tPad: Designing Transparent-Display Mobile Interactions

Juan David Hincapié-Ramos[1], Sophie Roscher[1,3], Wolfgang Büschel[2], Ulrike Kister[2], Raimund Dachselt[2], Pourang Irani[1]

[1]University of Manitoba
Winnipeg, MB, Canada
{jdhr, irani}@cs.umanitoba.ca

[2]Technische Universität Dresden
Dresden, Germany
{first.last}@tu-dresden.de

[3]Otto-von-Guericke-Universität
Magdeburg, Germany
sophie.roscher@st.ovgu.de

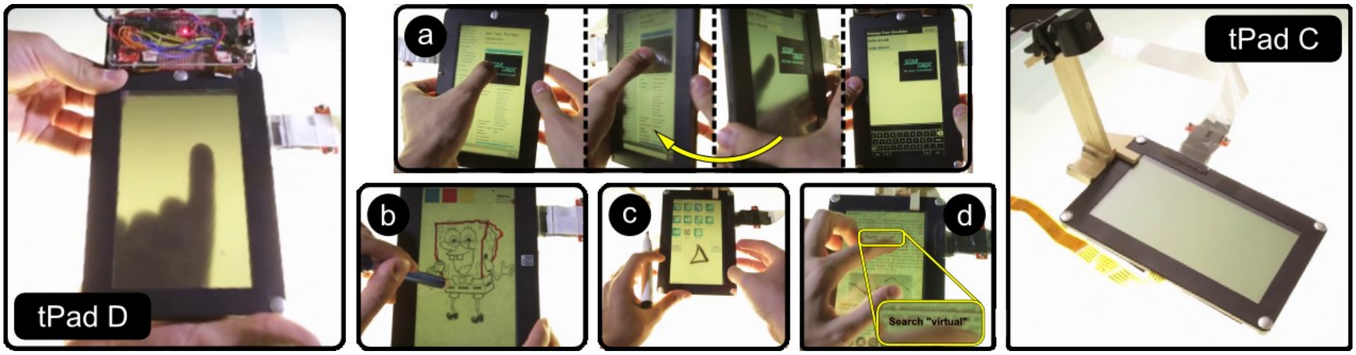

Figure 1. Left: tPad D, a transparent-display mobile with touch and movement sensors enables interactions such as *flipping* (a) and *tracing* (b). Right: tPad C simulates surface capture via an overhead camera, enabling *scribbles* (c) and document *search* (d).

ABSTRACT

As a novel class of mobile devices with rich interaction capabilities we introduce tPads – transparent display tablets. tPads are the result of a systematic design investigation into the ways and benefits of interacting with transparent mobiles which goes beyond traditional mobile interactions and augmented reality (AR) applications. Through a user-centered design process we explored interaction techniques for transparent-display mobiles and classified them into four categories: *overlay*, *dual display & input*, *surface capture* and *model-based interactions*. We investigated the technical feasibility of such interactions by designing and building two touch-enabled semi-transparent tablets called tPads and a range of tPad applications. Further, a user study shows that tPad interactions applied to everyday mobile tasks (application switching and image capture) outperform current mobile interactions and were preferred by users. Our hands-on design process and experimental evaluation demonstrate that transparent displays provide valuable interaction opportunities for mobile devices.

Author Keywords

Transparent Displays; Transparent Mobile Devices; tPad; Flipping; Tap'n Flip; Surface Capture; Contact AR

ACM Classification Keywords

H.5.2 Information Interfaces and Presentation: User Interfaces: Input Devices and Strategies, Interaction Styles.

INTRODUCTION

Advances in display technologies have paved the way for a new generation of mobile devices consisting of transparent displays, some of which are already commercially available (e.g. Lenovo S800). These advances have inspired a flurry of novel designs showcasing interactions not possible on traditional mobile devices [7, 9]. With the exception of a relatively small number of research prototypes that exploit the transparency factor to address known mobile interaction limitations [19, 24, 35], most conceptual depictions focus on mobile augmented reality applications [7, 9].

Unfortunately, such demonstrations of transparent-display mobile devices leave unanswered the important question of how beneficial this technology is for everyday tasks. Ideally, such benefits outweigh the primary limitations of transparent displays (such as color blending [12] and binocular parallax [19]) and can be used in a broad range of mobile applications.

This paper explores the usage of transparent-display mobile devices in everyday tasks. Based on a user-centered design process we propose interaction techniques grouped in four major categories, where each category depends on specific hardware capabilities. The most basic category, *overlay*, depends only on the display transparency. The *dual display & input* category builds upon the capacity for seeing and interacting with the device from both sides. Previously proposed techniques [24, 35] are representatives of this category. The *surface capture* category is based on the device's capacity to image capture real-world objects below the display. Finally, the *model-based* category depends on matching the image of the object below the display with pre-existing virtual models of the objects. Virtual annotations and multimedia content on a paper document [10, 23] are exemplary model-based interactions.

We built two transparent tablet prototypes or *tPads* as we call them and implemented a broad range of applications showing the value of transparent-displays interactions for everyday tasks. tPads use LCD displays with low-opacity filters to provide transparency and support *overlay* interactions. Movement and touch sensors on the tPad-D (Figure 1-Left) enable techniques in the *dual display & input* category. An overhead camera looking down and through the display in tPad-C (Figure 1-Right) enables the *surface capture* and *model-based* categories.

Finally, we conducted a user study to investigate how tPads benefit two everyday tasks: application switching [6, 20] and image acquisition [17]. For these tasks we used two novel techniques: *tap'n flip* and *grabbing*. Results show that tPad interactions performed better and users preferred them over non-transparent alternatives.

Our contributions include: (a) a categorical organization of transparent-display mobile interactions; (b) the implementation of two transparent-display prototypes and several apps; and (c) a user study demonstrating that two common mobile tasks, applications switching and information capture, are easier when using transparent-display mobile devices.

RELATED WORK

Our exploration relates to existing prototypes of transparent display devices, and, at a more general level, to research on digital augmentations through virtual and physical tools.

Transparent Display Devices

Research into transparent displays has largely focused on the window-size format (made possible by projectors and diffusive films) within the context of spatial augmented-reality (SAR). *SAR* systems rely on *spatial alignment* (display fixed in space) which facilitates overlaying digital content onto physical objects. SAR applications are limited to fixed settings [5], such as for use on industrial machinery [25] and vending machines [8]. More recently, researchers explored immersive experiences like the HoloDesk [13] and SpaceTop [18], where users directly interact with virtual objects by moving their hands behind a transparent display.

In this paper we focus on scenarios where the transparent display is mobile, i.e. non-fixed. With the first few mobile devices becoming available to the public and electronic companies announcing the mass production of such displays, it is important to investigate the broad range of interaction techniques made possible through such a form factor and their benefits for everyday tasks.

To the best of our knowledge, only a few projects explored this direction. Lee et al. [19] focused on how binocular parallax affects selection of real world objects through a transparent display and proposed the *binocular cursor*. Others focused on touch interaction on the back of the device (as supported by [33]). LucidTouch [35] simulates such transparency with a camera-based see-through portable device (pseudo-transparency). With LucidTouch users were able to overcome the fat-finger problem and acquire targets with higher precision using all 10 fingers simultaneously. The authors in LimpiDual [24] also studied back-of-device input, front and dual selections using an optical see-through display. Their results showed that while back of the device touch has indeed higher precision, it is slower than front and dual touch. Additionally, Glassified [30] embeds a transparent display into a ruler to augment hand drawings.

Mobile AR, Magic Lenses, and Tangible Views

Augmented Reality (AR) enhances the real world by embedding digital content onto it. Traditional AR relies on mobile displays carried by users (e.g. retinal, HMDs, phones and projectors, etc.), allowing the augmentation of virtually any object within the display's field-of-view but requiring complex registration (e.g. 3D location, object recognition) and rendering (e.g. perspective correction). Moreover, mobile displays present limitations in terms of resolution, focus, lighting and comfort. A comprehensive reference to AR can be found in [3, 5].

A transparent-display mobile device allows for what we call *contact augmented reality* (cAR), that is, when the device augments an object directly below and in contact with its display. The resulting interactions resemble known concepts shown with tool-glasses and magic lenses [4], tangible views [16, 31], and aspects of mobile AR [3].

Toolglass and magic lens widgets for WIMP interfaces sit between the application and the cursor to provide richer operations and visual filters on the digital content. For example, a toolglass widget could have different areas each with unique operations, such that by clicking the target object through the toolglass the digital content is modified in different ways. Similarly, the magic lens widget could hide or show details of an underlying digital object by simply placing the widget on top of it. Moving beyond the WIMP environment, Mackay et al.'s A-book implements tool-glass and magic lenses into a biology lab book [23].

Tangible views provide complementary displays for content visualized on tabletop computers. Spindler et al. [31] used spatially tracked, handheld lenses made of cardboard and propose an interaction vocabulary including: translation, rotation, freezing, gestures, direct pointing, a toolbox metaphor, visual feedback and multiple views. Other researchers built such tangibles using transparent acrylic plates with fiducial markers [16] and 3D head tracking [29].

Transparent-display mobile devices benefit from the above interaction paradigms as shown in our *model-based interactions* category. However, our exploration shows that there exist even simpler interactions which add significant value on their own to common mobile tasks. We propose interactions which do not require knowledge of the underlying object like the ones covered in the *overlay* and *dual display and input* categories.

ELICITATION OF INTERACTION TECHNIQUES

We followed a user-centered design approach with the goal of eliciting interaction techniques for transparent-display mobile devices. Six participants (two female) used mock devices (non-functional, made of cardboard, acrylic and non-permanent markers) to illustrate how they would use a transparent-display mobile and to draw the user interface (Figure 3). The mock device resembled a 10×15cm mid-size tablet to cover more cases than possible with a smaller device, yet remain highly mobile. We asked participants to consider using the device for everyday activities (e.g. reading papers, browsing the web) and general-purpose applications (e.g. calculator, messaging.). We asked follow up questions aimed at clarifying the proposed interactions.

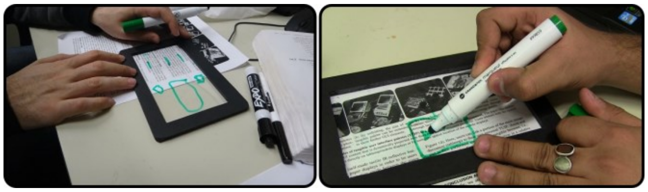

Figure 3. Hands-on user-centered design sessions for transparent display mobile interactions.

After the initial sessions the design team elaborated on the collected interactions focusing on those which are unique to transparent displays; i.e. that are not possible with existing devices. We discarded mobile AR interactions, as they are possible today using video-based see-through displays. This process resulted in 69 potential usages and functionalities, including application-specific and device-wide interactions. For instance, participants envisioned using a transparent-display mobile as an assistive magic lens to add virtual layers to paper documents; when multi-tasking on the mobile, participants proposed flipping it to get a second display and have two applications running simultaneously.

We iterated over the interactions, refining them and finding possible application scenarios. Through this process we identified 10 higher level interaction techniques and grouped them into 4 categories. The following sections present the categories and their interaction techniques, as well as two prototypes and applications that implement them.

TRANSPARENT MOBILE INTERACTIONS

We grouped interactions into four categories according to the display capabilities necessary for their implementation.

Overlay Interactions

In overlay interactions users place the transparent-display mobile device on top of any object (e.g. a Polaroid picture, a paper printout, a tree leaf) to create direct contact between the display and the object, so that users can see the object through the display. Overlay interactions require no knowledge about the object and they are based solely on the users' capacity to see through the display. Two interaction possibilities emerged from our study: *tracing* and *querying*.

Tracing refers to drawing based on the object as seen immediately beneath the display. The tPad *Tracer* software demonstrates basic tracing scenarios. For example, a user can trace items in a picture to create an artistic reproduction (Figure 2a). Users can also discuss changes to a building's structural plan by overlaying the device on top of the construction blueprints and sketching different alternatives; all without damaging the original (Figure 2b).

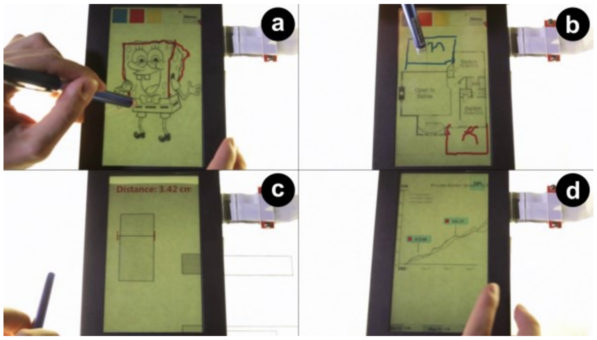

Figure 2. Overlay Interactions. Tracing (a, b). Querying with the Ruler (c) and the GraphExplorer (d).

Querying alludes to taking advantage of knowing the real-world size of display pixels and their correspondence to the underlying object. For example, a 30 pixel line might cover a 1 cm stretch in real-world dimensions. Querying exploits this physical/ virtual correspondence. For example, the *Ruler* application allows users to draw a line to obtain the corresponding length in cm/inches (Figure 2c). The tPad *Graph Explorer* application, after entering the right parameters, allows users to query values on a printed chart (Figure 2d); a user can query the value of a particular bar on a bar chart, or the highs and lows of a trend line.

Dual Display and Input Interactions

These techniques allow users to use the display from either side both for input and output. Given the existing research on back-of-device interactions [24, 33, 35] we focused on two novel interactions: *flipping* and *tap'n flip*.

Flipping relates to physically rotating the device to its rear side in order to access alternative visuals, menus or a secondary display. Unlike other dual-display devices [14] or opaque lenses [31], in a transparent-display mobile only one of the "displays" is usable at a given time. Applications can be attached to a specific side. Figure 4a-c shows a user

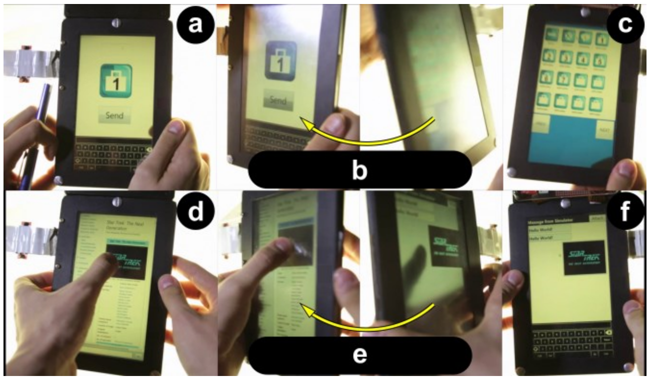

Figure 4. Dual display and input interactions. Flipping (a-c) to go to the home screen, and Tap'n Flip (d-f) to copy+paste.

flipping the device to transition from one application to the main menu. A new application can be launched on this side, and flipping back gives access to the initial application. A practical scenario for *flipping* is notification handling, a situation where users often lose their working context. Upon seeing a notification alert the user can flip the device to access the notifying application. When the task finishes the user flips to the original side to resume the previous task. tPad applications running on different sides can share a context object which allows them to share information at every flip. For example, flipping from the *Ruler* to the *Calculator* application sets a context string of the current value in the ruler (e.g. "7.4 cm") to be used for calculations.

Tap'n Flip allows a user to transfer a particular user interface element while flipping from one side to the other. For example, Figure 4d-f shows a user *Tap'n Flipping* on an image in the browser to send it as a chat message.

Surface Capture Interactions

Surface Capture interactions take the image of the object right below the transparent display (see the *surface capture* capability outlined in the hardware section below) as input for the interaction. Three interaction techniques are part of this category: *grabbing*, *markers* and *scribbles*.

Grabbing refers to capturing an image of the object (or part of it) beneath the display (Figure 5ab). A grabbing-based application can process the image and/or extract relevant information. For example, the tPad *Capture* application allows users to specify a cropping region before the image capture. The captured image is stored in the user's photo album and made available to be exported to other services like OCR, translation or social media.

Markers refers to reading markers (e.g. QR code) by simply moving the device on top of them. In contrast to traditional code readers, surface capture supports constant monitoring for markers (Figure 5cd). When a marker is found, the tPad *QReader* application reads the encoded value and handles the event. When the marker encodes a URL, the QReader launches the Browser pointing to the desired URL. A marker can also embed system settings such as "*silence mode"* or *"set alarm at 6:15am"*.

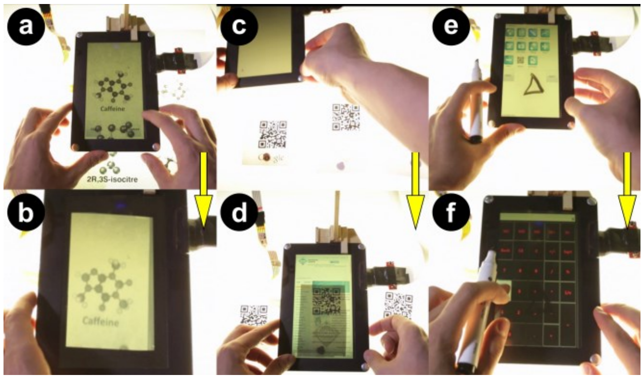

Figure 5. Surface capture interactions. Grabbing (ab) to get a picture. Markers (cd) and Scribbles (ef) are read implicitly.

Scribbles refers to recognizing jotted down pen gestures on an external surface (e.g. a paper sheet) as commands to the device. Similar to markers, the device constantly monitors for the presence of scribbles on the underlying surface. *Scribbles* can be used as input to the current application or to launch a new program. For example, Figure 5ef shows a triangle scribble use to invoke the Calculator application.

Model-based Interactions

Model-based interactions allow users to operate on digital content that is spatially aligned with the object below the device. Here we propose *Contact Augmented Reality* (cAR) as a special case where a handheld device with an optical see-through display rests directly on top of the object it augments, making cAR's interaction experience different from traditional AR. cAR depends not only on continuous surface capture as presented in the section before, but also on 1) a digital model of the underlying object, and 2) the continuous registration of the device (location and orientation) in relation to this object. Device registration allows superimposing digital content from the model on the physical object. Moving the device requires recalculating its location in the virtual model. Model-based interactions have been widely covered in AR [3, 5]. We limit our analysis to cAR interactions mentioned in our elicitation study. We present *extraction*, *annotation* and *area triggers*.

Extraction refers to interacting with elements of the digital augmentation of a physical object. For example, a cAR application for a magazine allows users to select words and look up definitions, translations, and other occurrences of these words in the document. Figure 6ab shows the results of a word search. Blue marks signal instances of the word, and the arrows point to other instances on the current page.

Annotation means attaching digital content to parts of the real object as seen through a cAR device. For example, Figure 6cd shows hand-written notes that stay anchored to the particular location where they were created.

Area triggers refers to special zones in the model that activate specific responses by the device when placed on top. For example, Figure 6ef shows a document image that triggers a video when the tPad is placed on top of it.

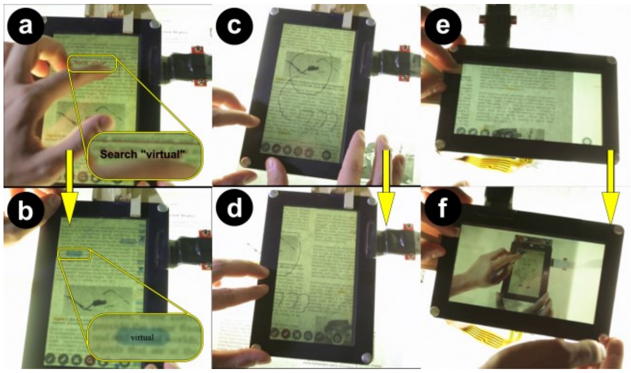

Figure 6. Model interactions. Extraction (ab) allows word search, Annotations (cd) are anchored to their location, and Area Triggers (ef) launch content automatically.

Transparent Mobile Interactions & Other Techniques

The proposed interactions are representative of operations possible on transparent display mobiles, some of which were investigated by others in different devices and form factors (see Table 1). Nonetheless, to the best of our knowledge, *querying*, *tap'n flip*[1] and *grabbing* are interactions unique to transparent-display mobile devices.

Interaction	Alternative Technique
Tracing	Vector graphics software in mobiles/PC.
Querying	Novel
Flipping	[31], Codex [14], Pens [21], DoubleFlip [27]
Tap'n Flip	Novel
Grabbing	Novel
Markers	Mobile AR applications [3].
Scribbles	Digital [14] and paper [28] pen interfaces.
Extraction	Mobile applications such as PACER [22].
Annotation	Document readers [14], projector UI [28,10].
Area Triggers	Pen interfaces [23] and marker-less AR.

Table 1. tPad interactions and other techniques.

On the other hand, transparent mobiles present affordances, such as transparency and tangibility, which alter the nature of those previously studied interactions. The transparency affordance allows *tracing*, *markers* and *annotations* to be carried out "on top" of the physical object and not through a virtual representation of it (i.e. 2D/3D model or camera capture). Working on the physical object allows users to perceive physical characteristics such as texture, density, age and wear, or material modifications not present on the model (e.g. handwritten notes on the object itself). Such physical contact with the objects enriches the interaction and learning experience. For example, a transparent mobile AR application for paper documents can support direct touch selections rather than, for example, PACER's "camera pointer". The same application is used while the device is lying on the paper, reducing the physical effort and fatigue of the interaction. Similarly, the tangibility affordance alters *tracing* and *markers*. When tracing, tangibility allows users to accommodate either the physical object or the transparent mobile for a better interaction. Reading *markers* becomes the implicit interaction of placing the device on top of the marker.

HARDWARE AND SOFTWARE

Based on the proposed interactions, an actual transparent-display mobile device requires the following capabilities:

Transparency – the capacity to see both the digital content and the world behind it. The display material plays a pivotal role on transparency and color perception [32].
Dual-sidedness – the capacity to see and interact with the display from either side. The challenge is to determine the active side and to classify touch as *front* or *back* input.

Surface capture – the capacity to image capture objects beneath the display. Not currently implemented in mobile displays, possible implementations exist [1, 34] and are available for large displays (e.g. PixelSense).

In an iterative design process, we built two hardware prototypes, called tPads, each with a subset of the technical capabilities. Our goal is to show that simple transparent-display mobiles enable novel and unique interactions. This section details our hardware and software implementations.

tPad-D – Orientation and Side Detection

The tPad-D (dual-side interaction), shown in Figure 1-Left, is made of a semi-transparent 7" LCD display, a resistive touch sensor on each side, and a board with an Arduino Pro Micro (5V/16MHz) controller, motion sensors (ADXL-335 triple axis accelerator), multiplexers (Quad SPDT Switch) and a push button. Figure 7-Bottom shows the sensor board schematics. A computer provides all computational and graphics processing needs. The Arduino controller communicates with the computer via serial. Both touch sensors are attached to the multiplexers. The drain pins of the multiplexers connect to the computer for touch processing. tPad-D implements the *transparency* capability, allowing users to see both digital content and real-world objects behind it. tPad-D implements the *dual-sidedness* capability by detecting the interaction side and adjusting the display orientation and active touch sensor.

The Arduino processes the accelerometer data to determine the interaction side (front-up or front-down) at 100 FPS. Figure 7-TopLeft shows the orientation vectors when holding the device at different angles with the side of interaction facing up. Similarly, Figure 7-TopRight shows the orientation vectors for the reverse side. Note the inverse orientation of values for the Z component (black trace). We used the Z component to determine the side, with a smoothing filter of 50 frames to reduce false side detections.

Once a side is detected the Arduino board communicates it to the computer and signals the multiplexers. When the device is in the front-down position, the computer flips the

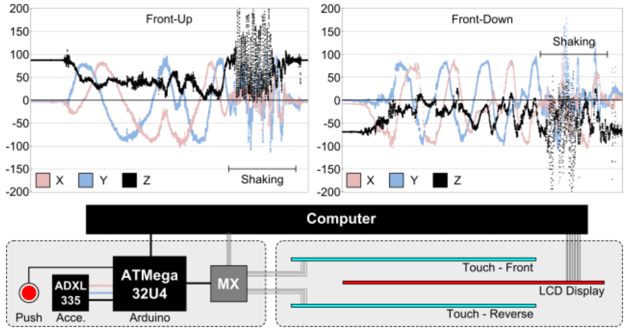

Figure 7. Top: Angle readings (Z in black) for front-up (left) and front-down (right). Bottom: board schematics.

[1] Hinckley et al. [15] combined *touch and tilt*. While related, *tap'n flip* requires dual-display and a complete flip.

graphics horizontally (appearing correctly to the viewer), and enables the bottom and disables the top touch sensors.

tPad-C - Surface Capture Emulation

The tPad-C (surface capture), shown in Figure 1-Right, is made of a semi-transparent LCD display, a resistive touch sensor, and an overhead camera looking down and through the display. A computer processes input and renders the interface. The tPad-C implements *surface capture* by mapping the camera image and display coordinates.

Being an overhead camera, it captures both the display contents and the underlying object. White display content renders the display transparent and allows capturing only the underlying object. Given that the camera position is at an angle with the display, we use a calibration process to determine warping and offset parameters. Figure 8 shows the camera capture and resulting warped image. We capture and transform the image at 30 FPS. Note that an ideal surface capture technology would use pixel capture such as the one proposed in [34] or Microsoft PixelSense. Although limited, this solution allows us to explore surface capture interactions and to gather user feedback.

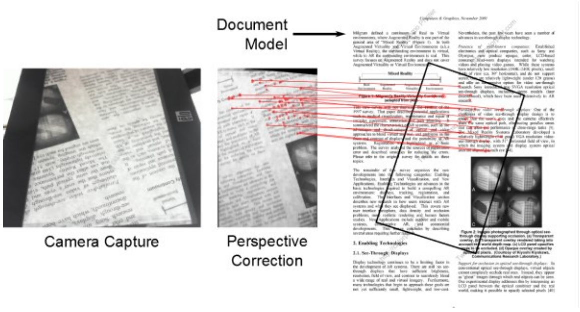

Figure 8. Registration process, red lines show matches.

Software

We use C# and Microsoft WPF for authoring and rendering and C++ and OpenCV 2.4.3 for image processing (warping, matching and scribble detection). The ZXing.NET library decodes markers, and the TallComponents' PDFKit.NET library provides getting pixel-level location information of the electronic documents for model-based interactions.

The registration process (location of the device on a paper document) uses surface capture and feature-matching. The registration algorithm processes the capture against known documents, and identifies the document and location (page, x-y coordinates, and rotation) by matching features from the capture image against document features (red lines in Figure 8). The features, also known as keypoints, efficiently describe image patches and are invariant to rotation, noise and scale. We detect significant keypoints using the FAST corner detector [26], and use the Fast Retina Keypoint (FREAK) descriptor [2] to perform the search.

USER EVALUATION

We assessed the impact of two mobile transparent interactions in everyday tasks and collected user feedback on their perceived value. 12 volunteers (4 female, avg. age 25 years) participated in two back-to-back experiments. All participants are smartphone users and none had previous experience with transparent displays.

Multitasking Experiment

On mobile devices, nearly 30% of tasks involve multiple applications [6, 11] with severe switching costs [20]. For this reason we investigate the use of tPad interactions on such a common and costly task. Current devices support multitasking via Apple's iPhone multitask bar and its equivalent in Android and Windows devices. In this experiment we evaluated the performance of *flipping* and *tap'n flip* on an information seeking task involving multiple apps. *Flipping* allows two applications to run on different sides, where one of them is the main application and the other the information source. *Tap'n flip* simplifies data transfer between apps by copy/pasting the tapped contents onto the other side.

Task – The experiment application asks users to 'collect' a number from another app (Figure 9a). Users navigate to the target application, collect the number, and navigate back to the experiment application. Figure 9 shows the process when the requested information is in the Red7 app. Information sources are organized by the distance from the main application, *blue* applications are on the same screen, while *red* applications are three screens away. A task consists in finding 3 numbers. Participants used the tPad-D prototype which supports application switching via the home button and flipping gestures.

Design – Independent variables were switching *method* and *number of applications*. Application distance was a random factor. We considered four switching methods: *home* (H – a button push shows the main screen), *multitasking bar* (MB – a button push shows the main screen and the recently used apps at the bottom – Figure 9b), *flipping* (F) and *tap'n flip* (TF). The number of apps varied from 1 to 3. Distance was random between 0 and 3. Participants trained with each condition after the experimenter had demoed the task. With a total of 4×3 = 12 conditions and 6 trials per condition, we registered 4×3×6 = 72 trials (each trial consisted of 3 selections) or 216 selections per participant. We used a Latin-square design to counter-balance the conditions. The experiment lasted approximately 30 minutes.

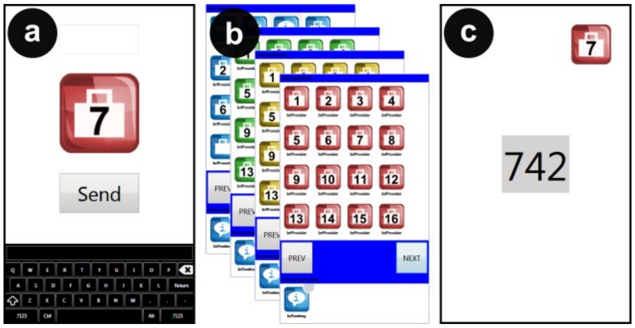

Figure 9. Information seeking experiment. A) Starting point. B) Finding the target application. C) Collecting the data.

Measures – We recorded time at different stages of the interaction (init, switch, found, end) and error rate. Users rated efficiency and enjoyment using a 5-point Likert scale, and ranked the *switching methods* according to preference.

Results

None of the measures complied with the assumptions for ANOVA and therefore we applied the Aligned Rank Transform for nonparametric factorial analysis with a Bonferroni correction for pairwise comparisons. We used Friedman's χ^2 and the Wilcoxon signed-rank tests to analyze ratings and rankings. Figure 10 shows the results.

Search Time – Results showed a main effect for *method* ($F_{3,33}$=20.44, p<0.001), *number of apps* ($F_{2,23}$=64.44, p<0.001) and *distance* ($F_{3,37}$= 24.20, p<0.001). Results showed interaction effects between *method* × *number of apps* ($F_{6,68}$=7.682, p<0.001), *method* × *distance* ($F_{9,123}$=2.77, p<0.05) and *number of apps* × *distance* ($F_{6,79}$=4.02, p<0.001). Post-hoc tests on *method* showed differences between all methods except between F and MB (p=0.163). Post-hoc tests on *number of apps* showed significant differences between all pairs. Post-hoc tests on *distance* showed significant differences between all pairs except between distances 2 and 3 (p=0.058). Users were fastest with TF at 6.6 sec (SD 3.6), F at 8.7 sec (SD 3.7), MB at 8.9 sec (SD 2.9) and H at 9.5 sec (SD 2.8).

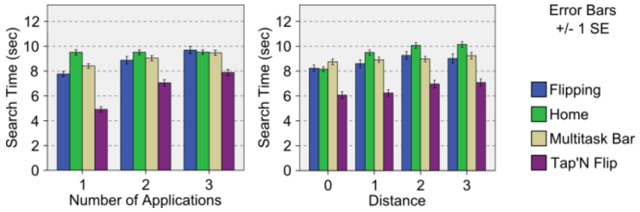

Figure 10. Information Seeking results.

Error Rate – Results showed a main effect for *method* ($F_{3,34}$=68.02, p<0.001), *number of apps* ($F_{2,24}$= 896.980, p<0.001), and *distance* ($F_{3,37}$=183.77, p<0.001). Results showed interaction effects between *method* × *number of apps* ($F_{6,76}$=206.89, p<0.001), *method* × *distance* ($F_{9,118}$= 62.54, p<0.001), *number of apps* × *distance* ($F_{6,81}$= 242.03, p<0.001), and *method* × *number of apps* × *distance* ($F_{18,197}$=50.39, p<0.001). Post-hoc tests on *method* showed significant differences between all pairs except between F and H (p=1.0). Post-hoc tests on *number of apps* showed significant differences between all pairs except between 1 and 3 (p=0.596). Post-hoc tests on *distance* showed differences between all pairs. Participants were more accurate with TF at 0% error (SD 0%), F at 0.7% (SD 8.3%), H at 0.8% (SD 10%), and MB at 1% (SD 9.8%).

Participant's Ratings – Results showed a significant difference between *methods* in perceived efficiency (χ^2(3)=13.757, p<0.005), enjoyment (χ^2(3)=13.294, p<0.005) and preference (χ^2(3)=13.900, p<0.005). Post-hoc tests on efficiency showed significant differences between TF and H (Z=-2.65, p<0.008), TF and F (Z=-3.02, p<0.003), and MB and H (Z=-2.5, p<0.011). Post-hoc tests on enjoyment showed significant differences between all pairs except between F and MB (Z=-.49, p=0.618) and F and H (Z=-1.81, p=0.07). Post-hoc tests on preference showed significant differences between TF and H (Z=-2.71, p<0.007) and TF and F (Z=-3.16, p<0.002). For all factors (efficiency, enjoyment, preference) users rated *switching methods* in the same order (best first): TF, MB, F and H.

Interaction Stages – A temporal analysis of the stages (start, search and return) of the switching methods sheds light on how the techniques differ. As each trial involved a random number of apps, users find themselves in one of three states after switching: *onhome*, *ontarget*, or *onother*. The *onhome* state is when no target app was launched before (first selection of the trial). The *ontarget* state is when the current and previous targets are the same. The *onother* state is when the current and previous targets differ. Figure 11 shows the stages for each switching method and state.

The temporal models show all switching methods have a similar start time of around 1.5secs, which indicates that performance differences arise at the search and return stages. In all but the flipping methods on the *ontarget* state, the search stage takes 2-3 seconds. Searching tends to take slightly longer time for the flipping methods (*onhome* and *onother* states), which could be attributed to the poor ergonomics of our device requiring users to adjust their grip before navigating to the target app. For most switching methods, except *tap'n flip*, the return stage takes the most time. This is explained by having to copy the content, to navigate back to the experiment app, and to paste it. *Flipping* returns show a small advantage over *home* and *multitask* (~1sec) due to the faster navigation (a back flip), yet the time for copying and pasting is significantly large.

The positive results of *tap'n flip* can be attributed to the time gains in the search and return stages. The short return time is due to the merging of the copying, navigating, and pasting actions into a single flipping gesture. Performance gains are particularly high when multitasking involves only one other app, as the flip results on the target app and no further

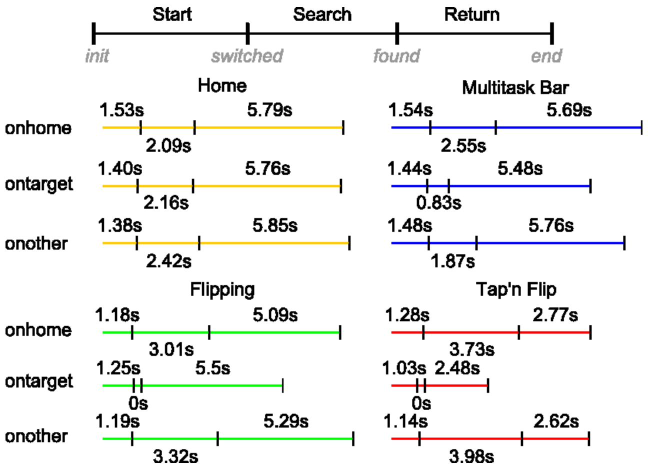

Figure 11. Temporal models of the switching methods.

navigation is required (see the zero search time in Figure 11). Similar return times in *flipping*, *home* and *multitask bar*, and their difference with *tap'n flip* reveal that the determinant performance factor is not the flipping gesture, but the *tap* modifier.

Discussion

Results show that *flipping* is as fast as the *multitasking bar* and generally faster than *home*. This difference persists as the number of multitasking applications increases and when the application distance is large. Results also showed that *tap'n flip* is faster than all other switching methods and, in general, 36% faster than *home*. This difference is increased to almost 50% when multitasking to only one other app. Moreover, *tap'n flip* had a marked 0% error rate, and *flipping* induced less errors than the *multitask bar*. Users also preferred *tap'n flip*, ranking it highest in all aspects and perceiving it as offering a pleasant user experience. This *tap'n flip* preference might be explained by its low cognitive demands allowing users to stay focused on the tasks rather than in the interaction. Users perceived *flipping* as equally good as the *multitask bar*, suggesting it alone might not make a difference for users. This can be due to do the long return stage and having to manually copy+paste the content. The preference and higher performance of *flipping* and *tap'n flip* surfaced even with our 7 inch tPad (mid-size) which made the gestures cumbersome. Smaller devices could optimize these interactions, particularly in the search stage (fixing the grip). Finally, our results show that better copy+paste mechanisms could positively impact all switching methods including *home* and *multitask bar*.

Image Capture Experiment

Another important everyday use of mobile devices is to capture information by taking a picture [17]. Providing a faster way to capture this information increases how frequently people could do it. A step in this direction, already available in commercial devices, is mechanisms to quickly access the camera of a mobile device without even unlocking it. We focus on the situation where the information is contained on an object smaller than the handheld device. In this experiment we evaluate the impact of taking a picture by means of the *grabbing* interaction. *Grabbing* allows for taking the picture of an underlying object by simply placing the device on top of it.

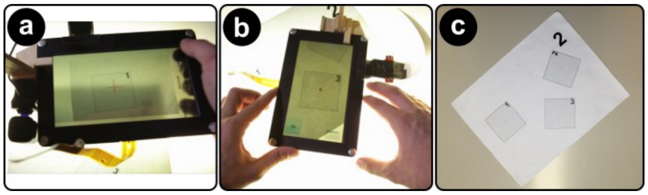

Figure 12. Information capture experiment. a) Normal picture taking. b) Grabbing by means of surface capture. c) Half-size targets on the paper sheet used.

Task – The experiment asked participants to take a picture of one of three squares printed on a paper (see Figure 12c). All squares in the paper had the same size but different orientation. We included three square sizes, meaning there were three paper documents on the table. Participants used tPad-C equipped with an extra camera configured to work as a traditional mobile device camera. The tPad laid on the table next to the document before each trial. We added a sub-task where users cropped out the target from the image.

Design – Independent variables were device, capture method, and target size. We considered two devices: *tPad* (grabbing interaction, Figure 12b) and *normal* (as in current mobile devices, Figure 12a). Capture method refers to cropping or not the picture. Three target sizes were considered: a *quarter*, *half*, and *three quarters* of the 7'' display size. Participants were trained with each condition after the experimenter demonstrated the task. With a total of 2×2×3=12 conditions and 9 trials per condition, we registered 2×2×3×9=108 trials per user. The experiment lasted approx. 30 minutes. The conditions were counter-balanced using a Latin-square design.

Measures – We collected the time to capture the target and the offset in angle and distance from the center. Users rated perceived efficiency and enjoyment using a 5-point Likert scale, and ranked the devices according to preference.

Results

Data did not comply with the ANOVA assumptions and therefore we used the same analytical tools as in the first experiment. Figure 13 shows the results.

Capture Time – Results showed a main effect for *device* ($F_{1,10}$=69.60, p<0.001) and *method* ($F_{1,12}$=149.34, p<0.001) but not for *target size* (p=0.275). Results showed interaction effects between *device* × *method* ($F_{1,10}$=23.75, p<0.001). Participants were fastest with the grabbing method at 7.8 sec (SD 5.7) and the normal method at 12.3 sec (SD 7).

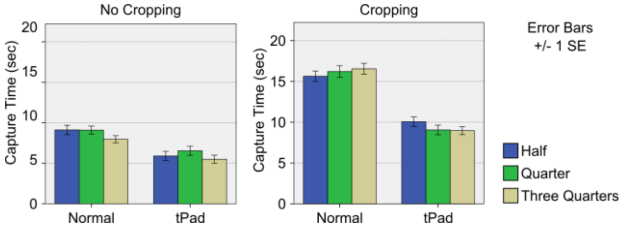

Figure 13. Information Capture results.

Distance Offset – Results did not show a main effect for *device* (p=0.056) or *method* (p=0.630) but they did for *target size* ($F_{2,23}$=10.10, p<0.001). Results showed interaction effects between *device* × *target size* ($F_{2,20}$=6.46, p<0.01). Post-hoc tests on *target size* showed differences between all pairs except between half and quarter (p=0.751). Participants were more accurate on the bigger size with an offset of 1.85 mm (SD 3), with half and at 2.26 mm (SD 3.32) and quarter at 2.49 mm (SD 3.46).

Angular Offset – Results showed a main effect for *device* ($F_{1,10}$=6.75, p<0.05) and *target size* ($F_{2,24}$=16.32, p<0.01) but not for *method* (p=0.155). Results did not show any significant interaction effects. Post-hoc tests on *target size* showed a significant difference between all pairs except between half and three quarters (p=1.000). Participants' captures were straighter with the tPad *device* at 1.33° (SD 1.4°) than with a normal device at 1.7° (SD 2°).

Participant's Ratings – Results showed a significant difference between *devices* in efficiency (Z=-2.71, p<0.007), enjoyment (Z=-2.20, p<0.027) and preference (Z=-2.27, p<0.023). For all factors (efficiency, enjoyment and preference) users rated the tPad device highest.

Discussion

Results suggest that, in general, the *grabbing* interaction is twice as fast as traditional picture taking. Grabbing also resulted in better-aligned images, which facilitates their reading later on. Faster performance can be attributed to the time saved by resting the device on the table. With current devices (normal photo-taking), users must lift the device, aim the camera, adjust focus, and correct for involuntary movement (hand trembling). With tPad, users simply slide the device over to the capture target without lifting it from the table, adjusting for focus or involuntary movement. Interestingly, results show that with tPad, acquiring and cropping the image takes roughly the same time than taking a picture with current devices (10 sec). This suggests that tPad users can have better aligned and cropped-out images without sacrificing efficiency.

Besides the higher performance, *grabbing* offers other benefits: the physical effort to take a photo is minimized as the device rests on the surface. Also, tPad *grabbing* was often single handed, while normal picture-taking usually required both hands. The higher performance and simpler user experience are reflected in the users' preference for tPad *grabbing* interactions.

GENERAL DISCUSSION

We presented a list of interaction techniques for transparent-display mobiles that break apart from the commonly discussed AR scenarios. While some of the techniques, or similar ones, have been studied in other devices, we showed how they differ when used in transparent-display mobiles. Nonetheless, our research shows techniques that are unique to transparent-display mobiles. Certainly, other approaches exist to interaction design; however this paper highlights how hands-on user involvement can serve as a starting point to be followed by analysis and synthesis as shown in our capability-based interaction categories.

The proposed categories reflect the scalable nature of our contribution, highlighting the value of transparent mobiles at different levels of sophistication (from simply transparent to surface capture displays). Simple devices with only a transparent display (*overlay* category) provide more than an aesthetic appeal as shown with the *tracing* and *querying* interactions of the overlay category. For example, the transparent mobile can serve as a ruler. In analytical settings, a graph explorer application enables information extraction from physical documents. Such usages can be readily implemented in the devices already commercially available and have not been previously showcased.

We also demonstrate the benefit of tPad interactions for two common mobile tasks, application switching and image capture. However, both interactions require advanced display capabilities beyond simple transparency. Dual-side display & input is a simple augmentation already present in industrial prototypes [33]. However, more elaborated is surface capture and the interactions it supports which, with our camera approach, we explored without focusing on low-level challenges. Other promising advanced capabilities are possible such as, for example, head-tracking to determine the actual alignment of the tPad with the user's field-of-view and the real world (for AR applications).

Technical Limitations

Even though hand-held and mobile, our tPads require external illumination for the LCD display. Our camera approach for surface capture is limited in its ergonomics, and it is also affected by the current content on the display, the display opacity and user hands or stylus. These factors reduce the quality of the image and affect the success rate of the registration algorithm. Our orientation detection takes gravity as the main indicator, and thus situations where the device is above the user's eyes cannot be detected. Our model-based interactions are limited to text documents for which we have the original PDF. Finally, our prototypes did not manage the color blending challenge of transparent displays which compromises legibility, particularly of text.

CONCLUSION

In this paper we demonstrated that transparent-display mobile devices facilitate novel interaction techniques, many of which are not easily possible on existing mobile devices. Through a user-centered design process we classified interactions in four categories: *overlay*, *dual display & input*, *surface capture* and *model-based interactions*. Each category is based on specific technical capabilities of the transparent display, with *overlay* needing simple transparency and *model-based interactions* requiring semantic knowledge of the overlaid objects. We implemented two transparent-display tablet prototypes, and validated a subset of our proposed techniques for everyday mobile tasks: multitasking and information capture. Our results showed that transparent-display mobile interactions outperform the non-transparent alternatives and users prefer them.

ACKNOWLEDGEMENTS

This work was funded by the NSERC Strategic Project Grant awarded to Pourang Irani on see-through displays, the German Research Foundation (DFG) through the projects IPAR (DA 1319/2-1) and GEMS (DA 1319/3-1), and the LEIF Transatlantic Exchange Partnership project.

REFERENCES

1. Abileah, A., Den Boer, W., and Green, P. 2010. Light sensitive display. U.S. Patent No. 7,830,461. 9 Nov. 2010.

2. Alahi, A. and Ortiz, R. and Vandergheynst, P. 2012. FREAK: Fast Retina Keypoint. In Proc. CVPR '12. IEEE.

3. Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., and MacIntyre, B. 2001. Recent Advances in Augmented Reality. Computer Graphics. 21, 6. IEEE.

4. Bier, E. A., Stone, M. C., Pier, K., Buxton, W., and DeRose, T. D. 1993. Toolglass and magic lenses: the see-through interface. In Proc. SIGGRAPH '93.

5. Bimber, O. and Raskar, R. 2005. Spatial Augmented Reality: Merging Real and Virtual Worlds. A. K. Peters, Ltd., Natick, MA, USA.

6. Böhmer, M., Hecht, B., Schöning, J., Krüger, A. and Bauer, G. 2011. Falling asleep with Angry Birds, Facebook and Kindle – A large scale study on mobile application usage. In Proc. MobileHCI '11, ACM.

7. Concept Phones. (2013, March 17). Transparent Phones. [Website]. Retrieved from http://www.concept-phones.com/tag/transparent-phone/.

8. Concept Vending Machine With See-through Display (2013, September 13). Youtube [Website]. Retrieved from http://www.youtube.com/watch?v=8pPCIPHTe3M.

9. Corning Incorporated. (2013, March 17). A Day Made of Glass 2: Unpacked. The Story Behind Corning's Vision. (2012) [Website]. Retrieved from http://www.youtube.com/watch?v=X-GXO_urMow.

10. Dachselt, R. and Al-Saiegh, S. 2011. Interacting with printed books using digital pens and smart mobile projection. In Proc. of the Workshop on Mobile and Personal Projection (MP2) @ ACM CHI '11.

11. Dunlop, M. and Brewster, S. 2002. The Challenge of Mobile Devices for Human Computer Interaction. Personal Ubiquitous Computing. 6, 4.

12. Gabbard, J.L., Swan, J.E., Zedlitz, J., and Winchester, W.W. 2010. More than meets the eye: An engineering study to empirically examine the blending of real and virtual color spaces. In Proc. VR '10. IEEE.

13. Hilliges, O., Kim, D., Izadi, S., Weiss, M., and Wilson, A. 2012. HoloDesk: direct 3d interactions with a situated see-through display. In Proc CHI '12. ACM.

14. Hinckley, K., Dixon, M., Sarin, R., Guimbretiere, F., and Balakrishnan, B. 2009. Codex: a dual screen tablet computer. In Proc. CHI '09. ACM.

15. Hinckley, K., and Song, H. 2011. Sensor synaesthesia: touch in motion, and motion in touch. In Proc. CHI '11. ACM.

16. Kim, K., and Elmqvist, N. 2012. Embodied lenses for collaborative visual queries on tabletop displays. Information Visualization, 11(4), 319-338.

17. Kindberg, T., Spasojevic, M., Fleck, R., and Sellen, A. 2005. The Ubiquitous Camera: An In-Depth Study of Camera Phone Use. IEEE Pervasive Computing.

18. Lee, J., Olwal, A., Ishii, H., and Boulanger, C. 2013. SpaceTop: integrating 2D and spatial 3D interactions in a see-through desktop environment. In Proc CHI '13. ACM.

19. Lee, J.H., Bae, S.H., Jung, J., and Choi, H. 2012. Transparent display interaction without binocular parallax. In Adjunct Proc. UIST '12. ACM.

20. Levia, L.A., Böhmer, M., Gehring, S. and Krüger, A. 2012. Back to the app: The costs of mobile application interrruptions. In Proc. MobileHCI '12, ACM.

21. Li, Y., Hinckley, K., Guan, Z., and Landay, J. A. 2005. Experimental analysis of mode switching techniques in pen-based user interfaces. In Proc. CHI '05. ACM.

22.Liao, C., Liu, Q., Liew, B., and Wilcox, L. 2010. Pacer: fine-grained interactive paper via camera-touch hybrid gestures on a cell phone. In Proc. CHI '10. ACM.

23. Mackay, W.E., Pothier, G., Letondal, C., Bøegh, K., and Sørensen, H.E. 2002. The missing link: augmenting biology laboratory notebooks. In Proc. UIST '02. ACM.

24. Ohtani, T., Hashida, T., Kakehi, Y., and Naemura, T. 2011. Comparison of front touch and back touch while using transparent double-sided touch display. In ACM SIGGRAPH 2011 Posters (SIGGRAPH '11). ACM.

25. Olwal, A., Lindfors, C., Gustafsson, J., Kjellberg, T., and Mattsson, L. 2005. ASTOR: An autostereoscopic optical see-thru augmented reality system. In Proc. ISMAR '05. IEEE.

26. Rosten, E. and Drummond, T. 2006. Machine learning for high-speed corner detection. In Proc. ECCV '06. Springer Berlin Heidelberg.

27.Ruiz, J., and Li, Y. 2011. DoubleFlip: a motion gesture delimiter for mobile interaction. In Proc. CHI '11. ACM.

28. Schilit, B. N., Golovchinsky, G., and Price, M. N. 1998. Beyond paper: supporting active reading with free form digital ink annotations. In Proc. CHI '98. ACM.

29. Schmalstieg, D., Encarnação, L. M., and Szalavári, Z. 1999. Using transparent props for interaction with the virtual table. In Proc. I3D '99. ACM.

30. Sharma, A., Liu, L., and Maes, P. 2013. Glassified: An Augmented Ruler based on a Transparent Display for Real-time Interactions with Paper. In Proc. UIST '13 Demos.

31. Spindler, M., Tominski, C., Schumann, H., and Dachselt, R. 2010. Tangible views for information visualization. In Proc. ITS '10. ACM.

32. Sridharan, S.K., Hincapié-Ramos, J.D., Flatla, D., and Irani, P. 2013. Color Correction for Optical See-Through Displays Using Display Color Profiles. In Proc. VRST '13. ACM.

33. Transparent dual-touch display Android smartphone prototype #DigInfo. (2012). [Website]. Retrieved from http://www. youtube.com/watch?v=Xbyoool5ZwY.

34. Uy, M. 2009. Integrated sensing display. U.S. Patent No. 7,535,468. 19 May 2009.

35. Wigdor, D., Forlines, C., Baudisch, P., Barnwell, J., and Shen, C. 2007. Lucid touch: a see-through mobile device. In Proc. UIST '07. ACM.