cAR: Contact Augmented Reality with Transparent-Display Mobile Devices

1st Author
1st author's affiliation
1st author's E-mail address

2nd Author
2nd author's affiliation
2nd E-mail

3rd Author
3rd author's affiliation
3rd E-mail

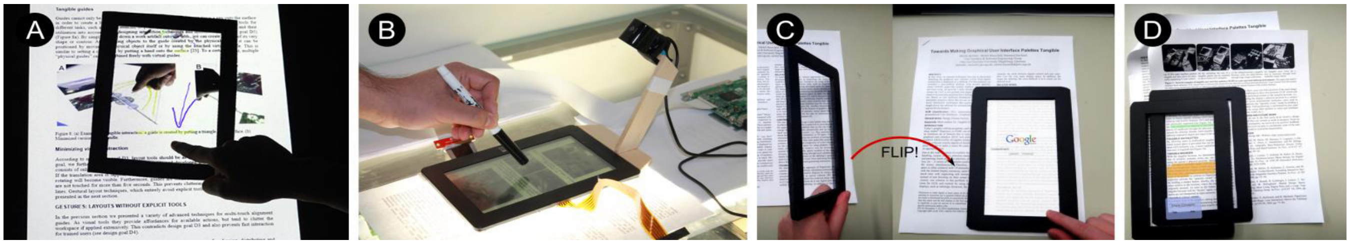

Figure 1: Transparent portable devices implementing Contact Augmented Reality for active reading tasks: A- Tabletop prototype. B- tPad prototype. C- Flipping to online-search for selected content (mock-up). D- Stacking for content sharing (mock-up).

ABSTRACT

We present Contact Augmented Reality (cAR), a form of AR where a mobile device with a transparent display rests on top of the augmented object. cAR is based on the notion that interactions with digital content are enriched by the tangibility of physically moving a device on and off the augmented object. We identify three categories of cAR interaction techniques: contact-based, off-contact and content-aware. We built two prototype cAR devices and explore how cAR can be applied to the sample domain of active reading. Our first low-level prototype, consisting of an interactive tabletop and transparent acrylic tangibles, allowed us to iteratively design and test interaction techniques. The second and higher-level prototype, called a tPad, uses a semi-transparent touch-enable 7" LCD display to be placed on top of back-lit paper documents. The tPad uses an external camera and feature matching algorithms to identify the document and to determine its location and orientation. We report on user feedback on cAR, and elaborate on the salient technical challenges for cAR devices.

Categories and Subject Descriptors

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

General Terms

Design, Experimentation, Human Factors.

Keywords

Contact Augmented Reality, Transparent Devices, Tablet Augmented Reality, Active Reading.

1. INTRODUCTION

Novel transparent display technologies allow users to view virtual content and physical objects at once, an affordance that enables new forms of interaction. While research into transparent display mobile interactions is scarce, conceptual designs portrait mobile interactions [8, 9] inspired on mobile Augmented Reality (AR). Mobile AR overlaps virtual content on top of images of the real-world captured using the mobile camera. Transparent display mobile AR bypasses the digital image of the world, as this can be seen directly with the naked eye. However, transparent display mobile AR faces new challenges. In order to determine the pixel location of display content the system needs access to the relative 3D locations of world objects, the device and the user's head and gaze. Also, binocular parallax affects how users perceive content alignment and their capacity to perform touch interactions [18].

We propose Contact Augmented Reality (cAR), a simple and relatively unexplored form of AR for transparent display mobiles. cAR renders virtual content on top of physical artefacts directly underneath and in contact with the display, such as maps, forms or books. This direct contact provides spatial alignment between the display and the object, avoiding the need to track the user's head and gaze and the binocular parallax problem. Spatial alignment simplifies the *registration* and *rendering* complexities of mobile AR [4, 6]. *Registration* is reduced to identifying the object below the device and calculating their relative 2D locations/orientations. *Rendering* does not apply perspective corrections to the content.

Moreover, cAR brings the benefits of digital interactions without losing the affordances of physical artifacts, such as when planning a trip [25] or active reading [1, 2]. A user browsing a physical foldout of a map can place a cAR device on top of it to highlight points of interest, draw routes and make notes on the device, without affecting the paper map. The cAR-enabled device can be lifted-off so that the user can continue browsing the physical map, flipping parts of it and checking legends without losing context. Resting the device on the map again allows the user to access other virtual content, such as videos or images associated with a specific point of interest. Finally, the cAR device shows user created content as the user re-visits previously annotated regions.

Our contributions are at the conceptual, interaction design and technical levels. First, we introduce cAR and identify interaction techniques for cAR devices. We built two cAR prototypes. Our first low-level prototype consists of an interactive tabletop and transparent acrylic tangibles (Figure 3). A higher-level prototype, called tPad, uses a semi-transparent touch-enable 7" LCD display to be placed on top of back-lit paper documents (Figure 4). We show how cAR can be applied to the sample application area of active reading [1], leveraging the affordances of paper [12, 14] and digital systems [10, 20, 22, 24]. Finally, we gather user feedback and discuss the technical challenges of cAR devices.

Conference'10, Month 1–2, 2010, City, State, Country.

2. CONTACT AUGMENTED REALITY

Our conceptualization and implementation of cAR is guided by the vision that printed material is related to rich amounts of digital information. Wall posters, newspapers, book pages, are associated with far more content, and in much diverse formats (multimedia), than is possible to etch in ink. With cAR, such associated content can be retrieved by simply placing a transparent-display mobile device directly on top of the object, be it a poster, map, or newspaper. This vision acknowledges the affordances of physical objects; in the case of paper documents the naturalness and convenience of reading, scribbling, folding or manipulating printed paper is unmatched by its digital counterparts [2, 12]. While existing mobile devices already offer access to digital information by means of mobile AR, cAR is based on the notion that interactions with the digital can be enriched by the tangibility of physically moving a device on and off the augmented object.

We envision cAR devices that provide digital content when in direct contact, and behave as normal mobile devices when they are not resting on top of objects. This interaction model has several implications. First, augmentation is triggered by means of implicit interaction: placing the cAR device on top of the object. Once the object is identified, the device launches the application associated with it. Not having to launch the AR app explicitly lowers the time and cognitive effort required to access digital content. Second, cAR does not require the user to hold the device in front or above the object, mitigating the adverse effects of physical effort on the arms [13]. A final implication is that cAR devices augment objects without being the center of attention all the time; acting as ambient or secondary displays, and allowing an undisturbed view of the underlying object due to transparency.

We identify three categories of interaction techniques for cAR devices: First, *contact-based* interactions, e.g. placing the device on a newspaper could retrieve additional data about that object such as audio or video. Second, *off-contact* interactions, e.g. information between devices can be easily exchanged by stacking one on top of another. Third, *content-aware* interactions, e.g. tapping on unknown words triggers a search. Other interactions within these categories include content extraction, scribble triggers, orientation to content, and flipping for dual-side access.

For cAR to operate, the fundamental requirement is to establish a frame of reference between the device and the object to augment (a coordinate system). cAR interactions require knowing the 2D location of the device relative to the origin of the physical object's coordinate system. This implies that, for example, using a cAR device on a book while in bed or while sitting on a table makes no difference when determining their relative locations. An important consequence of the spatial alignment between the transparent display of the cAR device and the augmented object is that digital content is rendered on a virtual plane parallel to the object surface; this means that no homographic transformations are needed for aligning the digital augmentation.

In summary cAR integrates virtual and physical worlds by:

- augmenting physical objects upon contact,
- preserving the affordances of physical objects,
- integrating display and input functionalities, and
- simplifying registration and rendering to two dimensions.

3. RELATED WORK

cAR builds on work in augmented reality, magic lenses and transparent portable devices.

3.1 Augmented Reality

Augmented Reality (AR) enhances the real world by embedding digital content onto it. Bimber and Raskar list three basic AR challenges: display technology, registration and rendering [6]. The display technology determines the complexity of the other two. *Traditional AR* relies on mobile displays carried by the users (e.g. smartphones, pico-projectors, HMDs) allowing the augmentation of any object within the display's field-of-view but requiring complex operations for registration (i.e. 3D object recognition) and rendering (i.e. field-of-view and perspective calculations). 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 [4, 7].

On the other hand, *Spatial AR* (SAR) relies on displays fixed in the environment (e.g. projectors, transparent LCDs) [6]. Knowing the exact location of the display and the augmented object provides SAR applications with *spatial alignment*, a linear correspondence between virtual content and real world objects. Spatial alignment facilitates the creation of AR applications because the registration and rendering operations required are simpler. Nonetheless, preserving spatial alignment requires both the display and augmented object to remain spatially fixed, limiting SAR to non-mobile applications.

Contact Augmented Reality (cAR) incorporates elements from both traditional and spatial AR. From traditional AR it maintains the vision of a mobile device that augments any object and is carried 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 their correspondence. In brief, cAR is both mobile and spatially aligned.

3.2 Magic Lenses and Tangible Views

cAR is inspired by Bier et al.'s Toolglass and Magic Lenses [5]. Toolglass and magic lenses for WIMP interfaces sit between the application and the cursor to provide rich operations and visual filters on the digital content. For example, a toolglass widget can 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 can hide or show details of an underlying digital object by placing the widget on top of it. Moving beyond WIMP, Mackay et al.'s implemented a tool-glass and magic lens approach to augmenting a biology laboratory book [21]. Researchers built physical magic lenses using transparent acrylic plates with fiducial markers [17] and 3D head tracking [28]. Similarly, non-transparent tangible views [30] provide secondary displays for tabletop computers to be used as physical magic lenses or application menus.

cAR, a concept developed for transparent mobiles, encapsulates ideas from toolglasses and magic lenses. With cAR, the physical object is visible and modifications happen on its digital model. cAR advances the concept introduced by the A-book [21] in several ways: we use feature tracking registration, explore off-contact and content-aware interactions, and rely on transparent displays, none of which have been explored before.

3.3 Transparent Handheld Devices

Transparent handheld devices are the subject of popular design concepts covering smartphones and tablets [8, 9]. Such concepts are instrumental in proposing novel interactions (some of them similar to the ones we explore); however they do not discuss usage contexts and technical limitations. While such devices are becoming commercially available (e.g. Lenovo S800), we possess limited understanding of the breadth of interaction techniques they afford. One explored aspect is their support for touch interaction on the back of the device. LucidTouch [34] and LimpiDual [15, 23] studied back-the-device touch to overcome the fat-finger and finger occlusion problems. Lee et al. [18] studied the binocular parallax problem.

Our work moves beyond such problem-oriented foci to include a broad range of techniques for augmenting objects upon contact. Glassified [29], a ruler with an embedded transparent display and a track-pad used to augment paper hand drawings, embodies aspects of the cAR vision. In this paper, we propose a conceptual framework for cAR and present two alternative implementations.

4. cAR INTERACTION TECHNIQUES

Figure 2 shows three categories of cAR interaction techniques: *contact-based, off-contact* and *content-aware*. Contact-based and off-contact interactions are what primarily distinguish cAR from other AR approaches, which focus largely on interacting with digital content (*content-aware)*. Although off-contact techniques resemble other non-AR technologies [30], their cAR versions are performed in relation to the augmented object and on top of it.

4.1 Contact-based

Contact-based interactions are manipulations of the cAR device in relation to the object below.

Placing/Removing – The basic cAR interaction 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, such as maps, the device needs access to a model of the underlying object. Conversely, removing the cAR device is an implicit interaction to change mode or exit the system.

Translation – Horizontal translation of the cAR device in each axis can be interpreted as input. An application can use x/y translation to accommodate a virtual layer accordingly and maintain a correspondence between content 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". For example, rotation could be used to change display settings like opacity and zoom factor.

Freezing – Freezing ignores changes in translation and rotation. When freezing, users can move the device freely while preserving the application state, e.g. the current view in the virtual plane (see Figure 3B). For example, once triggered, video content is reproduced regardless of changes on the device's location.

Shaking – Shaking the device while lying on the augmented object is a gesture to, for example, modify or remove the display content. Shaking can be added as a modifier to other interactions such as translation and *rotation* in order to change their default meaning.

4.2 Off-Contact

Off-contact interaction techniques do not require the cAR device to lie on the augmented object.

Flipping – A cAR device can be flipped around to bring the other side of the screen on top. Upon flipping, the cAR device can apply modifications on the user interface like, for example, zooming, inverse color filters, a translate feature, or launching a secondary application for the actual document.

Stacking – A cAR device can be stacked on top of another one. Given that both displays are transparent, the digital content of both devices and the physical object could be visible at once. This interaction can be used to support content sharing: digital content from one device is pulled up pushed down between devices.

4.3 Content-Aware

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

Direct Pointing – Direct pointing allows users to use their finger or stylus to interact with spatially-aligned digital content, click on user-interface elements such as buttons menus, or issue gestures.

Extraction – Users can interact with elements of the digital model of the augmented object. For example, a cAR magazine app allows users to select and look up word definitions, translations, and their occurrences in the document.

Triggers – Triggers are regions of the physical object that activate special responses by the cAR device. Triggers can be area-based or scribble-based. Area-based triggers are special zones statically defined in the document being augmented, such as an image on a newspaper that triggers associated video. Scribble-based triggers are hand-drawn glyphs on the physical object which are read and interpreted by the cAR device; e.g. a hand-drawn square launches the calculator application by moving the cAR device on top of it.

Anchoring – Anchoring refers to attaching digital content to a fixed location on the physical object. For example, digital hand-written notes can be anchored to paragraphs of a paper book.

Orientation – A cAR application can adjust the orientation of its user-interface based on the coordinate system of the augmented object. This technique resembles adaptation of mobile phone interfaces to the way users hold them (portrait vs. landscape). For example, a cAR application can adjust to the direction of the text and orientation of the paper document.

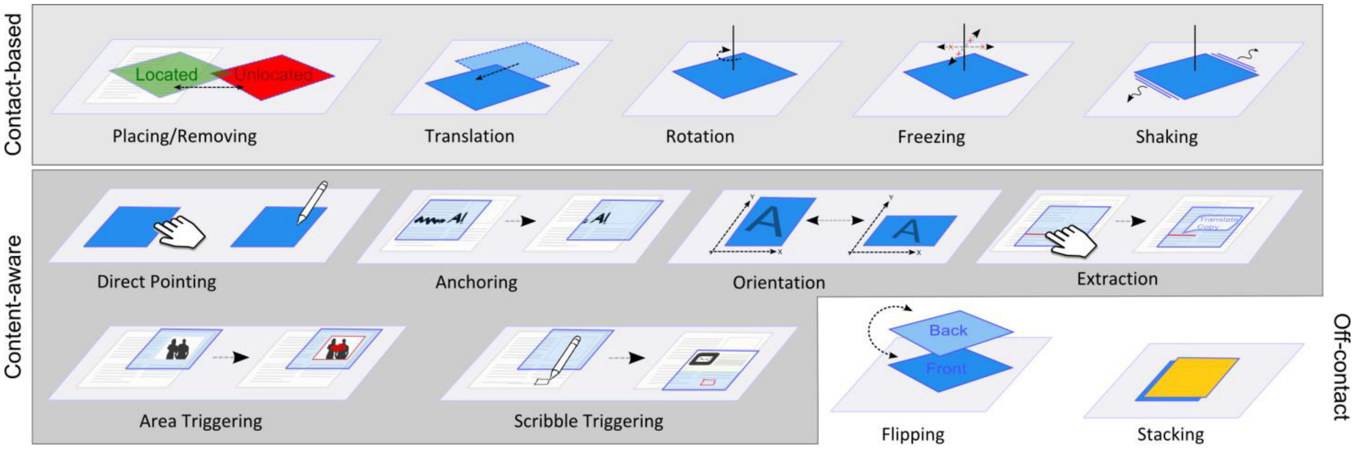

Figure 2. cAR interaction techniques grouped by the three identified categories.

5. SAMPLE APPLICATION: MAPPING cAR TO ACTIVE READING

To demonstrate the novel interactions provided by cAR, we chose to apply our implementation to active reading [1, 2]. Active reading refers to reading to gain knowledge and understanding on a specific topic. Active reading includes reading to self-inform, to cross-reference or to support discussion [2]. Some basic features for active reading include: outlining, underlining, highlighting, searching, scribbling, digital annotations, note-taking, non-sequential navigation, information seeking, quoting, comparing, and content sharing. Note that our goal is not to create a cAR active reading system that outperforms existing systems [10, 20, 22, 24, 27, 33, 35]. Rather, our interest is to use active reading as an example application area to explore a range of cAR techniques.

Figure 1 (front page) shows prototypes and sketches with different active reading features and the supporting interaction techniques. For example, users can add hand-written notes using touch (Figure 1A) or a stylus (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.

Technique	Active Reading Feature	Prot
Placing/Rem.	Document recognition, access, exit	Both
Translation/ Rotation	Browsing virtual content anchored to locations in the document.	Both
Freezing	Ignores translation and rotation, thus maintaining the current digital view.	tPad
Shaking	Undo for highlights and scribbles	tPad
Direct Pointing (hand and pen)	UI interaction, creating and manipulating digital contents	Both
Anchoring	Adds notes and scribbles to fixed locations of the physical document.	Both
Orientation	Adjust the 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/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 between devices.	tPad

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

5.1 Implementation Requirements

cAR interaction techniques and applications, require a hardware and software platforms that fulfill the following requirements:

RQ1 – Registration: a cAR device should identify the object upon which it rests, and determine its position and orientation relative to it. For example, document id, page, x-y location and rotation.

RQ2 – Touch/Pen Input: a cAR device should allow interactions with touch (coarse interactions like pushing buttons) and pen (fine interactions like scribbling text).

RQ3 – Dual Side Interaction: a cAR device can flip over and allow input and output on either side of the screen.

RQ4 – Device Integration: cAR devices should identify and seamlessly integrate with other cAR devices when stacked-up.

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).

6. TABLETOP PROTOTYPE

We built a tabletop prototype to support our design process by allowing fast 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 square that is spatially tracked on the tabletop via fiducial markers (RQ1). Different markers on both sides enable flipping (RQ3). A document viewer shows the document to augment, aligns the created content to the actual page, and extracts the words users tap on for further interaction (RQ5). The UI for the simulated device is shown at the location and orientation of the probe, giving the impression of a translucent display that can be moved freely on top of a document.

6.1 Implementation

We implemented the prototype on a Samsung SUR40 tabletop as seen in Figure 3. We attached Microsoft ByteTags to a 7” acrylic glass probe using IR reflective foil to minimize obtrusiveness.

The prototype supports touch and pen input (IR pen) (Figure 3C-D). We implemented the following features: users can write free-hand annotations or highlight text (Figure 3A), tap on figures to show an overlay with additional information (e.g. video), or tap on references to show the corresponding bibliographic entry. Flipping the display after selecting a word, switches to a web browser showing an online encyclopedia’s entry for the word (Figure 3B). If nothing is selected, a color-inverted view is shown, illustrating different ways of presenting content (Figure 3C).

6.1.1 Limitations

While this prototype is well suited for rapid prototyping it is also limited. The transparent probe lacks its own display which limits the real experience of a transparent device like feeling its actual weight, or facing a transparent display’s problems such as color blending problem and limited color profiles [31]. Moreover, the tabletop cannot accurately simulate the haptics of working with real paper or other physical objects: users cannot grab the paper, move it around or feel its texture. Users are also affected by the bright illumination of the tabletop. Finally, the prototype does not support device integration (*stacking* - RQ4). Nonetheless, the benefits of using such a quick form of prototyping helped us to gather user feedback and improve the design of cAR techniques.

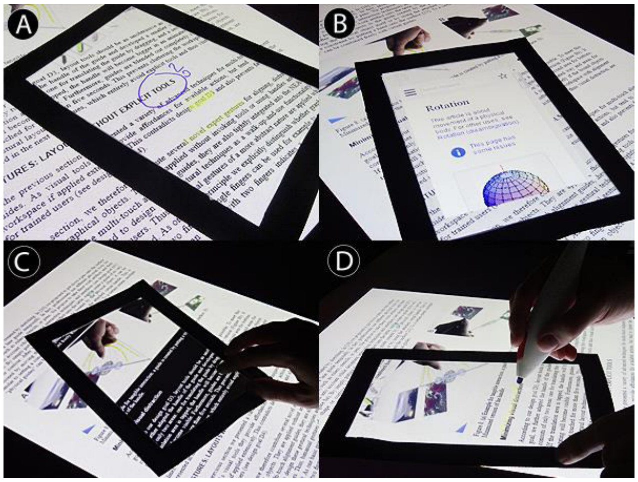

Figure 3. A cAR tabletop prototype: A) highlights and scribbles, B) online-search of selected content, C) inversion lens when flipping without selection, and D) pen input.

6.2 User Feedback

We gathered early user feedback on cAR interactions from eight participants (2 female, 28 years old in average) using the tabletop prototype. After introducing participants to cAR and active reading, a researcher demonstrate the interaction techniques and participants had the chance to try out the system. The session consisted of a semi-structured interview (~30 minutes) where participants could use the prototype while answering questions.

6.2.1 cAR and Interaction Techniques

In general participants appreciated the cAR concept and its usage for active reading. Participants highlighted the value of getting access to information not already included in the text (e.g. video or color images) as well as the benefits of having highlights and annotations in digital format for later use. Some users indicated it would be better suited for books (rather than for short documents) and for situations where a table is available to limit fatigue from holding the device against, e.g., a poster in the wall.

Users easily grasped the value of the *translation*, *rotation*, *direct pointing*, and *anchoring* interactions, and their effects on the contents on the display (e.g. menus) and on the virtual layer (e.g. scribbles and notes). Similarly, they appreciated the possibilities offered by content *extraction*, and suggested other usages like translation and social media sharing. On the other hand, *flipping* received mixed reactions. It was perceived to be laborious by six of the participants. This may have been influenced by the form factor and limitations of the prototype, and further emphasizes the need for careful mapping between interaction techniques and application tasks. Thus, complicated gestures should be used only for complex mode switches or rare tasks.

6.2.2 Active Reading Support

Feedback was mixed for both highlighting and annotating: For three participants, these were the most important features of the prototype, while the others did not see a clear advantage of combining digital annotations with physical, printed documents or preferred techniques similar to PDF readers like word-based marking and comment boxes instead of free-hand marking and scribbling. These opinions might have differed if our tests included maps, as such documents are often marked heavily when used [25]. Two participants mentioned the importance of keeping track of where annotations are, pointing to the need for overviews of the digital content and off-screen markers. Finally, users frequently mentioned the possibility to export such annotations and extracted content to other digital formats, ranging from simple clipboard functionality to integration of some form of social network for sharing comments about specific parts of a document.

Six users especially 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 looking up terms in an online encyclopedia, even before they were shown this feature, thinking of it as “convenient” and “quite cool”. Hence, this feature should be considered essential. Moreover, completely replacing content was also well received. Four participants mentioned zooming text for reading assistance as a useful feature, half of the participants also proposed automatic translation of the text under the device. However, this contradicts the notion of see-through augmentation.

6.2.3 Other Feedback

Participants also mentioned the offset and a possible loss of context when working through the device due to its thickness and border frame. Finally, several users mentioned the flexibility of paper as a problem when combined with the inflexible prototype. This is aggravated when the user holds the device and a paper stack in the same hand. Working on a desk or other surface, like three users suggested, would eliminate this problem.

7. tPad PROTOTYPE

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

Our prototype uses a semi-transparent 7 inch LCD display on top of a light table. The documents to augment are printed single sided on white paper. The light table acted as back light necessary for the LCD-based display. Future transparent displays (e.g. T-OLEDs) do not require such a setup as they emit their own light. We used an overhead camera attached to the display for registration (RQ1). A touch-overlay supports touch and pen input (RQ2), an accelerometer enables flipping (RQ3) and magnetic sensors enable device integration (RQ4). The tPad runtime holds a PDF version of the document and meta-data as object models (RQ5).

Figure 4 shows the tPad prototype (top). The device 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.

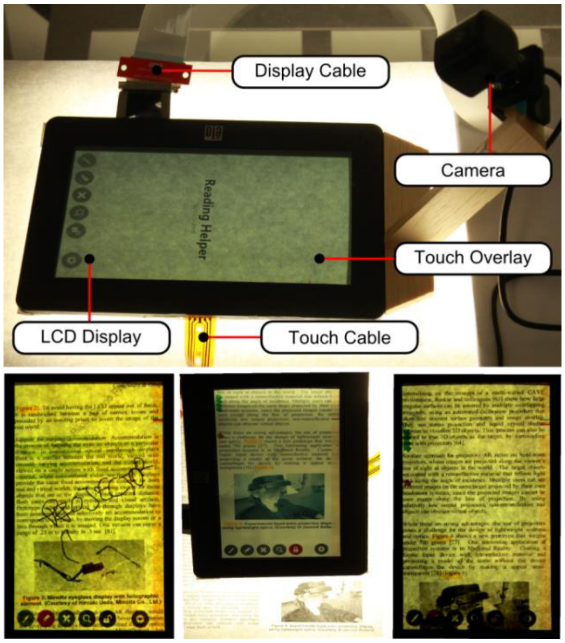

Figure 4. Top – tPad system components. Bottom – tPad at runtime: scribbles, lock-in function and off-screen markers.

The ActiveReader application supports all the features listed in Table 1, except flipping due to the camera protrusion. The tPad includes a soft-keyboard to support text entry, and uses *rotation* to control opacity and zooming. When presented with the settings screen, the user rotates the tPad to control the transparency of the digital content. Users can also zoom into the virtual layer to “create” more space for scribbles. The device uses off-screen markers to indicate the location of off-screen anchored content.

The tPad supports the *orientation* and *freezing* interaction techniques. For orientation the ActiveReader relocates its menus according to the text flow, so that the menus are away from the main reading and interaction area. This strategy reduces the presence of fingers and stylus in the image captured. Reducing the number of fingers, stylus and display content captured by the overhead camera is important because they reduce the success rate of the feature-based registration process (see section 7.1.2). In other functionality users can *freeze* the tPad on a particular location and the actual digital content will remain visible regardless of the device's movements; a user could then move to a different page or pass the device to another person while having the digital content visible at that particular location.

The tPad also supports *stacking* and *triggers*. We implemented stacking via magnetic switches and magnets embedded on the device's frame. When physically stacked, the magnets of the device on top align with the magnetic sensors of the device below, starting 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 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 in the current document created in the device below. *Pull current page* limits the transfer to the current page. *Pull selection* transfers only manually selected content. Stacking finishes either by explicitly selecting the un-pair button, or by physically separating the devices.

The device uses scribble-based triggers to launch specific apps. For example, placing the tPad over a hand-drawn square launches the CalculatorApp. Area-triggers launch predefined content associated to an area in the document. For example, a video is played when the tPad is placed on top of a particular image.

7.1 Implementation

7.1.1 Hardware and Software Architecture

Figure 4-top shows the hardware components used for the tPad prototype. We re-purposed a 7'' inch semi-transparent LCD resistive-touch USB display by removing the backlight. The tPad rests on top the physical papers it augments (one sheet of paper at a time) which in turn rest on a custom-built D65 light table (glass table with fluorescent lights underneath). Display and touch overlay are connected to the original display controller board. We added a Microsoft LifeCam 6000, and an Arduino Pro Micro controller board at 5V, 4 reed-switches and a ADXL335 3-axis accelero-meter. The display controller board, camera and Arduino are all connected to a computer running Windows 7. We use C# and Microsoft WPF for authoring and rendering, and C++ and OpenCV 2.4.3 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[1] for accessing pixel-level location data of the PDF contents.

7.1.2 Camera-based 2D registration

To determine the location and orientation of the tPad relative to the document (i.e. registration) we use the camera attached to the device to capture the device's screen and the underlying surface from above 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 the location of the actual tPad to the printed version of the document. The features, also known as keypoints, efficiently describe image patches and should be 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 [26]. Once keypoints are located, we need a robust and compact keypoint descriptor. We use Fast Retina Keypoint (FREAK), a descriptor inspired by the human retina which is generally faster to compute and more robust to scale, rotation and noise than conventional keypoint descriptors like SIFT, SURF or BRISK [3]. The tPad maintains a database of documents with it augments. The tPad stores the content, graphical representation, and pre-computed FAST-features for each document. Our approach is similar to PACER's [19] with two differences: the camera works from a fixed perspective and the captured-image contains non-matching objects like display content, fingers and reflections.

Since the feature matching algorithm is not invariant to perspective transformation, for each captured image a correction of the perspective distortion is applied by prior to extracting features. If the algorithm finds the corrected captured image in the document, a transformation matrix between both is computed and used to derive location and orientation of the device.

Figure 5 shows the results of our registration algorithm tested at the center of the 10 pages of our sample document [4] at 9 different angles and under three levels of obtrusion: without screen, screen (no content shown) and partly occluded (a finger touching the middle of the display). Results show our matching algorithms work efficiently for most angles, with performance decreasing with the image quality (with screen and partially occluded conditions), particularly for angles between 45 and 90 degrees. Results demonstrate the feasibility of a self-contained camera-based cAR device, noting that further research is needed for different usage conditions and other display technologies.

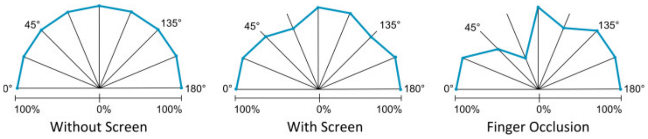

Figure 5. Registration accuracy at different angles and usages.

7.1.3 Technical Limitations

Even though hand-held and mobile, our tPad is limited by the need of a light table. 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 10 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 limited our exploration of stacking as the display on-top did not receive enough backlight.

8. DISCUSSION

Our two prototypes and user evaluation demonstrate how cAR differs from existing approaches to mobile AR. Activated when placing the device on top of that augmented object, cAR breaks apart from the current mobile AR paradigm, where an application has to be invoked explicitly, and uses an always-on paradigm supported by implicit interactions. Our experiments show that users understand and appreciate cAR, particularly in the active reading application scenario we visited. Moreover, our prototypes demonstrate its technical feasibility and highlight the challenges that lie ahead. The rest of this section details the main such challenges for the design and implementation of cAR devices.

[1] https://www.tallcomponents.com/pdfkit4.aspx

8.1 Object Model

A model of the object being augmented (RQ5) is a fundamental piece for cAR because it is the base for 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 distributed to a cAR device. In the case of document-based applications such a model could be made available by, for example, the publisher of the physical document, either as a self-contained cAR application or as a file formatted for a general purpose reader. In this case the content, meta-data, and media files should be bundled by a third party and provisioned to the device. We envision a scenario where the device, upon laying on a document for the first time, tries to locate itself within a list of known documents or, should it fail, delegates the search to a document recognition online service.

In an alternative approach the cAR device creates the model of the document based on images captured. Image-stitching algorithms 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 (text flow detection, extract text from images, etc).

8.2 Camera-based Registration and Optimizations

A major aspect 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. Our feature matching algorithm is sensible to lighting conditions and device orientation, and dependent on 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.

Other registration limitations can be handled by optimizing the feature extraction and matching stage. 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 on the adjacent pages, 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 seconds in the CPU. 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. Finally, descriptors and matchers particularly tailored for text documents can be used [16], together with hardware acceleration via GPU or AR dedicated chipsets such as metaio's AR Engine[2].

8.3 Color Blending

A major usability obstacle when using the tPad prototype was the interference of colors between the paper document and the content on the LCD display, affecting the legibility of the digital content [11, 31]. This phenomenon is known as color blending and also occurs in T-OLED displays. Handling color blending is important for consumer-ready cAR devices and which could be done in several ways. The simplest solution is to re-locate UI elements to areas where color blending is minimal (similar to [32]). However, this solution affects the application usability as users have to keep searching for the buttons and menus. A more elaborate solution relies on concepts of color theory to predict color blending and compensate the displayed color (similar to [31]). Another 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 and the OLED a black one, i.e. nothing.

8.4 Device Ecologies

A recurring topic during the design and user-feedback session was the integration of content created in the cAR device with other tools including other cAR devices (RQ4). For example, if we consider active reading as a creative process, the content from the cAR device should be accessed later for archival or reference. A cAR device should then provide mechanisms for integrating with other computing devices like traditional PCs.

9. CONCLUSIONS

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

Our first low-fidelity prototype uses a tabletop computer and transparent acrylic tangibles. User feedback showed participants understand the cAR concept and interaction techniques, and appreciate the opportunities it offers, particularly *extraction* of content for online sharing, translation and saving. 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. tPad confirmed that a camera-based approach can efficiently identify a text document and determine the device's location in a document. Moreover, the tPad helped us identify the hardware and software elements necessary to support off-contact interaction techniques (*flipping* and *stacking*). Building both prototypes provides the basis to discuss the importance of an object model, registration optimizations, color blending and device ecologies.

Our future work includes building a complete self-contained tPad with improved image capture, batteries, and an embedded-computer without the need of a light table. We will extend our exploration to other application areas like map navigation, music playback, and support for hand-written forms. We also want to explore the multi-level registration, ad-hoc model acquisition, and different model distribution strategies.

10. REFERENCES

1. Adler, M. J. and Van Doren, C. How to Read a Book. Revised edition, Simon and Schuster, New York, 1972.
2. Adler, A., Gujar, A., Harrison, B., O'Hara, K., and Sellen, A. A diary study of work-related reading: design implications for digital reading devices. In Proc. CHI '98.

[2] http://www.metaio.com/products/

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. 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.
8. Concept Phones. (2013, March 17). Transparent Phones. [Website]. Retrieved from http://www.concept-phones. com/tag/transparent-phone/.
9. 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.
10. Dachselt, R. and Al-Saiegh, S. Interacting with printed books using digital pens and smart mobile projection. In Proc. of the Workshop on Mobile and Personal Projection (MP2) @ ACM CHI 2011, 2011.
11. 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. of VR '10. IEEE.
12. Golovchinsky, G. Reading in the office. In Proc. 2008 ACM workshop on Research advances in large digital book repositories, BooksOnline '08, pp 21-24
13. Hincapié-Ramos, J.D., Guo, X., Moghadasian, P., and Irani, P. 2014. Consumed Endurance: A Metric to Quantify Arm Fatigue of Mid-Air Interactions. In Proc. CHI 2014. ACM. *To Appear.*
14. Hinckley, K., Bi, X., Pahud, M. and Buxton, B. Informal information gathering techniques for active reading. In Proc. CHI '12. 2012. ACM.
15. 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.
16. Iwamura, M., Nakai, T., and Kise, K. 2007. Improvement of Retrieval Speed and Required Amount of Memory for Geometric Hashing by Combining Local Invariants. In Proc. BMVC '07.
17. Kim, K., and Elmqvist, N. 2012. Embodied lenses for collaborative visual queries on tabletop displays. Information Visualization, 11(4), 319-338.
18. Lee, J.H., Bae, S.H., Jung, J., and Choi, H. 2012. Transparent display interaction without binocular parallax. In Adjunct Proc. UIST '12. ACM.
19. Liao, C., Liu, Q., Liew, B., and Wilcox, L. 2010. Pacer: fine-grained interactive paper via camera-touch hybrid gestures on a cell phone. In Proc. CHI '10. ACM.
20. 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.
21. 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.
22. Matulic, F., and Norrie, M. C. Supporting active reading on pen and touch-operated tabletops. In Proc. AVI '12. 2012. ACM.
23. 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.
24. Price, M. N., Schilit, B. N. and Golovchinsky, G. 1998. Xlibris: the active reading machine. In Proc. CHI 98.
25. Reitmayr, G., Eade, E., and Drummond, T. 2005. Localisation and Interaction for Augmented Maps. In Proc. ISMAR '05. IEEE.
26. 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
27. Schilit, B., Golovchinsky, G., and Price, M. Beyond paper: supporting active reading with free form digital ink annotations. In Proc. CHI '98, pp 249-256.
28. Schmalstieg, D., Encarnação, L. M., and Szalavári, Z. 1999. Using transparent props for interaction with the virtual table. In Proc. I3D '99. ACM.
29. Sharma, A., Liu, L., and Maes, P. 2013. Glassified: An Augmented Ruler based on a Transparent Display for Real-time Interactions with Paper. In Proc. UIST '13 Demos.
30. Spindler, M., Tominski, C., Schumann, H., and Dachselt, R. 2010. Tangible views for information visualization. In Proc. ITS '10. ACM.
31. Sridharan, S.K., Hincapié-Ramos, J.D., Flatla, D., and Irani, P. 2013. Color Correction for Optical See-Through Displays Using Display Color Profiles. In Proc. VRST '13. ACM.
32. Tanaka, K., Kishino, Y., Miyamae, M., Terada, M., 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.
33. Tashman, C. S. and Edwards, W. K. Liquidtext: a flexible, multitouch environment to support active reading. In Proc. CHI '11. 2011. ACM.
34. Wigdor, D., Forlines, C., Baudisch, P., Barnwell, J., and Shen C. 2007. Lucid touch: a see-through mobile device. In Proc. UIST '07. ACM.
35. 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.