensor found in recent Apple iPhones. We also note that our RGB cameras are subject to tracking issues from poor (or no) illumination, and in general, visual approaches to hand tracking can suffer from occlusion (including self-occlusion from the hand itself). These issues could be mitigated with infrared illumination and the inclusion of more cameras to capture different viewpoints of the hands. Nonetheless, the accuracy of our prototype was sufficient for us to enable pull gestures across different use scenarios.

Another significant drawback of our approach is increased power consumption from tracking hands in-the-air using cameras and deep learning. Power is, of course, at a premium on mobile devices, and so further optimization would be needed to make this approach viable on mobile devices. Fortunately, mobile device chipsets continue to make impressive computational and efficiency advances, and flagship devices now often include hardware-accelerated deep learning capabilities. Indeed, our 3D hand pose model, BlazePose, can run at camera framerate on modern smartphones. To limit power consumption, we envision the pull-gesture-tracking process only starting following touch screen input (e.g., a selection pinch), or periodically checking for a hovering hand (perhaps using the highly efficient proximity sensor), activating at full frame rate only when it might be needed.

Finally, in this work, we considered a laptop-like arrangement of the two screens, though we note that some dual-screen mobile devices can also be oriented like a book, with the fold running vertically. Of course, most of our interactions are immediately applicable in the latter form as well. Nonetheless, as we did not explicitly explore this arrangement in our elicitation study or example applications, we believe there may be new and interesting interactions not yet identified.

6 CONCLUSION

In exploring natural interactions desirable for dual-screen devices, our elicitation study led us to the idea of a "pull" gesture, where the second, orthogonal screen supports the interaction dimension by providing useful graphical feedback. This affords devices greater flexibility in manipulation and layout, and utilizes the two screens in conjunction with each other rather than as two separate interactive experiences. We created a proof-of-concept device by integrating two fisheye RGB cameras into a computer-like prototype that utilized two touchscreen displays. We also created a suite of example interactions built around everyday computing tasks to demonstrate the potential utility of pull gestures. We believe this simple idea complements traditional multitouch gestures such as tap and pinch, and is an intriguing and useful way to take advantage of the unique geometry of recently released dual-screen mobile devices.

REFERENCES

[1] Arducam. 2022. Arducam Fisheye Camera for Raspberry Pi. https://www.arducam.com/product/arducam-raspberry-pi-camera-5mp-fisheye-m12-picam-b0055/
[2] Amartya Banerjee, Jesse Burstyn, Audrey Girouard, and Roel Vertegaal. 2011. Pointable: an in-air pointing technique to manipulate out-of-reach targets on tabletops. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*. 11–20.
[3] Valentin Bazarevsky, Ivan Grishchenko, Karthik Raveendran, Tyler Zhu, Fan Zhang, and Matthias Grundmann. 2020. Blazepose: On-device real-time body pose tracking. *arXiv preprint arXiv:2006.10204* (2020).
[4] Andy Brown, Michael Evans, Caroline Jay, Maxine Glancy, Rhianne Jones, and Simon Harper. 2014. HCI over Multiple Screens. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI EA '14)*. Association for Computing Machinery, New York, NY, USA, 665–674. https://doi.org/10.1145/2559206.2578869
[5] Wolfgang Büschel, Patrick Reipschläger, and Raimund Dachselt. 2016. Foldable3d: Interacting with 3d content using dual-display devices. In *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces*. 367–372.
[6] Ke-Yu Chen, Daniel Ashbrook, Mayank Goel, Sung-Hyuck Lee, and Shwetak Patel. 2014. AirLink: Sharing Files between Multiple Devices Using in-Air Gestures. In *Proceedings of the 2014 ACM International Joint Conference on*

Pervasive and Ubiquitous Computing (Seattle, Washington) *(UbiComp '14)*. Association for Computing Machinery, New York, NY, USA, 565–569. https://doi.org/10.1145/2632048.2632090

[7] 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*. 519–525.

[8] Connor Dickie, Jamie Hart, Roel Vertegaal, and Alex Eiser. 2006. LookPoint: an evaluation of eye input for hands-free switching of input devices between multiple computers. In *Proceedings of the 18th Australia Conference on Computer-Human Interaction: Design: Activities, Artefacts and Environments*. 119–126.

[9] Microsoft Surface Duo. 2022. https://www.microsoft.com/en-us/surface/devices/surface-duo

[10] Samsung Z Flip. 2022. https://samsung.com/us/smartphones/galaxy-z-flip3-5g/

[11] Samsung Z Fold. 2022. https://samsung.com/us/smartphones/galaxy-z-fold3-5g/

[12] Antonio Gomes and Roel Vertegaal. 2014. Paperfold: a shape changing mobile device with multiple reconfigurable electrophoretic magnetic display tiles. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*. 535–538.

[13] Kenneth P. Herndon, Robert C. Zeleznik, Daniel C. Robbins, D. Brookshire Conner, Scott S. Snibbe, and Andries van Dam. 1992. Interactive Shadows. In *Proceedings of the 5th Annual ACM Symposium on User Interface Software and Technology* (Monteray, California, USA) *(UIST '92)*. Association for Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/142621.142622

[14] Otmar Hilliges, Shahram Izadi, Andrew D Wilson, Steve Hodges, Armando Garcia-Mendoza, and Andreas Butz. 2009. Interactions in the air: adding further depth to interactive tabletops. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*. 139–148.

[15] Ken Hinckley, Morgan Dixon, Raman Sarin, Francois Guimbretiere, and Ravin Balakrishnan. 2009. Codex: a dual screen tablet computer. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1933–1942.

[16] Ken Hinckley, Seongkook Heo, Michel Pahud, Christian Holz, Hrvoje Benko, Abigail Sellen, Richard Banks, Kenton O'Hara, Gavin Smyth, and William Buxton. 2016. Pre-touch sensing for mobile interaction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 2869–2881.

[17] Ken Hinckley, Jeff Pierce, Mike Sinclair, and Eric Horvitz. 2000. Sensing techniques for mobile interaction. In *Proceedings of the 13th annual ACM symposium on User interface software and technology*. 91–100.

[18] Ken Hinckley, Gonzalo Ramos, Francois Guimbretiere, Patrick Baudisch, and Marc Smith. 2004. Stitching: pen gestures that span multiple displays. In *Proceedings of the working conference on Advanced visual interfaces*. 23–31.

[19] Hamed Ketabdar, Kamer Ali Yüksel, and Mehran Roshandel. 2010. MagiTact: interaction with mobile devices based on compass (magnetic) sensor. In *Proceedings of the 15th international conference on Intelligent user interfaces*. 413–414.

[20] Mohammadreza Khalilbeigi, Roman Lissermann, Wolfgang Kleine, and Jürgen Steimle. 2012. FoldMe: interacting with double-sided foldable displays. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction*. 33–40.

[21] Sven Kratz and Michael Rohs. 2009. HoverFlow: expanding the design space of around-device interaction. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*. 1–8.

[22] 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 international working conference on advanced visual interfaces*. 181–188.

[23] Zhihan Lv, Alaa Halawani, Muhammad Sikandar Lal Khan, Shafiq Ur Réhman, and Haibo Li. 2013. Finger in air: touch-less interaction on smartphone. In *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia*. 1–4.

[24] Kent Lyons, David Nguyen, Daniel Ashbrook, and Sean White. 2012. *Facet: A Multi-Segment Wrist Worn System*. Association for Computing Machinery, New York, NY, USA, 123–130. https://doi.org/10.1145/2380116.2380134

[25] Kent Lyons, Trevor Pering, Barbara Rosario, Shivani Sud, and Roy Want. 2009. Multi-display composition: Supporting display sharing for collocated mobile devices. In *IFIP Conference on Human-Computer Interaction*. Springer, 758–771.

[26] Regan L Mandryk, Malcolm E Rodgers, and Kori M Inkpen. 2005. Sticky widgets: pseudo-haptic widget enhancements for multi-monitor displays. In *CHI'05 Extended Abstracts on Human Factors in Computing Systems*. 1621–1624.

[27] Nicolai Marquardt, Ricardo Jota, Saul Greenberg, and Joaquim A Jorge. 2011. The continuous interaction space: interaction techniques unifying touch and gesture on and above a digital surface. In *IFIP Conference on Human-Computer Interaction*. Springer, 461–476.

[28] Takashi Matsumoto, Daisuke Horiguchi, Shihori Nakashima, and Naohito Okude. 2006. Z-agon: mobile multi-display browser cube. In *CHI'06 extended abstracts on Human factors in computing systems*. 351–356.

[29] Meredith Ringel Morris, Andreea Danielescu, Steven Drucker, Danyel Fisher, Bongshin Lee, MC Schraefel, and Jacob O Wobbrock. 2014. Reducing legacy bias in gesture elicitation studies. *interactions* 21, 3 (2014), 40–45.

[30] Meredith Ringel Morris, Jacob O Wobbrock, and Andrew D Wilson. 2010. Understanding users' preferences for surface gestures. In *Proceedings of graphics interface 2010*. 261–268.

[31] Miguel A Nacenta, Samer Sallam, Bernard Champoux, Sriram Subramanian, and Carl Gutwin. 2006. Perspective cursor: perspective-based interaction for multi-display environments. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. 289–298.

[32] 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*. 1–1.

[33] Takehiro Niikura, Yoshihiro Watanabe, Takashi Komuro, and Masatoshi Ishikawa. 2012. In-air Typing Interface: Realizing 3D operation for mobile devices. In *The 1st IEEE Global Conference on Consumer Electronics 2012*. IEEE, 223–227.

[34] Uran Oh and Leah Findlater. 2013. The challenges and potential of end-user gesture customization. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 1129–1138.

[35] Antti Oulasvirta and Lauri Sumari. 2007. Mobile kits and laptop trays: managing multiple devices in mobile information work. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 1127–1136.

[36] Keunwoo Park, Sunbum Kim, Youngwoo Yoon, Tae-Kyun Kim, and Geehyuk Lee. 2020. DeepFisheye: Near-surface multi-finger tracking technology using fisheye camera. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. 1132–1146.

[37] Jun Rekimoto. 1997. Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology* (Banff, Alberta, Canada) *(UIST '97)*. Association for Computing Machinery, New York, NY, USA, 31–39. https://doi.org/10.1145/263407.263505

[38] Arne Schmitz, Ming Li, Volker Schönefeld, and Leif Kobbelt. 2010. Ad-Hoc Multi-Displays for Mobile Interactive Applications.. In *Eurographics (Areas Papers)*. 45–52.

[39] Kazuhiro Terajima, Takashi Komuro, and Masatoshi Ishikawa. 2009. Fast finger tracking system for in-air typing interface. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*. 3739–3744.

[40] Radu-Daniel Vatavu, Annette Mossel, and Christian Schönauer. 2016. Digital Vibrons: Understanding Users' Perceptions of Interacting with Invisible, Zero-Weight Matter. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services* (Florence, Italy) *(MobileHCI '16)*. Association for Computing Machinery, New York, NY, USA, 217–226. https://doi.org/10.1145/2935334.2935364

[41] Guo-Zhen Wang, Yi-Pai Huang, Tian-Sheuan Chang, and Tsu-Han Chen. 2013. Bare finger 3D air-touch system using an embedded optical sensor array for mobile displays. *Journal of Display Technology* 10, 1 (2013), 13–18.

[42] Andrew D Wilson. 2004. TouchLight: an imaging touch screen and display for gesture-based interaction. In *Proceedings of the 6th international conference on Multimodal interfaces*. 69–76.

[43] Jacob O Wobbrock, Htet Htet Aung, Brandon Rothrock, and Brad A Myers. 2005. Maximizing the guessability of symbolic input. In *CHI'05 extended abstracts on Human Factors in Computing Systems*. 1869–1872.

[44] Jacob O Wobbrock, Meredith Ringel Morris, and Andrew D Wilson. 2009. User-defined gestures for surface computing. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 1083–1092.

[45] Shengzhi Wu, Daragh Byrne, Ruofei Du, and Molly Steenson. 2022. "Slurp" Revisited: Using'system re-presencing'to look back on, encounter, and design with the history of spatial interactivity and locative media. (2022).

[46] Xing-Dong Yang, Tovi Grossman, Pourang Irani, and George Fitzmaurice. 2011. TouchCuts and TouchZoom: Enhanced Target Selection for Touch Displays Using Finger Proximity Sensing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI '11)*. Association for Computing Machinery, New York, NY, USA, 2585–2594. https://doi.org/10.1145/1978942.1979319

[47] Xin Yi, Chun Yu, Mingrui Zhang, Sida Gao, Ke Sun, and Yuanchun Shi. 2015. Atk: Enabling ten-finger freehand typing in air based on 3d hand tracking data. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. 539–548.

[48] Alex. Yuktanon. 2022. Best Wide Angle Selfie Smartphone. https://www.alexi.xyz/blog/best-wide-angle-selfie-phone.

[49] ASUS Zenbook. 2022. https://www.asus.com/Laptops/For-Home/Zenbook/

[50] Jamie Zigelbaum, Adam Kumpf, Alejandro Vazquez, and Hiroshi Ishii. 2008. Slurp: tangibility spatiality and an eyedropper. In *CHI'08 Extended Abstracts on Human Factors in Computing Systems*. 2565–2574.