Contact Augmented Reality: An Exploration of its Design and Implementation

1st Author Name
Affiliation
Address
e-mail address

2nd Author Name
Affiliation
Address
e-mail address

3rd Author Name
Affiliation
Address
e-mail address

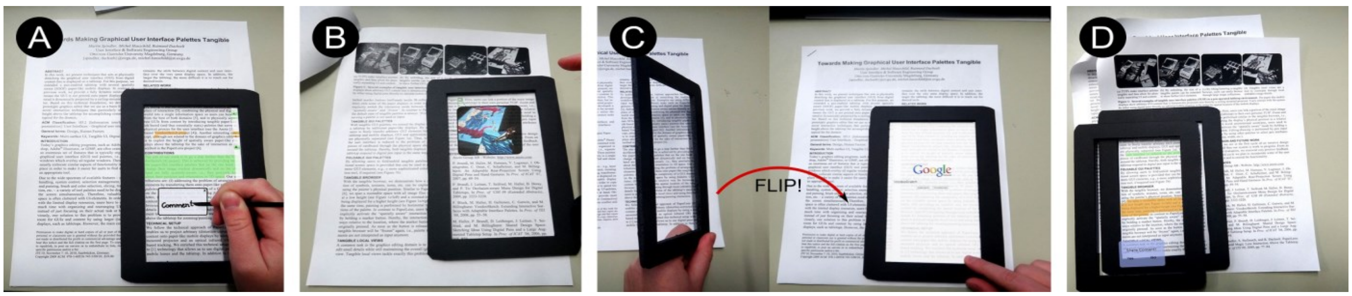

Figure 1: Mock-ups for a transparent portable device implementing Contact Augmented Reality for active reading tasks: A) Pen-based scribbles, B) access to color images, C) flipping to online-search for selected content, and D) stacking for content sharing.

ABSTRACT

We present Contact Augmented Reality (cAR), an approach to augmented reality where a mobile device with a transparent display rests on top of the augmented object. By following an iterative and user-centric design approach we identified interaction techniques cAR devices and organized them in three categories: contact-based, content-based and off-contact. We built two prototype cAR devices and explore their usage for active reading tasks. A first low-level prototype uses tabletop and transparent tangibles and allowed us to iterate quickly over alternative designs. The second and higher-level prototype, called the tPad, is a 7'' touch LCD display with an external camera and resting on a light-table. A paper document is placed on the light-table and the tPad is placed on top of the document. The tPad uses the external camera to identify the document, and determine its location and orientation via feature tracking. We collected initial user feedback and elaborate on the HCI and technical challenges to address before realizing cAR.

Author Keywords

Contact Augmented Reality, Transparent Portable Devices, tPad, Transparent Displays, Tablets, Active Reading

ACM Classification Keywords

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

General Terms

Design, Human Factors

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

INTRODUCTION

We are witnessing a new trend in display technologies where the display itself is transparent and allows one to view both virtual content and physical objects at once. Most conceptual demonstrations of this technology [7, 8, 11, 19] resemble existing augmented reality (AR) applications [3, 6]. One less explored space is that of directly resting the display on top of a physical object, such as a paper form, a book or a map. This direct contact provides spatial alignment between the display and object at a very short distance, enabling the creation of a mobile augmented reality which does not present the registration and rendering complexities of traditional mobile AR displays [4]. We call this approach *Contact Augmented Reality* (cAR). In cAR registration is reduced to identifying the current document and finding the *relative* 2D location and orientation of the cAR device on-top of it. In cAR rendering no longer requires perspective corrections.

By following an iterative and user-centric design approach, and taking Active Reading [1] as a sample application area, we identified interaction techniques cAR devices. We group these into three categories: 1) *contact-based,* 2) *content-aware and* 3) *off-contact interactions*. Some of these interaction techniques are already known like translation and rotation. However, there are some unique ones to cAR devices like extraction, hand-written triggers, orientation to content, flipping and stacking. We mapped such interaction techniques to known active reading tasks [REF]. The following scenario illustrates the usage of a cAR device and how it differs from current AR technologies:

Jane, a foreign anthropologist, is reading the daily newspaper at a café when she finds an article with the transcript of an interview with a known researcher. As it pique`s her interest she retrieves her transparent tablet (tPad), puts it *on the table, and glides it over the image in the article. tPad retrieves the video associated with the interview and plays it on the right half of its display. As she watches the interview she moves the tPad over the article and through the left-part of the screen taps on the words unknown to her for translation. Jane realizes there might be an error with the data, then she quickly scribbles the # sign on the edge on the page and moves the tPad over it launching a calculator. After finding the error she moves the tPad back to the problematic answer, highlights a few sentences and scribbles the calculation error. Further down she finds the name of the reporter. Jane taps on the name and asks tPad to perform a Google search for the person. Jane finds the authors website, selects his email address and flips the tPad bringing up the email editor. Jane writes to him about the error and attaches both the highlights and the scribbles supporting her point. Jane's colleague Mark shows up at the café and starts discussing the interview with Jane. He puts his own tPad on top of the paper and shows Jane his own comments and highlights. Jane stacks her tPad on top of Mark's and moves them in tandem around the document "pulling" Mark's content she finds interesting. Once at home, Jane reviews the annotations and highlights from her home computer and saves them locally for later study.*

We built two cAR prototypes which helped us explore and apply such interaction techniques. The first prototype is tabletop-based with transparent tangibles, enabling the rapid prototyping and testing of the interaction techniques. The second prototype is a mobile device called the *tPad* (see Figure 1), allowing us to face challenges unique to such standalone device like display and registration.

Our contributions are at the conceptual, interaction design, and technical levels: First, we introduce cAR and differentiate it from existing AR approaches. Second, we identify a series of interaction techniques for cAR devices. Finally, we present two cAR prototypes with their software and hardware components, and show how cAR can be applied to and benefit an every-day task such as active reading.

RELATED WORK

cAR augments a physical object with digital information via a transparent-display portable device that lays directly on top of the object. cAR builds on work on augmented reality, transparent portable devices, and magic lenses.

Augmented Reality

Augmented Reality (AR) enhances the real world by embedding digital content onto it. Bimber and Raskar list three challenges AR faces at its basic level: display technology, registration and rendering [4]. The display technology used determines the complexity of the other two. *Traditional AR* relies on mobile displays carried by the 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 technologies and applications can be found here [3, 6].

On the other side, *Spatial AR* (SAR) relies on displays fixed in the environment (e.g. projections, transparent LCDs, etc). Knowing the exact location of the display and the augmented object provides SAR applications with a property called *spatial alignment*, a linear correspondence between virtual content and real world objects. Spatial alignment makes it easier to create AR applications because they require simpler operations for registration and rendering, while offering solutions to the limitations of traditional AR. Nonetheless, to preserve the spatial alignment both display and augmented object should remain spatially fixed, limiting SAR to non-mobile applications [4].

Contact Augmented Reality (cAR) incorporates elements from both traditional and spatial AR. From traditional AR it maintains the vision of a mobile device that can augment virtually any object and can be moved around by the user. From spatial AR it incorporates the property of spatial alignment, thus the knowledge about the location of both display and object and the linearity between them. In brief, cAR is both mobile and spatially aligned.

Transparent Handheld Devices

Transparent handheld devices are the topic of popular design concepts ranging from smartphones to tablets [7, 8]. Such concepts are instrumental in proposing novel interacttions, some of them similar to the ones explored in this paper; however they are short in discussing appropriate usage contexts and technical limitations. Nonetheless, a few devices are reaching the public (e.g. [10]) and several electronic components companies outline the production of such displays as objectives for 2014 and onwards [REF].

While the development of such displays moves forward there exists limited HCI and interaction design (IxD) research on the topic. One studied aspect is their support for touch interaction on the back of the device. LucidTouch [19] simulates such transparency with a camera-based see-through portable device (pseudo-transparency) and user studies showed that users are able to overcome the fat-finger problem and acquired obstacles with higher precision and using all 10 fingers simultaneously. LimpiDual [11, 16] also studied back of the device, front and dual interactions using an optical see-through display. Their results showed that while back of the device has indeed higher precision, it's slower than front and dual.

Other researchers investigated the challenges for optical HMDs in field deployments of AR, the major one being color mixing; color mixing happens when the display pixels mix with the background. Color mixing changes the colors (affecting color encoded information) and reduces the legibility of the display content.

Our research goes beyond back of the device input and field deployments, and proposes an exploration of the interaction design and applications of transparent displays for cAR.

Magic Lenses and Tangible Views

cAR is inspired by Bier et al.'s Toolglass and Magic Lenses [4]. Toolglass and magic lenses widgets were designed for WIMP interfaces and 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 a tool-glass and magic lenses approach to augmenting a biology laboratory book.

Similarly, Spindler et al.'s tangible views also inspire our work [19]. Tangible views provide a complementary display for the content visualized in the tabletop computer. The authors propose an interaction vocabulary for manipulating such tangibles including: translation, rotation, freezing, gestures, direct pointing, toolbox metaphor, visual feedback and multiple views.

We take the idea of toolglasses and magic lenses to the field of transparent displays; here the underlying physical object is visible and modifications happen on the digital model of such object. From the tangible views we adapt some of the interactions they propose, and adjust them to a tangible view that is both transparent and only with 3DOF (2D translation and rotation). Finally, we take the concept introduced by the a-book forward and generalize it into the notion of cAR. However, we depart from it in several ways: first we use a fiducial markers and feature tracking registration, second we explore off-contact and content-aware interactions, third we rely on transparent display technologies.

CONTACT AUGMENTED REALITY – cAR

We introduce Contact Augmented Reality (cAR) as a particular case of augmented reality where a handheld device with an optical see-through display rests directly on top of the object it augments with the display aligned to it.

Our understanding of cAR is guided by the vision, portrayed by Jane's scenario and illustrated in Figure 2, ~~of users placing their transparent handhelds directly on top of paper documents they want to enrich with digital properties.~~ Key to this vision is users place their devices on any paper document; the device rests on the document or on the table and the user moves it in and out of the document when digital functionalities are needed. Moreover, this vision acknowledges the affordances of physical objects; in the case of paper documents the convenience of reading on printed paper, the naturalness of manipulating it, and the preservation of properties like texture, color, age, and wear.

Figure 2. The cAR concept – Jane uses her cAR device to annotate the newspaper and watch related video content.

This vision also extends to other surfaces like a map or a poster in the wall, and any other object where the device could be overlaid. The fundamental requirement is for the device to be able to establish a frame of reference with the object (a coordinate system), and to locate the device in relation to such frame of reference. In this sense, cAR takes distance from traditional and spatial AR systems which determine the location of both the display and the augmented object in three dimensions[1]. cAR requires to know the location in two dimensions from the frame of reference of the object's coordinate system; i.e. using a cAR device on a book while in bed or while sitting on a table makes no difference when determining its location.

An important consequence of the spatial alignment between the cAR device's display and the augmented surface is that digital content is to be rendered on a virtual plane parallel to the object and to the screen; which means that no homographic transformations are needed for aligning the digital content on augmented. Therefore, building graphical user-interfaces for cAR devices can leverage most existing 2D authoring frameworks and developments platforms.

According to the vision cAR devices augment only when in contact with the augmented object, acting as normal mobile device when not on top of an augmentable object. Therefore, augmenting an object does not require the user to hold the cAR device in front or above the object for an

---

[1] Even projector-based spatial augmented reality systems require knowing the location in three dimensions of the augmented objects in order to do the perspective corrections of the projected image.

extended period of time, with the expected costs in terms of physical demand. Another implication is that augmentation is triggered by means of implicit interaction, that is, placing of the cAR on top of the object and a successful registration can trigger the application associated with the target object. A final implication is that a cAR device can augment an object without being the center of attention all the time; acting at times as an ambient or a secondary display.

In summary a cAR device integrates virtual and physical worlds by:

- augmenting physical objects upon contact,
- preserving the affordances of physical objects,
- integrating display and input functionalities, and
- reducing registration and rendering to 2D.

RESEARCH APPROACH

The main contribution of this paper is to investigate the notion of contact augmented reality in terms of interaction techniques and implementation challenges. For this purpose we followed a three phase research approach. In the first phase we engaged in a user-centric design process in order to give wider validity to our ideas. To effectively involve users and ground the design discussion between researchers and with the users we used "active reading" as a scenario and inspiration tool. We chose active reading because it is a familiar activity and, as shown later, provides context for the elicitation of concrete design choices. Moreover, active reading leverages the affordances of paper [REF, REF] and benefits from digital functionalities like e.g. search and copy+paste [REF, REF, REF]. We ran several design sessions and our methods included brainstorming, focus groups and semi-structured interviews. These sessions led to particular interaction techniques and application features.

In the second phase we built a low-fidelity cAR prototype using a tabletop computer and transparent tangibles. We used this implementation to explore multiple interaction techniques and early architectural design challenges. This prototype allowed us to collect initial feedback from users of such systems by exposing them to active reading tasks using the implemented interaction techniques and features.

In the third phase we built the tPad, a prototype cAR device using a 7'' touch-sensitive LCD display, a light table, and an embedded camera. We used this implementation to explore the actual hardware and software architecture challenges of such device.

Active Reading

Adler and Van Doren define active reading as the combination of reading, critical thinking, and learning [1]. Active reading happens in tasks where people read to gain knowledge and understand all aspects of a certain topic like reading to self-inform, reading to cross-reference or reading to support discussion [REF-Adler98]. This process often includes tasks such as underlining, highlighting and annotating as well as information seeking and non-sequential navigation within documents [REF - Schilit98, Golovchinsky08, 20].

A number of projects build technologies to support active reading. Some projects seek to augment phy-sical paper documents adding digital functionalities through fixed or mobile projectors [9, 22]. Other projects focus on digital documents exploring different form factors like desktop, tabletop, and tablet computers [14, 17, 15, 11, 20]. The following list contains typical features often found in such projects: reading, outlining, underlining, highlighting, searching, scribbling, digital annotations, note-taking, non-sequential navigation, information seeking, quoting, comparing, and content sharing.

Although our goal is not to create a cAR system for active reading that outperforms the existing ones, we use the list of features as an inspiration source for the exploration of cAR devices. For example, in the user-centric design sessions, we asked ourselves and our participants how they would support highlighting or annotations with a transparent portable device. As interaction techniques start to emerge, we re-analyzed the active reading features looking for compelling mappings.

cAR INTERACTION DESIGN

We followed a user-centric interaction design with two rounds of activities carried out by two geographically distributed design teams. In the first round we explored the possibilities of a cAR device for active reading, aiming at uncovering relevant interaction techniques (see Figure 3). Activities included a series of interviews with potential users (design team 1), and two brainstorming sessions (one for each team). ~~Figure 3 shows users drawing the user-interface they envisioned on a mock cAR device.~~ In the second round we explored advanced affordances unique to cAR devices (e.g. flipping, content-awareness, and visibility upon stacking) and tried matching all of the identified techniques to active reading features. Activities included two brainstorming sessions ~~(one for each team).~~ The results of this user-centric interaction design are grouped into interaction techniques, mappings of the techniques to particular active-reading features, and implementation requirements.

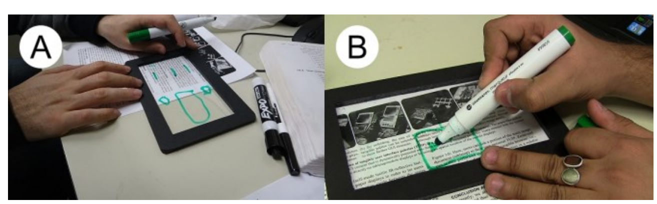

Figure 3. User-centric design of a cAR active reading app: A) viewing related content, and B) playing associated video.

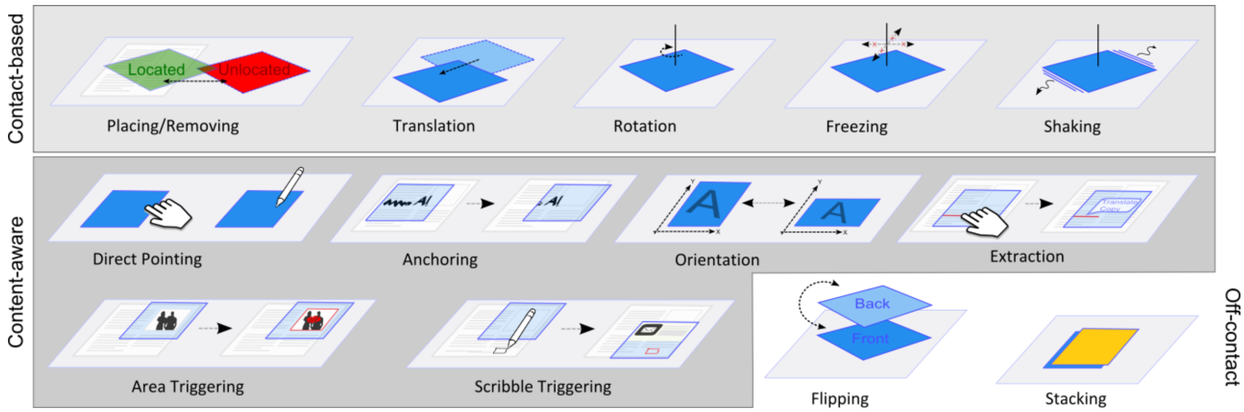

Figure 4. cAR interaction techniques

Interaction Techniques

We present cAR interaction techniques in three categories: contact-based, content-aware and off-contact. Some techniques come from related areas like tangible computing or mobile interactions; however, others are unique to cAR devices like flipping and scribble triggers. Figure 4 illustrates the basic interaction techniques for cAR devices.

Contact-based

Contact-based interactions are manipulations of the cAR device in relation to the object upon which it rests.

Placing/Removing – The basic interaction for cAR devices is placing the device on top of the augmentable object. Upon contact, the device tries to identify the object below and respond to it. In simple cases the device can respond to simple properties like color or type (e.g. text, drawing, skin, paint, etc.). In complex cases the device has a model of the underlying object (e.g. a PDF of the printed paper document). Conversely, removing the cAR device from the object can be interpreted as an implicit interaction technique ~~and respond to it~~. This interaction is similar to the one studied in tabletop systems [REF, REF], with the difference that it is the tangible device the one responding to it.

Translation – A user can translate a cAR device horizontally (in X and Y). The device can interpret movement in each axis as input. An application can use translation in X and Y to translate a virtual plane accordingly and maintain a correspondence between such virtual plane and the real object.

Rotation – The device can also interpret the rotation of the device as input. This rotation can either be relative to the original placement or to the object's "north". Similar to *translation*, an application can use changes in rotation to rotate the virtual plane accordingly and preserve spatial alignment.

Freezing – The device might ~~also~~ freeze its current state by ignoring changes in translation and rotation. By freezing the current state, the user can move the device freely around while preserving the current state of the application, e.g. the current view in the virtual plane.

Shaking – Similarly to tangible views [19] a user can manipulate a cAR device in known patterns known as gestures. In this first exploration we focus on shaking the device while still lying on the augmented object. Further explorations could focus on trajectory movements and combinations of translations, rotations, and placing and removals of the device from the augmented object.

Content-Aware

Content-aware interaction techniques leverage knowledge about the underlying physical object.

Direct Pointing – Direct pointing allows the users to interact with the content displayed on the cAR display either by touch or pen. By means of direct pointing users can interact with user-interface elements like menus or issue touch gestures. Users can also interact with spatially-aligned digital content visible at the time.

Anchoring – Anchoring is the basic interaction to attach digital content to a particular location of the physical object. Anchored content exists on a virtual layer spatially aligned with the physical object. For example, scribbles are created anchored to specific locations of a paper document.

Orientation – A cAR application can adjust the orientation of its user-interface based on the coordinate system of the augmented object. This technique is similar to the change of orientation of mobile phones according to the way users hold them (portrait –vs– landscape). For example, a cAR application's user-interface can adjust to the direction of the text and orientation of the paper document.

Extraction – Users can ~~also~~ interact with digital elements of the model of the augmented object. For example, a cAR application for a magazine can allow users to select words and look up for definitions, translations, copy, etc.

Triggers – Triggers are areas in the physical object that activate special responses by the cAR device when placed of top of them. Triggers can be model-define or user-defined. Model-defined triggers are special zones statically defined in the model like a coordinate or a bounding box; e.g. an image on a newspaper could trigger a video associated with the article. User-defined triggers are physical modifications on the physical object which are read and interpreted by the device; e.g. a hand-drawn square launches the browser application by moving the cAR device on top of it.

Off-Contact

Off-contact interaction techniques do not require the cAR device to rest on top of the augmented object. When not in contact, a cAR device could incorporate any interaction technique for tablets or smartphones. In this section we focus on techniques particular to cAR devices.

Flipping – A cAR device can be flipped around and have the other side of the screen on top. The device can apply different modifications on the user interface upon flipping like, for example, a 2x zoom, an inverse color filter, a translate feature, or launching a secondary application for the actual document.

Stacking – A cAR device can also be stacked on top of another one. Given that both displays are transparent, the content on both devices is visible at once. A natural usage of stacking is content sharing, where digital content created in one device can be pulled up (by the device on top) or pushed down (to the device below).

Mapping to Active Reading

Figure 1 (front page) shows design sketches showing different active reading features and the supporting interaction techniques. For example, users can add hand-written notes using a stylus (Figure 1A), access a color version of an image by tapping on the "Figure X" text on the document (Figure 1B), perform an online search on a selected word by simply flipping the device (Figure 1C), and share content by stacking one device on top of the other (Figure 1D). Table 1 shows the complete set of mappings between interaction techniques, and active reading features.

Implementation Requirements

In order to implement the proposed interaction techniques and thus build a cAR application to support active reading, an implementation should fulfill the following technical requirements:

RQ1 – Registration: a cAR device should identify the object upon which it rests, and determine its position and orientation in relation to such object. For example, document id, page number, x, y, and theta.

RQ2 – Touch and Pen Input: a cAR device should allow users to interact with digital content via touch (for coarse interactions like pushing buttons or accessing content) and pen (for fine interactions like scribbling text).

RQ3 – Dual Side Interaction: a cAR device can flip over, thus interaction should happen on either side of the screen.

RQ4 – Device Integration: cAR devices should be able to identify one another when stacked, and seamlessly integrate for content sharing purposes.

Int. Technique	Active Reading Feature	Prot
Placing/Removal	Document recognition, reading	Both
Translation/ Rotation	Browsing virtual content anchored to locations in the physical document.	Both
Freezing	Ignores physical changes in translations and rotation, thus maintaining the current digital view.	tPad
Shaking	Undo for highlights and scribbles	Both
Direct Pointing (hand and pen)	UI interaction, creating and manipulating digital contents	Both
Anchoring	Adds notes and scribbles to particular places of the physical document.	Both
Orientation	Adjust the device's UI to the text orientation	tPad
Extraction	Selecting words from the text for the purpose of in-document search, online search, and translation.	Both
Area and Scribble Triggers	Starting a video when hovering an image, and launching app when hovering a particular glyph.	tPad
Flipping	Full screen online-search of selected word, and magic-lens color filter.	TT
Stacking	Content sharing from bottom device to top device.	tPad

Table 1. Mapping between cAR interaction techniques, active reading features and the prototypes. TT: Tabletop prototype.

RQ5 – Object Model: An object model for the augmented object should exist, containing meta-data for the cAR device to interpret its current location (frame of reference, orientation, and contents).

TABLETOP PROTOTYPE

We built a tabletop prototype in order to support the design process by allowing for faster prototyping and testing of design alternatives without the technological limitations of a high-fidelity prototype.

Conceptually, the physical documents (books or sheets of paper) are substituted with the interactive surface of the tabletop which also provides touch input capabilities (RQ2). The cAR device itself is simulated by a transparent tangible (acrylic glass) that is spatially tracked on the tabletop via fiducial markers (RQ1). By flipping the device, the tabletop recognizes a different marker located on the flipped side of the acrylic and linked to the same physical device (RQ3). A document viewer shows the augmented document, aligns the created content according the actual page, and extracts the words users tap on for further interaction (RQ5). Finally, this prototype does not support device integration (*stacking* - RQ4).

We render an image of a blank page, overlaid by a rich-text document, to simulate real paper. References or figures in the document are linked to other media, including videos. The UI for the simulated device is shown at the location and orientation of the tangible, giving the impression of a translucent display that can be moved freely on top of a document.

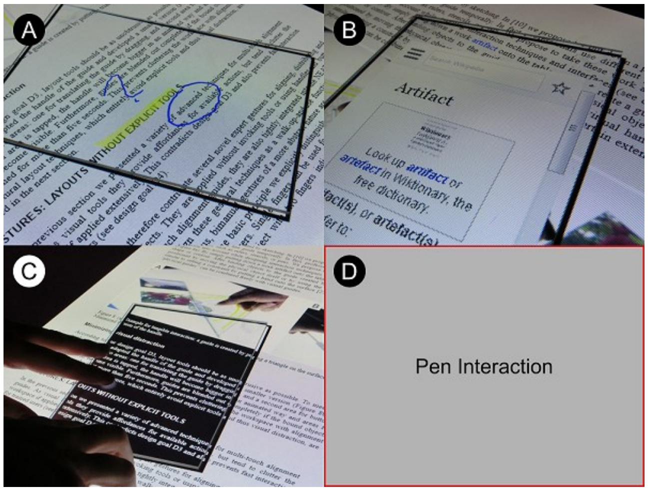

Figure 5. A cAR tabletop prototype: A) highlights and scribbles, B) online-search of selected content, C) inversion magic lens when flipping without selection, and D) pen input.

Figure 5 shows the tabletop prototype. In its final iteration, the prototype supports touch and pen input with an IR pen (Figure 5C-D). We implemented the following features: the user can write free-hand annotations or highlight text (Figure 5A), tap on figures to show an overlay with additional information, e.g., a video, or tap on references to show the corresponding bibliographic entry. Additionally, by flipping the display after selecting a word, users can switch to a browser view showing an online encyclopedia's entry for the word (Figure 5B). If nothing is selected, a color-inverted view is shown, illustrating of how content can be presented through different adaptations (Figure 5C).

Implementation

We implemented the prototype on a Samsung SUR40 tabletop. We attached Microsoft ByteTags to a 7" acrylic glass mockup device. We create transparent ByteTags using IR reflective foil to minimize their obtrusiveness.

Limitations

A natural limitation of this prototype is not being able to show the user interface when the device is not placed on the tabletop. Moreover, it cannot handle stacked devices given that the ByteTags become too difuse for the tabletop to recognize them. Furthermore, quick manipulations of the tangible (translations and rotations) make the fiducial unreadable by the tabletop during the movement, thus hiding the cAR device's interface until the movement stops.

User Feedback

We used the prototype to gather user feedback. Eight users not involved in this project (6 male, 2 female, on average 28 years old) participated. After introducing participants to active reading and the background of the project they had the chance to try out the prototype. The session consisted of a semi-structured interview.

cAR and Interaction Techniques

In general users appreciated the cAR concept and its applicability to active reading situations. Particularly, users highlighted the access to information not already included in the text (e.g. video or color images, or online content), and potential availability of highlights and annotations in digital format later on. Some users indicated it would be better suited for books (rather than for short documents) and for situations where a table is available.

Users understood the functioning of the *translation*, *rotation*, *direct pointing*, and *anchoring* interaction techniques, and their effects on the contents on the device (e.g. menus) and on the virtual layer (e.g. scribbles and notes). Similarly, users appreciated the possibilities offered by content *extraction*, and suggested novel usages like translation and social media sharing. On the other hand, *shaking* and *flipping* received mixed reactions. Shaking was considered inconvenient especially when the device rests on the augmented object. Flipping was perceived to be unintuitive, laborious and too complicated – something the form factor (poor ergonomics) and limitations of the prototype influenced.

Active Reading Support

Feedback was mixed for both highlighting and annotating: For some participants, these are the most important features of the prototype, while others do not see an advantage of combining digital annotations with physical, printed documents. Moreover, some users preferred techniques similar to PDF readers like word-based marking and comment boxes, while others opted for free-hand marking and scribbling. Several participants mentioned the importance of keeping track of where annotations are, highlighting the need for overviews of the content or off-screen markers. Users noted the ability of the virtual layer to offer a larger area for note-taking, than the space available on the physical document. Finally, users frequently mentioned the possibility to export such annotations and extracted content, ranging from simple export or clipboard functionality to integration of some form of social network, where readers could share comments about specific parts of a document.

Most users liked the concept of linking text and pictures in the physical document to additional media (e.g., videos) or metadata (e.g., reference list entries). Users proposed translating individual words or look up terms in an online encyclopedia, even before they were shown this feature, thinking of it as "convenient" and "quite cool". Moreover, completely replacing content (instead of augmenting it) was also well received. Most participants mentioned zooming text for reading assistance as a useful feature, while others proposed automatic translation of the text under the device.

Other Feedback

Several participants mentioned the flexibility of paper as a problem when combined with the inflexible prototype. A small stack of paper, e.g., a printed scientific publication, is unstable when held in hand and often starts buckling. This is aggravated when the user has to hold the device and the paper stack in the same hand. Users mentioned the possibility of using some form of paperclip to fix the device on the paper or proposed a desk as working environment.

Participants also mentioned the offset and a possible loss of context when working through the device due to its thickness and border frame.

TPAD PROTOTYPE

Our second prototype, the *tPad*, is a high-fidelity prototype which helped us to further explore the proposed interaction techniques and the technical challenges to building a self-contained cAR device.

Our prototype uses a transparent LCD display on top of a light table and the documents to augment are printed single sided in white paper. We use a camera attached to the display for registration (RQ1). A touch-overlay allows the display to receive touch and pen input (RQ2), while an accelerometer and magnetic sensors enable flipping (RQ3) and device integration (RQ4) respectively. The tPad runtime holds a PDF version of the document with added meta-data as object model (RQ5).

Figure 6 shows the tPad prototype (top) and the runtime system components. The tPad runtime is designed as an application container with an application launcher (DashboardApp), a general purpose application (CalculatorApp), and a cAR application for active reading (ActiveReader). On startup, the DashboardApp lists the installed applications for the user to select.

The ActiveReader supports all the features listed in Table 1, except flipping due to the fixed and prominent location of the camera. The tPad includes a soft-keyboard to support text entry, and uses *rotation* to control opacity and zooming levels. When on the configuration screen, the user can rotate the tPad in order to make the digital content more or less transparent. Users can also zoon into the digital layer in order to "make" more space for scribbles. The tPad introduces off-screen markers to indicate the location of anchored digital content.

Two novel simple interaction techniques explored in the tPad are *orientation* and *freezing*. For orientation the ActiveReader reorganizes its menus according to the text flow, so that the menus are away from the main reading and interaction area thus avoiding obstruction. This strategy also reduces the presence of fingers and stylus in the image captured for registration. Moreover, users can freeze the tPad on a particular location and the visible digital content will remain fixed regardless of the tPad movements; a user could then move to a different page or pass the tPad to another person while having the digital content visible at that particular location.

The tPad also introduces two complex interaction techniques: *stacking* and *triggers.* We implemented stacking via magnetic switches and magnets embedded on the tPad frame. When physically stacked, the magnets of the device on top align with the magnetic sensors of the device below, which starts a networked pairing process. Upon pairing, users can see both the physical document and the digital content of both displays. At this point, the device

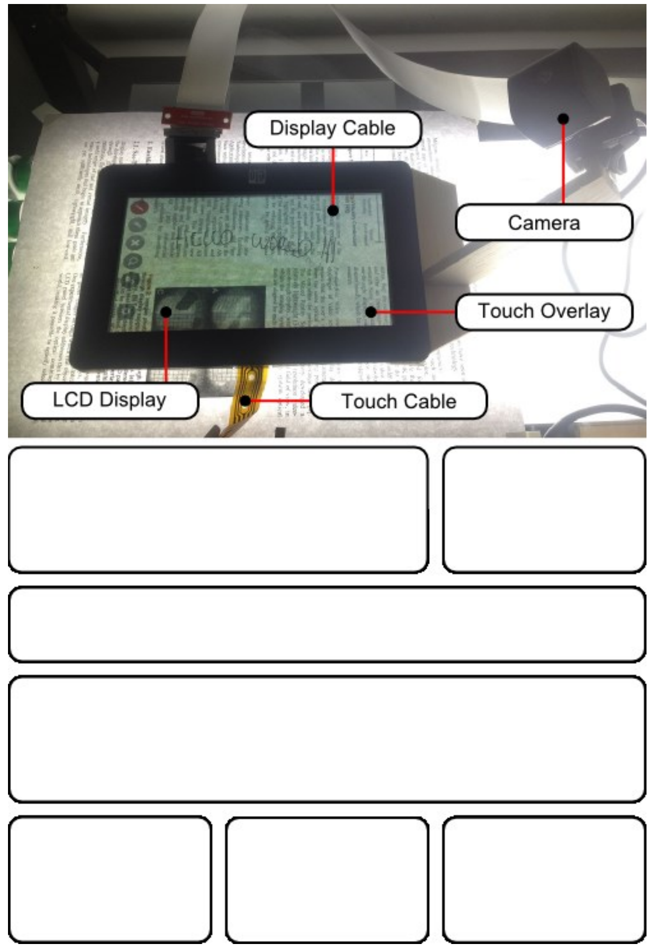

Figure 6. tPad system components.

on top can "pull content" from the device below according to three strategies: *pull all*, *pull current page*, and *pull selection*. *Pull all* transfers all annotations, scribbles and highlights created in the device below for all pages for the current document. Likewise, *pull current page* limits the transfer to the current page. When *pull selection* is chosen the user manually picks the particular objects to transfer. Stacking finishes either by explicitly selecting the un-pair button, or by physically separating the devices.

The tPad supports both model and user-defined triggers. The tPad uses user-defined triggers to launch specific apps; in this implementation a square of roughly 1x1 cms launches the CalculatorApp when the tPad is placed over it. User-defined triggers are device wide and independent of the currently running application. To showcase model-defined triggers, we created meta-data for a sample document such that a video is automatically reproduced when the tPad is place on top of a particular image in the document.

Implementation

Hardware and Software Architecture

Figure 6-top shows the hardware components used for the tPad prototype. We re-purposed an Elo 7'' inch LCD resistive-touch USB display by removing the backlight and

extending its display and touch bus cables [REF]. The LCD rests on top of the physical paper (one sheet of paper at the time) which in turn rests on a custom-built D65 light table. The display and touch overlay are connected to the original Elo controller board. We added a Microsoft LifeCam 6000, and an Arduino Pro Micro controller board at 5V with 4 reed-switches and a ADXL335 3-axis accelerometer. All three components (display controller board, camera and Arduino) are connected to a computer running Windows 7. We use C# and Microsoft WPF for authoring and rendering, and C++ and OpenCV 2.4 for image processing and feature matching. Network messages for content sharing are JSON-encoded and sent via UDP in the local network. The ActiveReader uses the TallComponents PDF kit for accessing pixel-level location information of the PDF contents [REF].

Camera-based 2D registration

To determine location and orientation of the tPad within the document (a.k.a registration) we use a camera attached to the device and a feature-matching algorithm. The camera captures the tPad screen and the underlying surface from above, and the registration algorithm processes it against potential known documents. The algorithm detects the position of the captured image within a digital version of the physical document by matching features from the captured image with features from the document. The location of the image (page number, x-y coordinates, and rotation) maps to the location of the tPad on the document.

The features, also known as keypoints, efficiently describe image patches and are invariant to rotation, noise and scale. To detect significant keypoints we use FAST (Features from Accelerated Segment Test), a fast and high quality corner detector [17, 18]. Once keypoints are computed we use Fast Retina Keypoint (FREAK), a descriptor inspired by the human retina generally faster to compute with lower memory load and more robust to scale, rotation and noise than conventional keypoint descriptors like SIFT, SURF or BRISK [2]. The tPad maintains a database of documents which physical print-outs it can augment. The tPad stores the digital content, graphical representation, and pre-computed FAST-features for each document.

Given the camera position, the captured images have a perspective which is incompatible with our feature matching algorithm; i.e. the algorithm is not invariant to perspective transformations. For each captured image the algorithm corrects the perspective distortion using a static warping matrix, then calculates the FAST-features for the corrected image. If the algorithm finds the corrected captured image within the document, a transformation matrix between both is computed and used to derive location and orientation of the device.

Figure 7 shows the results of our registration algorithm tested at the center of the 10 pages of our sample document [3] at 9 different angles and under three conditions: without screen, with screen (no digital content shown) and partly occluded (with a finger pointing to the middle of the display). The initial results show that our implementation works efficiently for most angles, with performance decreasing as the quality of the image decreases (with screen and partially occluded conditions) particularly for the angles between 45 and 90 degrees. These results demonstrate the feasibility of a self-contained camera-based cAR device, noting that further investigations are necessary in different light conditions and with other technologies like transparent OLED displays.

Limitations

Even though hand-held and mobile the tPad is limited to the light-table and the computer it is attached to. Nonetheless, both hardware and software architecture were designed for a self-contained device, meaning it all could work with minimal modification when transparent OLED displays become available. On the other hand, registration works at only 5 FPS approximately. The prominence of the camera keeps us from exploring flipping, and touch and pen input are limited to a single side (the tPad does not implement RQ3). Finally, the nature of LCD displays kept us from really exploring stacking as the display on-top did not received enough backlight.

DISCUSSION

This section presents the main the challenges we identified for designing and implementing cAR devices.

Object Model

A model of the object being augmented (RQ5) is a fundamental piece for cAR because it supports multiple interaction techniques (e.g. anchoring, orientation, extraction, triggers) and compelling application features (e.g. search, video playback, etc). The question remains on how such model should be created and potentially distributed to a cAR device. In the case of document-based applications like active reading such model could be made available by, for example, the publisher of the physical document, either as a self-contained cAR device application or as a file formatted for a general purpose reader (like our

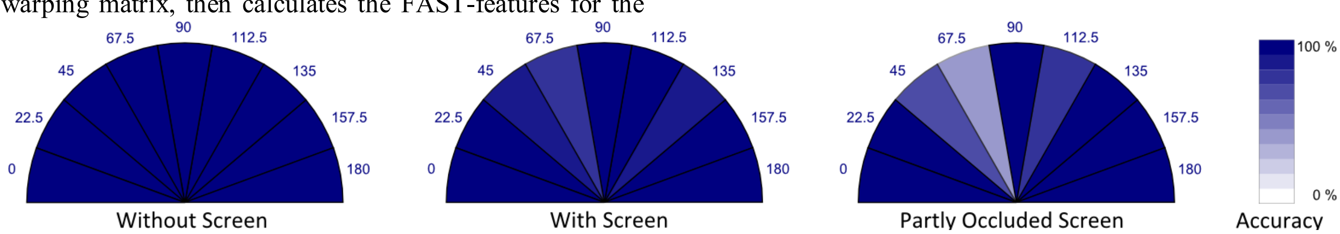

Figure 7. Registration

ActiveReader). In any case the basic contents, meta-data, and media files should be bundled by a third party and distributed to the device. We envision a scenario where the device, upon touching a document for the first time, tries to locate itself within a list of known documents (probably in local storage). If failed, the device delegates the identification of the current document to an online service. The service returns a list of possible titles and content providers for the user to choose.

In another approach, the cAR device attempts to create the model of the document based on the images captured by its camera. A basic image-stitching algorithm could provide the basic frame of reference for *contact-based* interaction techniques, *direct pointing* and *anchoring*. More elaborate techniques like *orientation* and *extraction* would need user assistance or some degree of artificial intelligence (detect text flow direction, extract text from images, etc).

Camera-based Registration and Optimizations

A mayor portion of our implementation was dedicated to the camera-based registration algorithm (RQ1), allowing us to identify its flaws and propose several optimizations.

The main flaw we found in this approach is its impact on the device itself. Given that the camera needs to capture a good portion of the screen, it needs to be elevated from the display plane. Moreover, feature matching algorithms like the ones we used are computationally demanding, sensible to lighting conditions and device orientation, and dependent of the number of observable features. To ameliorate the hardware requirements the device could use wide-angle lenses (with the added image distortion that needs to be compensated), multiple smaller cameras distributed around the display, or pixel-level capture similar to Microsoft's PixelSense technology for tabletops.

The other limitations can be handled through software optimizations. For example, registration could use a multi-level approach to reduce computational load: if the current page is already known, the device tries to detect its new location by searching only on the actual page. In case of failure a search is performed in the surrounding pages (next and previous), and ultimately in the complete document. However, when the document is large enough (e.g. a 700 pages chemistry book) such search could take several minutes. In this case the cAR device could calculate the camera features locally and delegate the search to an online service. For better energy efficiency, a cAR device could only call the registration algorithm when a movement is detected through the accelerometer or upon request by pressing a physical button.

Color Mixing

A mayor usability obstacle when using the tPad was the interference of colors between the paper document and the content on the LCD display, affecting the legibility of the digital content. This phenomenon is similar to the one described by Gabbard et al for projector-based transparent displays [REF], and also occurs in transparent OLED displays. Handling color mixing is an important issue for consumer-ready cAR devices and we propose three potential solutions. The simplest solution tries to re-locate UI elements to areas where color mixing is minimal (similar to [REF]). A more elaborate solution relies on concepts of color theory to predict color mixing and compensate the displayed color (similar to [REF]). An optimal solution uses a display technology that integrates the advantages of LCD and OLED displays: 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 shows a white pixel while the OLED shows nothing.

Device Ecologies

A recurring topic during the design and user-feedback session was the integration of content created in the cAR device with other computational tools including other cAR devices (RQ4). If we consider active reading as part of the creative process, the content from the cAR device should be accessed later for archival or reference. A cAR device should then provide mechanism for integrating not only with others of the same tPad, but also with traditional PCs.

CONCLUSIONS

In this paper we proposed the notion of Contact Augmented Reality, presented a series of interaction techniques for cAR devices, and implemented two cAR prototypes to support active reading. Rather than proposing an optimal tool for active reading support, we aimed at exploring how users would interact with cAR devices and their feasibility with existing technologies.

Our first low-fidelity prototype uses a tabletop computer and acrylic tangibles. A semi-structured used feedback session showed participants understand the cAR concept and the interaction techniques, and appreciated the opportunities offered by *extraction* such as online sharing, translation and saving for later. Participants also highlighted the possibilities it opens for active reading such as rich-media (video), content search and references lookup. We used the insight gained from users of the tabletop prototype to design the tPad, a prototype with all of the elements necessary for a self-contained device. From building the tPad we learnt that a camera-based approach can efficiently determine location in a document, and identified the hardware and software elements necessary to support off-device interaction techniques (*flipping* and *stacking*). The experience building these two prototypes provided the ground to discuss the importance of the *object model*, possible *registration optimizations*, *color mixing* and *device ecologies*.

Our future work includes building a complete self-contained tPad with better image capture, batteries, and embedded-computer without the need of a light table. We also want to explore the multi-level registration and ad-hoc model acquisition.

REFERENCES

1. Adler, M. J. and Van Doren, C. How to Read a Book. Revised edition, Simon and Schuster, New York, 1972.
2. Alahi, A. and Ortiz, R. and Vandergheynst, P. 2012. FREAK: Fast Retina Keypoint, IEEE Conference on Computer Vision and Pattern Recognition, 2012.
3. 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.
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. Carmigniani, J., Furht, B., Anisetti, M., Ceravolo, P., Damiani, E., and Ivkovic, M. 2011. Augmented reality technologies, systems and applications. Multimedia Tools Appl. 51, 1 (January 2011), 341-377.
7. Concept Phones. (2013, March 17). Transparent Phones. [Website]. Retrieved from http://www.concept-phones.com/tag/transparent-phone/
8. Corning Incorporated. (2013, March 17). A Day Made of Glass 2: Unpacked. The Story Behind Corning's Vision. (2012) [Video file]. Retrieved from http://www.youtube.com/watch?v=X-GXO_urMow
9. Dachselt, R. and Al-Saiegh, S. Interacting with printed books using digital pens and smart mobile projection. In Procedings of the Workshop on Mobile and Personal Projection (MP2) @ ACM CHI 2011, 2011
10. GSM arena. (2013, March 19). Lenovo S800 – Full phone specifications. (2011) [Website]. Retrieved from http://www.gsmarena.com/lenovo_s800-4862.php
11. Hinckley, K., Bi, X., Pahud, M. and Buxton, B. Informal information gathering techniques for active reading. In Proc. CHI '12. 2012. ACM.
12. 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, New York, NY, USA.
13. Japan's transparent display vending machine. http://techcrunch.com/2011/12/20/video-japan-vending-machine-2-0/
14. Marshall, C. C., Price, M. N. Golovchinsky, G., and Schilit, B. N. Designing e-books for legal research. In Proc. of the 1st ACM/IEEE-CS joint conference on Digital libraries, JCDL '01, 2001. ACM.
15. Matulic, F., and Norrie, M. C. Supporting active reading on pen and touch-operated tabletops. In Proc. AVI '12. 2012. ACM
16. 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, 2011.
17. Price, M. N., Schilit, B. N. and Golovchinsky, G. 1998. Xlibris: the active reading machine. In Proc. CHI 98.
18. 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
19. Spindler, M., Tominski, C., Schumann, H., and Dachselt, R. 2010. Tangible views for information visualization. In Proc. ITS '10.
20. Tashman, C. S. and Edwards, W. K. Liquidtext: a flexible, multitouch environment to support active reading. In Proc. CHI '11. 2011. ACM.
21. Wigdor, D., Forlines, C., Baudisch, P., Barnwell, J., and Shen C. 2007. Lucid touch: a see-through mobile device. In Proc. UIST '07. ACM, New York, NY, USA, 269-278.
22. 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.

The columns on the last page should be of approximately equal length.

Remove these two lines from your final version.