tPad: Rich Interaction with Transparent-Display Mobiles

1st Author Name
Affiliation
Address
e-mail address

2nd Author Name
Affiliation
Address
e-mail address

3rd Author Name
Affiliation
Address
e-mail address

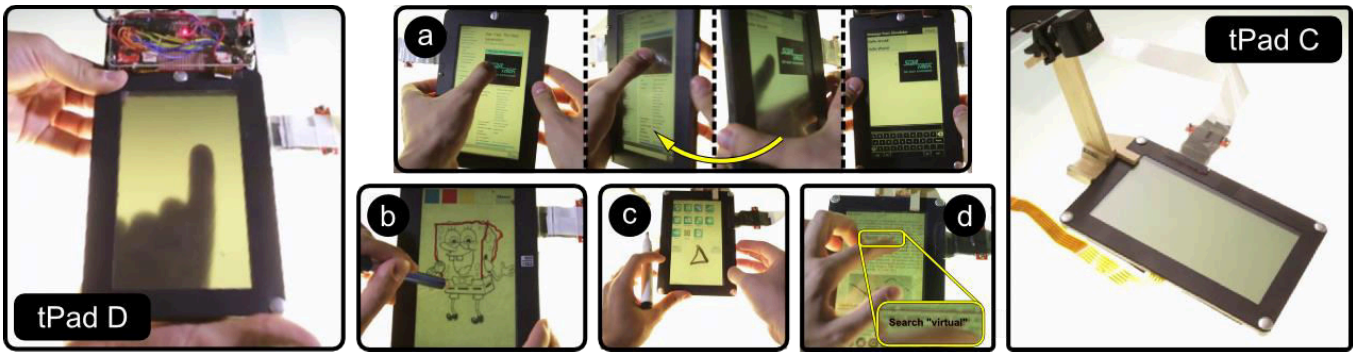

Figure 1. Left: tPad D, a transparent-display mobile with touch and movement sensors enables interactions such as *flipping* (a) and *tracing* (b). Right: tPad C, simulates surface capture via an overhead camera, enabling *scribbles* (c) and document *search* (d).

ABSTRACT

As a novel class of mobile devices with rich interaction capabilities we introduce tPads – tablets with transparent displays. Whereas concept studies and first commercial devices outline their potential especially for augmented reality applications, a systematic investigation of ways and benefits of interacting with transparent mobiles is still missing. As a result of a user-centered design process, this paper explores the unique interaction opportunities of transparent tablets and classifies them into the four categories: *overlay*, *dual display & input*, *surface capture* and *model-based interactions*. During an iterative development process, we built two touch-enabled, semi-transparent 7” LCD tablet prototypes we call tPads. A range of implemented applications deploy the proposed interactions and show the value of transparent mobiles beyond augmented reality. We also conducted a user study evaluating two common tasks on mobile devices: application switching and image capturing. Results show that tPads allow more efficient performance and are preferred by users over the non-transparent alternatives.

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 [17]. These advances have resulted in a flurry of novel initiatives showcasing interactions not possible on traditional mobile devices [8, 10]. With the exception of a relatively small number of research prototypes that exploit the transparency factor to address known mobile interaction limitations [25, 20, 40], most conceptual depictions focus on mobile augmented reality applications [8, 10].

Unfortunately, such demonstrations of transparent-display mobile devices still leave unanswered the important question of how beneficial this technology is for common tasks on mobile devices. Even further, experts question whether transparent displays have a place in non-niche applications [13]. For the adoption of such a technology, it is therefore important to identify a broader range of applications and the potential benefits of transparent-display mobile devices for common tasks. Ideally, such benefits outweigh the primary limitations of transparent displays such as screen resolution [13], color blending [16] and binocular parallax [25].

Our primary goal was to explore how mobile devices can benefit from the adoption of transparent displays. We followed a user-centered design process aimed at finding mobile interaction techniques that exploit display transparency in everyday tasks. The process focused on use cases with which participants are familiar (e.g. reading email, instant-messaging, browsing the web, or reading printed papers) in order to elicit concrete design choices.

Four major categories of interaction techniques emerged from this process, each of which depends on specific hardware capabilities. The most basic category, *overlay*, includes techniques that only depend on the display being transparent. One step further, the *dual display & input* category includes techniques that build upon the capacity for seeing and interacting with the device from both sides. Previously proposed techniques [28, 40] exploit the dual-input properties from this category. The *surface capture* category depends on the device's capacity to image capture real-world objects below the display. Finally, the *model-based interaction* category depends on matching the image of the object below the display with a pre-existing virtual model of the object. Augmenting a paper document to support active reading [2, 11, 27, 32, 41] is a prime example of model-based interactions.

We extend our exploration beyond this conceptual framing and built two transparent tablet prototypes or *tPads*. Each tPad possesses capabilities to support the above-mentioned categories. 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. We also implemented the *surface capture* and *model-based* categories by means of an overhead camera looking down and through the display, as in tPad-C (Figure 1-Right). For each prototype, we implemented a broad range of applications using the proposed interactions and showing the value of transparent-displays for common mobile tasks.

In a user study we investigated the benefits of tPads in two common tasks with mobiles: application switching [7, 26] and image acquisition [22]. Results show that tPad interactions, based on the *dual display & input* and *surface capture* capabilities, allow more efficient performance and are preferred by users over the non-transparent alternatives.

Our contributions include: (a) a categorical organization of novel transparent-display mobile interactions; (b) the implementation of two transparent-display prototypes, tPads, to demonstrate the value of the various interaction techniques; and (c) a user study demonstrating that two common mobile tasks, switching between applications and capturing information from the physical world, can be facilitated by means of a transparent-display mobile device.

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 with transparent displays has focused to a large extent on the window-size format. These devices are made possible by the use of projectors and diffusive films, and have been applied within the context of spatial augmented-reality (SAR). *SAR* systems require a display fixed in space, providing *spatial alignment* which facilitates overlaying digital content on physical objects. This also limits SAR applications to fixed settings [6], such as for use on industrial machinery [29] and vending machines [9]. More recently, researchers explored immersive experiences like the HoloDesk [18] and SpaceTop [24], where users can directly interact with virtual objects by moving their hands in the space behind a transparent display.

In this paper we focus on scenarios where the transparent display is mobile, i.e. non-fixed. With a few mobile devices becoming available to the public [17] and electronic companies announcing the mass production of such displays (e.g. Futaba Corp [15] and Fujitsu [14]) it is important to investigate the broad range of interaction techniques made possible through such a form factor.

To our best knowledge only a few projects explored this direction. Lee et al. [25] 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 [38]). LucidTouch [40] 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 [20, 28] 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. Glassified [34] embedded a small transparent display into a ruler and explores the augmentation of 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, smartphones and handheld projectors, etc.), allowing the augmentation of virtually any object within the display's field-of-view but requiring complex operations for registration (e.g. 3D location, object recognition) and rendering (e.g. field-of-view calculation, perspective correction). Moreover, mobile displays present limitations in terms of resolution, focus, lighting and comfort. A complete reference to AR can be found here [4, 6].

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 the tool-glass and magic lenses [5], tangible views [35, 21], and aspects of mobile AR [4].

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 [20].

Tangible views provide complementary displays for content visualized on tabletop computers. Spindler et al. [35] used spatially tracked, handheld lenses made of cardboard and propose an interaction vocabulary including: translation, rotation, freezing, gestures, direct pointing, toolbox metaphor, visual feedback and multiple views. Other researchers built such tangibles using transparent acrylic plates with fiducial markers [21] and 3D head tracking [33].

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 a transparent-display mobile device. The process included hands-on sessions, semi-structured interviews and a focus group. Six participants (two female) took part in the design process. Participants proposed interaction techniques using mock devices with transparent acrylic 'displays' and non-permanent markers to draw the expected user-interfaces (Figure 2). The mock device resembled a 10×15cm mid-size tablet to cover more cases than possible with a smaller device, yet remain highly mobile. To effectively ground the discussions and involve participants we drew inspiration from everyday activities (e.g. reading papers, browsing the web, etc.) and general-purpose applications (e.g. calculator, messaging, etc.). The researcher asked follow up questions aimed at refining the proposed interaction.

We focused on interaction techniques that would take advantage of the display being transparent and that are not possible with existing devices. Thus, we discarded mobile AR interactions, as they are possible today using video-based see-through displays. After the initial one-to-one sessions, we ran a focus group aimed at refining the individually gathered interactions. The focus group iterated over the interactions, refining them further and finding alternative application scenarios.

Figure 2. Hands-on user-centered design sessions for transparent display mobile interactions.

Results

Users proposed 69 potential usages and functionalities where they considered transparent-display mobiles could have an edge over existing devices. The proposed interactions range from application specific functionalities to device-wide interactions. For example, when working on paper documents users envisioned using a transparent-display mobile as an assistive magic lens to add virtual layers to the document; when multi-tasking on the mobile, users proposed flipping it in order to get a second display and have two applications running "simultaneously".

We classified the proposed usages and functionalities into higher level interactions and grouped them into the *overlay*, *dual display & input*, *surface capture* and *model-based interactions* categories. The following sections present the interactions in each category and demonstrate their implementation through different tPad prototypes.

TRANSPARENT MOBILE INTERACTIONS

We identified four major categories of interactions, each supported by specific transparent display capabilities.

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, or another mobile device) to create direct contact between the display and the object. The object becomes visible through the display. Overlay interactions require no information about the underlying 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 features of the object seen immediately beneath the display. For example, a user might trace items in a picture to create an artistic reproduction (Figure 3a). 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 3b). The tPad *Tracer* application enables basic tracing scenarios.

Querying refers to taking advantage of knowing the size of a pixel on the display (in real-world units) and its correspondence in the underlying object. For example, it

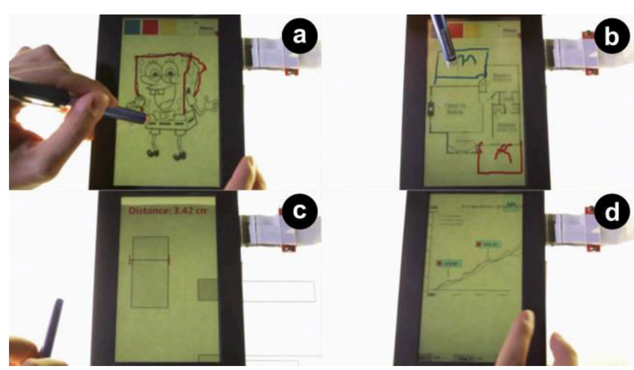

Figure 3. Overlay Interactions. Tracing (a, b). Querying with the Ruler (c) and the GraphExplorer (d).

can be known that a 30 pixel line covers 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 3c). Another example is the tPad Graph Explorer application which, after entering the right parameters, allows users to query values on a printed chart (Figure 3d); 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 interact with the display from either side both for display output and input. Given the existing research on back-of-device interactions [20, 38, 40] we focused on two interactions: *flipping* and *tap'n flip*.

Flipping refers to physically rotating the device around so as to see the display from the other side. Upon flipping, a tPad can offer alternative visuals or work as a secondary display. Unlike other dual-display devices [19], in a transparent-display mobile only one of the "displays" is usable at a given time. For example, applications can be attached to a specific side. Figure 4a-c shows a user 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 in which users often lose their working context. Upon seeing a notification alert while using one application, the user can flip the device to access the notifying application. When the task is completed the user flips back to the original side to continue the previous activity. tPad applications running on different sides can share a context object which allows them to share information at every flip. For example, when flipping from the *Ruler* to the *Calculator* application, the calculator receives a context string of the current ruler value (e.g. "7.4 cm") which is used for math operations.

Tap'n Flip allows a user to hold a particular user interface element while flipping sides to transfer that particular element from one side to the other. For example, Figure 4d-f shows a user *Tap'n Flipping* on one image in the web browser to attach it to a chat conversation.

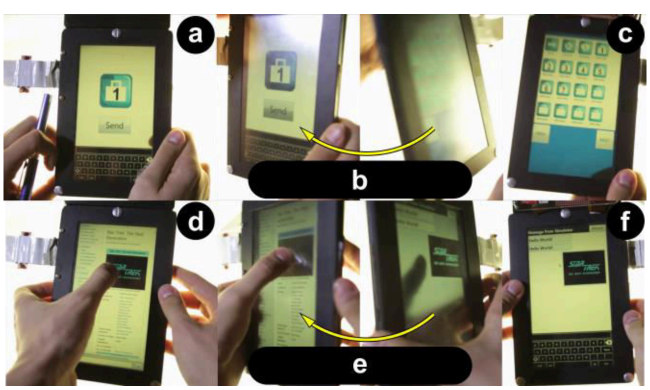

Figure 4. Dual display and input interactions. Flipping (a-c) to go to the home screen, and Tap'n Flip (d-f) to copy+paste.

Surface Capture Interactions

Surface Capture interactions allow users to interact with the device based on the image of the object right below the transparent display (see the *surface capture* capability outlined in the hardware section below). Three interaction possibilities are covered by this category: *grabbing*, *markers* and *scribbles*.

Grabbing refers to capturing an image of the whole or a cropped out area of the object beneath the transparent display (Figure 5ab). An application using grabbing can later post-process the image to extract relevant information about it. For example, the tPad *Capture* application allows users to specify a cropping region before executing 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 moving the device on top of a marker (e.g. QR code) for the device to read it. In contrast to traditional code readers, surface capture allows constant monitoring for markers, and turns reading markers into the simply act of placing the device on top of it (Figure 5cd). When a marker is found, the tPad *QReader* application is notified and handles the event according to the encoded value. When the marker encodes a URL, the QReader launches the Browser pointing to the desired page. A marker can also embed predefined sets of settings such as "*silence mode*" or *"set alarm at 6:15am"*.

Scribbles refers to recognizing jotted down pen gestures on an external surface as commands when the device is placed on top of them. Similar to markers, the device constantly monitors for the presence of scribbles on the underlying surface. *Scribbles* can be used to launch applications or to enter commands to the currently running application. For example, Figure 5ef shows a triangle scribble which allows a user 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

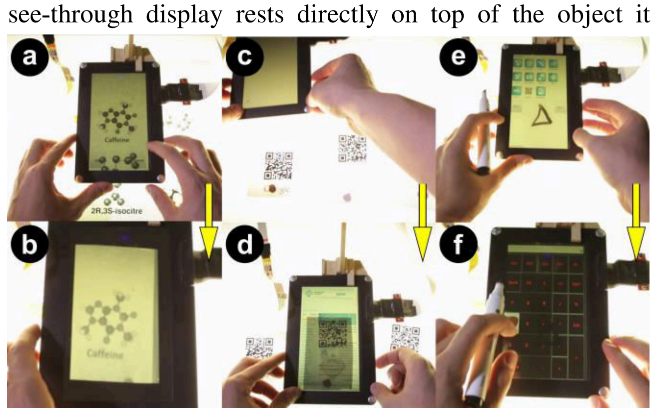

Figure 5. Surface capture interactions. Grabbing (ab) to get a picture. Markers (cd) and Scribbles (ef) are read implicitly.

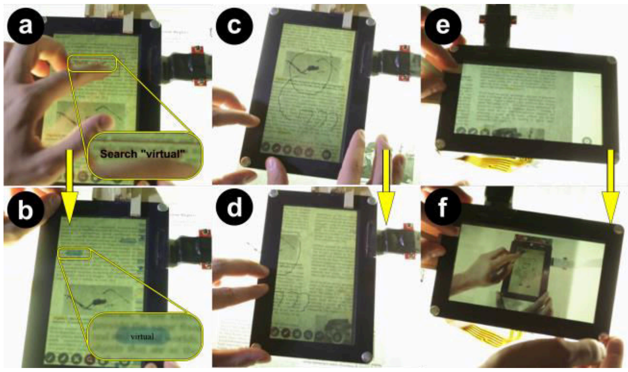

Figure 6. Model interactions. Extraction (ab) allows word search, Annotations (cd) are anchored to their location, and Area Triggers (ef) launch content automatically.

augments. 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 [4, 6]. 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 refers to 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.

HARDWARE AND SOFTWARE

Based on the proposed interactions, an actual device able to support them should rely on a transparent display with the following technical capabilities:

Transparency – the basic trait which allows users to see both the digital content and the world behind it. The display material plays a pivotal role on the amount of transparency and its impact on color perception [36].

Dual-sidedness – the capacity to see content and interact with the display from both sides. The challenge here is to determine the current side of interaction in order to show the user interface with the right orientation, and to correctly classify touch gestures as *front* or *back* of the device.

Surface capture – the capacity to image capture the object beneath the display. Not currently implemented in mobile displays, this capability is already part of commercial displays (e.g. Microsoft's PixelSense), and has been discussed in concept designs [40] and patents [1, 39].

We built two hardware prototypes, called tPads, each providing a subset of the desired technical capabilities. This section presents the hardware and software implementations. Our goal is to show that even simple transparent-display mobiles enable novel and unique interactions.

tPad-D – Orientation and Side Detection

The tPad-D (dual-side interaction), shown in Figure 1-Left, is made of a single semi-transparent 7" LCD display, a resistive touch sensor for each side, and an attached 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). A nearby computer provides all computational and graphics processing needs. The Arduino controller provides a serial communication channel with the computer. 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.

We used an Arduino Pro Micro at 5V/16MHz to process the accelerometer data and to determine the interaction side (front-up or front-down). We calibrated and transformed the accelerometer data to obtain the orientation vectors for each axis 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 for the last 50 frames to reduce false side detections induced by shaking or tapping. Figure 7-Bottom shows the sensor board schematics.

Once a side is detected the Arduino board communicates it to the computer and signals the multiplexers. When the

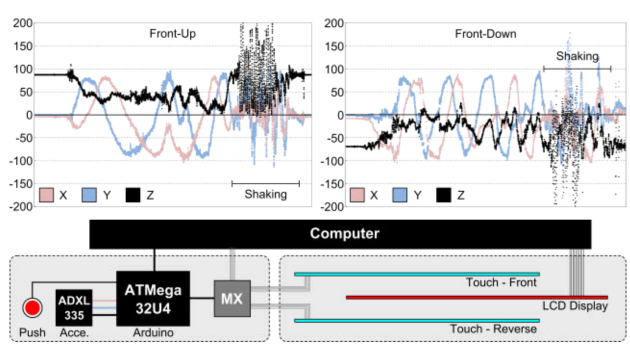

Figure 7. Top: Angle readings (Z in black) for front-up (left) and front-down (right). Bottom: board schematics.

device is in the front-down position the computer flips the 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 single semi-transparent LCD display, a resistive touch sensor, and an overhead camera looking down and through the display. The computer processes the touch and camera feeds and renders the display. The tPad-C implements *surface capture* by mapping the pixels of the camera image to those of the display.

In this set-up the camera captures both the display contents and the underlying object. When the display renders white content, the camera captures only the underlying object. Given that the camera position is at an angle with the display, we use a calibration process to determine the warping and offset parameters. Figure 9-Top 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 [10] or the one used by the Microsoft PixelSense tabletop. Although limited, this simplistic solution allows us to explore the interactions which such technology enables, and to gather user feedback.

Software

Figure 8 presents the software components used for all tPad implementations. A tPad is created by launching the ShellApp application (pink), which in turn instantiates the tPad Core (gray) and other tPad applications (yellow). The tPad Core is in charge of managing *monitors* (green) and *services* (blue). *Monitors* are interfaces to low level hardware components like the Arduino controller and the overhead camera. *Monitors* notify services about changes in the data they collect. *Services* receive notifications from monitors and expose their results to applications through event handlers and data objects.

We use C# and Microsoft WPF for authoring and rendering and C++ and OpenCV 2.4.3 for image processing and feature matching. Scribbles are recognized using standard OpenCV figure detection. Markers are decoded using the ZXing.NET library [42]. Model-based interactions use the TallComponents' PDFKit.NET [30] for getting pixel-level location information of the PDF contents.

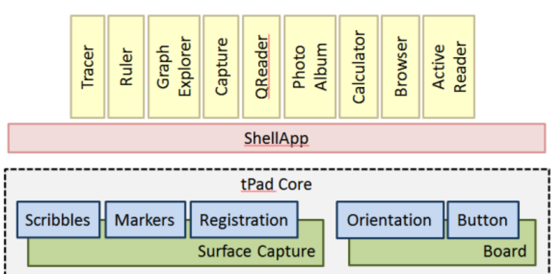

Figure 8. tPad applications (yellow), system shell (pink), services (blue) and monitors (green) – Best seen in color.

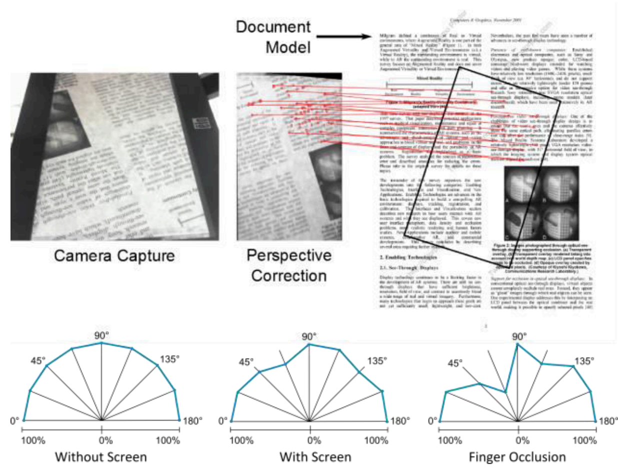

Figure 9. Top: Registration process, red lines show matches. Bottom: Location accuracy at different angles without screen (left), with screen (center) and partial occlusion (right).

Capture-based 2D Registration

Model-based interactions require knowing the location and orientation of the tPad in relation to a physical object below. We implemented this process, known as registration, for paper documents using surface capture and a feature-matching algorithm. 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 features from the document (red lines in Figure 9-Top). 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 [31], and use the Fast Retina Keypoint (FREAK) descriptor [3] to perform the search.

Figure 9-Bottom shows the success rate of our registration algorithm tested at the center of the 10 pages of a sample document [4] at 9 different angles and under three conditions: no-screen, with screen (no digital content shown) and partly occluded (with a finger pointing to the middle of the display). Registration works at 10 FPS approx. Results show our implementation works efficiently for most angles, with performance decreasing with the quality of the image (screen and partly occluded conditions) particularly for the angles between 45° and 90°. Results similar to the no-screen condition can be expected when using a surface-capture display due to the higher quality image, lack of perspective inaccuracy, and direct contact with the object (no screen or finger occlusion).

USER EVALUATION

We assessed the impact of transparent display interactions in everyday mobile tasks and collected user feedback on the perceived value of the proposed interactions. 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 [7, 12] and the costs of switching are severe [26]. For this reason we investigate the use of tPad on such a common and costly task. A step in supporting better multitasking in mobile devices is Apple's iPhone multitask bar (two taps on the home button). In this experiment we evaluated the impact of *flipping* and *tap'n flip*, two novel tPad interactions, on an information seeking task involving switching between multiple applications. *Flipping* allows having two applications running on different sides, where one of them can be the main application and the other the source of information. *Tap'n flip* simplifies data transfer between applications as the tapped contents are copy/pasted onto the other side. We study two hypotheses:

H1.a – *Flipping* enables faster application switching than existing methods, i.e. the home button and multitasking bar, when working with multiple applications.
H1.b – *Flipping* enables faster application switching than existing methods, when the number of applications in the device is large.
H2 – *Tap'n Flip* outperforms flipping when multitasking between multiple applications and when the device has a large number of applications installed.

Task – The experiment application asks users to 'collect' a number from another application (Figure 11a). Users navigate to the requested application, collect the number, and navigate back to the experiment application. Figure 11 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 three numbers. Participants used the tPad-D prototype which supports application switching via the home button and flipping gestures.

Design – Independent variables were application switching *method* and *number of applications*. Application distance was a random factor. We considered four switching methods: *home* (H – a button push goes to the main screen), *multitasking bar* (MB – a button press goes to the main screen and shows the recently used apps at the bottom – Figure 11b), *flipping* (F) and *tap'n flip* (TF). The number of apps varied from 1 to 3. The application distance was random between 0 and 3. Participants were trained with each condition after the experimenter demoed the task.

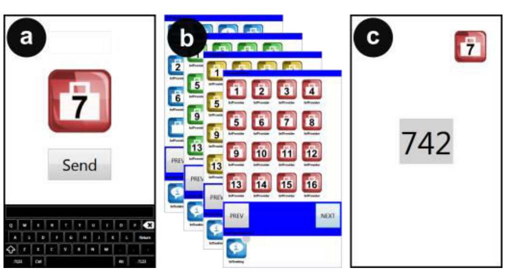

Figure 11. Information seeking experiment. A) Starting point. B) Finding the target application. C) Collecting the data.

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. The experiment lasted approximately 30 minutes. We used a Latin-square design to counter-balance the conditions.

Measures – We recorded search time 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 (stdev 3.6), F at 8.7 sec (stdev 3.7), MB at 8.9 sec (stdev 2.9) and H at 9.5 sec (stdev 2.8).

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 (stdev 0%), F at 0.7% (stdev 8.3%), H at 0.8% (stdev 10%), and MB at 1% (stdev 9.8%).

Participant's Ratings – Results showed a significant difference between *methods* in efficiency ($\chi^2(3)$=13.757,

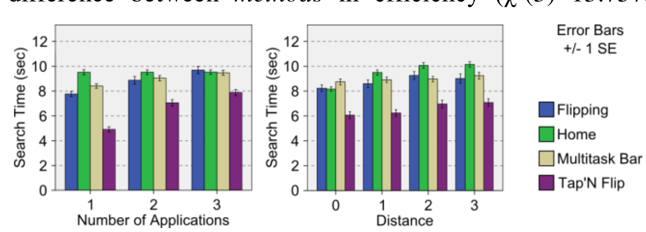

Figure 10. Information Seeking Results.

p<0.005), enjoyment ($\chi^2(3)=13.294$, p<0.005) and preference ($\chi^2(3)=13.900$, 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 and preference) users rated the *switching methods* the same order: TF, MB, F and H.

Discussion

Results showed that flipping is generally faster than *home*, and equally as fast as the *multitasking bar*. This difference is maintained as the number of multitasking applications increases and when the number of installed applications is large (navigating between multiple screens). Therefore, our results support **H1.a** and **H1.b** against *home,* but reject them against the *multitask bar*. 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 application. Therefore, **H2** is confirmed. Moreover, *tap'n flip* had a marked 0% error rate, and *flipping* induced less errors than the *multitask bar*. Finally, preferences reflect the superiority of *tap'n flip*, with the highest rank in all three aspects and significantly higher in enjoyment, meaning that it offers a pleasant user experience. *Flipping* is perceived as equally good as the *multitask bar*, suggesting it alone might not make a difference for users.

Image Capture Experiment

Another important everyday use of mobile devices is to capture information by taking a picture [22]. We focus on the situation where the information is contained on an object smaller than the handheld device. 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. 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. We study one hypothesis:

H3 – *Grabbing* enables faster image capture of nearby planar surfaces than traditional camera devices.

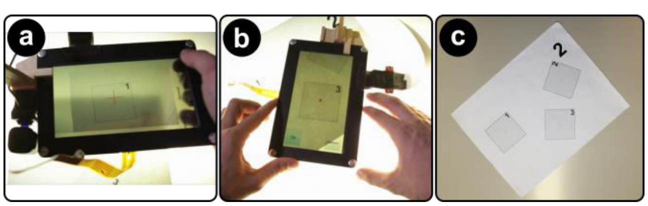

Figure 12. Information capture experiment. a) Normal picture taking. b) Grabbing by means of surface capture. c) Half-size targets on the paper sheet used.

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. We included conditions where participants cropped 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 demoed 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

The data did not comply with the ANOVA assumptions and therefore we used the same analytical tools as in the previous 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 (stdev 5.7) and the normal method at 12.3 sec (stdev 7).

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 (stdev 3), with half and at 2.26 mm (stdev 3.32) and quarter at 2.49 mm (stdev 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)

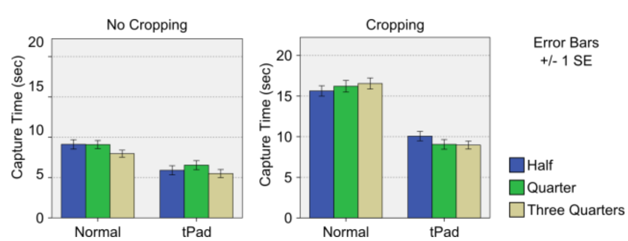

Figure 13. Information Capture results.

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° (stdev 1.4°) than with a normal device at 1.7° (stdev 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

These results suggest that, in general, the *grabbing* interaction on planar objects supported by a surface capture display is twice as fast as picture taking with current devices. Grabbing also results in better aligned images (for easier reading). Therefore, our results support **H3**. This performance gain is reflected in the participants' preference for the tPad grabbing interaction. Interestingly, results show that with tPad, acquiring a cropped image takes roughly the same time than taking a picture with current devices (10 sec). This suggests that tPad users can have higher quality images without sacrificing efficiency.

DISCUSSION

We have shown that transparent displays enhance the usage of mobile devices beyond the commonly discussed mobile augmented reality applications. We further highlight our contributions and discuss limitations of our implementation.

Value of Transparency Alone

We demonstrated the value that transparency alone can provide to mobile devices. Overlay interactions, such as sketching or querying can prove beneficial in many contexts. For example, the transparent display can serve as a ruler. In analytic settings, the graph explorer application demonstrates how, after minimal configuration, tPads enable information extraction from physical documents. To our best knowledge, such usages have not been previously demonstrated with transparent-display mobile devices.

Advanced Transparent-Display Capabilities

We demonstrate the benefit of tPad for two common mobile tasks, application switching and image capture. However, for both we used advanced tPad capabilities beyond simple transparency. Dual-side display & input is a simple augmentation already present in industrial prototypes [38]. More elaborated is surface capture which, with our camera approach, prevented us from presenting all set of interaction techniques with one single prototype. Nonetheless, it allowed us to focus on the interactions exploration without focusing on low-level challenges. Alternatives include, as already mentioned, pixel-level capture similar to Microsoft's PixelSense technology for tabletops, or the use of multiple smaller cameras with wide angle lenses affixed to the corners of the display. Other promising unexplored 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).

Color Mixing Challenges

A significant usability obstacle when using transparent displays is the interference of colors between the physical and virtual objects which can impair legibility of the digital content [11, 36]. We propose three potential solutions to resolve this conflict. The simplest solution tries to re-locate UI elements to areas where color mixing is minimal (similar to [37]). A more elaborate solution relies on concepts of color theory to predict color mixing and compensate the displayed color [36]. An optimal solution uses a display technology that integrates the advantages of LCD and OLED displays [23]. To show digital content, the LCD layer shows a black pixel to block the light from behind, while the OLED layer shows the colored pixel. For transparent areas the LCD could show a white pixel while the OLED shows nothing.

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 interactions. 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. Finally, our model-based interactions are limited to text documents for which we have the original PDF.

CONCLUSION

In this paper we demonstrate that transparent-display mobile devices facilitate rich interaction techniques, many of which are not easily possible on common 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.

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. Adler, A., Gujar, A., Harrison, B., O'Hara, K., and Sellen, A. 1998. A diary study of work-related reading: design implications for digital reading devices. In Proc. CHI '98.
3. Alahi, A. and Ortiz, R. and Vandergheynst, P. 2012. FREAK: Fast Retina Keypoint, IEEE Conference on Computer Vision and Pattern Recognition, 2012.

4. Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., and MacIntyre, B. 2001. Recent Advances in Augmented Reality. IEEE Comput. Graph. Appl. 21, 6.
5. 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.
6. Bimber, O. and Raskar, R. 2005. Spatial Augmented Reality: Merging Real and Virtual Worlds. A. K. Peters, Ltd., Natick, MA, USA.
7. Böhmer, M., Hecht, B., Schöning, J., Krüger, A. and Bauer, G. Falling asleep with Angry Birds, Facebook and Kindle – A large scale study on mobile application usage. Proc. MobileHCI ’11, ACM (2011), 47-56.
8. Concept Phones. (2013, March 17). Transparent Phones. [Website]. Retrieved from http://www.concept-phones. com/tag/transparent-phone/.
9. Concept Vending Machine With See-through Display (2013, September 13). Youtube [Website]. Retrieved from http://www.youtube.com/watch?v=8pPCIPHTe3M.
10. 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.
11. 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.
12. Dunlop, M. and Brewster, S. 2002. The Challenge of Mobile Devices for Human Computer Interaction. Personal Ubiquitous Comput. 6, 4 (January 2002).
13. Elgan, Mike. ComputerWorld.com. (2013, Sept. 4). Are transparent displays dumb or brilliant? (2013) [Website], Retrieved from http://goo.gl/R9MhQN.
14. Fujitsu.com. (2013, Sept. 16). The Future of Workplace. (2011) [Website], Retrieved from http://goo.gl/79IZ8c.
15. Futuba’s OLED road map. oled-info.com. (2013, Sept. 16). [Website], Retrieved from http://goo.gl/mT4pAq.
16. 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.
17.GSM arena. (2013, March 19). Lenovo S800 – Full phone specifications. (2011) [Website]. Retrieved from http://www.gsmarena.com/lenovo_s800-4862.php.
18. 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.
19. Hinckley, K., Dixon, M., Sarin, R., Guimbretiere, F., and Balakrishnan, B. 2009. Codex: a dual screen tablet computer. In Proc. CHI '09. ACM.
20. Iwabuchi, M., Kakehi, Y., and Naemura, T. 2008. LimpiDual touch: interactive limpid display with dual-sided touch sensing. In ACM SIGGRAPH 2008 posters (SIGGRAPH '08). ACM.
21. Kim, K., and Elmqvist, N. 2012. Embodied lenses for collaborative visual queries on tabletop displays. Information Visualization, 11(4), 319-338.
22. Kindberg, T., Spasojevic, M., Fleck, R., and Sellen, A. 2005. The Ubiquitous Camera: An In-Depth Study of Camera Phone Use. IEEE Pervasive Computing.
23. Kiyokawa, K., Ohno, H., and Kurata, Y. 2002. Occlusive optical see-through displays in a collaborative setup. In Proc. SIGGRAPH '02. ACM.
24. 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.
25. Lee, J.H., Bae, S.H., Jung, J., and Choi, H. 2012. Transparent display interaction without binocular parallax. In adjunct proc. UIST '12. ACM.
26. Levia, L.A., Böhmer, M., Gehring, S. and Krüger, A. Back to the app: The costs of mobile application interrruptions. Proc. MobileHCI ’12, ACM (2012), 291-294.
27. 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.
28. 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.
29. 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.
30. PDFKit.Net. TallComponents.com. (2013, Sept. 16). [Website], Retrieved from http://goo.gl/Vr67lV.
31. Rosten, E. and Drummond, T. 2006. Machine learning for high-speed corner detection, European Conference on Computer Vision, vol. 1, pp. 430-443, May. 2006
32. Schilit, B., Golovchinsky, G., and Price, M. 1998. Beyond paper: supporting active reading with free form digital ink annotations. In Proc. CHI '98, pp 249-256
33. Schmalstieg, D., Encarnação, L. M., & Szalavári, Z. 1999. Using transparent props for interaction with the virtual table. In Proc. I3D '99. ACM.
34. Sharma, A., Liu, L., Maes, P. 2013. Glassified: An Augmented Ruler based on a Transparent Display for Real-time Interactions with Paper. In Proc. UIST '13 Demos.
35. Spindler, M., Tominski, C., Schumann, H., and Dachselt, R. 2010. Tangible views for information visualization. In Proc. ITS '10. ACM.
36. 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.
37. Tanaka, K., Kishino, Y., Miyamae, M., Terada, T., and Nishio, S. 2008. An information layout method for an optical see-through head mounted display focusing on the viewability. In Proc. ISMAR '08. IEEE.
38. Transparent dual-touch display Android smartphone prototype #DigInfo. (2012). [Website]. Retrieved from http://www. youtube.com/watch?v=Xbyoool5ZwY.
39. Uy, M. 2009. Integrated sensing display. U.S. Patent No. 7,535,468. 19 May 2009.
40. Wigdor, D., Forlines, C., Baudisch, P., Barnwell, J., and Shen, C. 2007. Lucid touch: a see-through mobile device. In Proceedings of the 20th annual ACM symposium on User interface software and technology (UIST '07). ACM.
41. Wu, C. H., Robinson, S. J. and Mazalek, A. 2008. Turning a page on the digital annotation of physical books. In Proc. TEI '08. ACM.
42. ZXing.Net. (2013) [Website]. Retrieved from https://code.google.com/p/zxing/