PaperComp 2010

1st International Workshop on Paper Computing

26 September 2010, Copenhagen, Denmark
a Ubicomp2010 workshop

ORGANIZING COMMITTEE

Frederic Kaplan, EPFL CRAFT (Chair)
Patrick Jermann, EPFL CRAFT (Chair)
Quentin Bonnard, EPFL CRAFT
Andrea Mazzei, EPFL CRAFT
Guillaume Zufferey, EPFL CRAFT

PROGRAMME COMMITTEE

Leah Buechley, MIT Media Lab
Enrico Costanza, University of Southampton
Denis Lalanne, Université de Fribourg
Vincent Lepetit, EPFL
Chunyuan Liao, FX Palo Alto Laboratory
Xu Liu, Ricoh California Research Center
Les Nelson, Xerox PARC
Moira Norrie, ETHZ GlobIS
Beat Signer, Vrije Universiteit Brussel
Jürgen Steimle, TU Darmstadt
Roel Vertegaal, Human Media Lab, Queen's University
Pierre Wellner, SpiderPhone
Janet C. Read, University of Central Lancashire

CONTACT

papercomp2010@papercomp.org
Twitter: @papercomp

http://doc.hn/7vrr

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

PaperComp 2010

1st International Workshop on Paper Computing

26 September 2010, Copenhagen, Denmark

a Ubicomp2010 workshop

Preliminary program

(9h00) Introduction to the PaperComp 2010
Frederic Kaplan and Patrick Jermann

(09h30) Direct Latitude/Longitude Identification of Paper Maps Using a Camera Phone
Genta Suzuki, Nobuyasu Yamaguchi, Mineki Takechi

(09h55) Extraction and Classification of Handwritten Annotations
Andrea Mazzei, Frederic Kaplan, Pierre Dillenbourg

(10h20) MobARDoc: Mobile Augmented Printed Documents
Tom Holland, Aaron Quigley

(10h25) Towards Understanding Erasing-based Interactions: Adding Erasing Capabilities to Anoto Pens
Simon Olberding, Jürgen Steimle

(10h30) Improving paper books: searchable books
Ming Ki Chong, Fahim Kawsar

10h35 - 11h00 Coffee Break

(11h00) What do U-Note? An Augmented Note Taking System for the Classroom
Thomas Pietrzak, Sylvain Malacria, Aurelien Tabard, Eric Lecolinet

(11h25) A Tabletop System for supporting Paper Prototyping of Mobile Interfaces
Benjamin Bähr, Sven Kratz, Michael Rohs

(11h50) Using Tangible Symbols with people with profound learning disabilities to access computer based media
NickWeldin, Gosia Kwiatkowska, Karen Bunning

(11h55) Adapting the TinkerSheet Augmented Papers to Facilitate Student's Reflection
Son Do-Lenh, Patrick Jermann, Guillaume Zufferey, and Pierre Dillenbourg

(12h00) A Paper Interface for Code Exploration
Quentin Bonnard Frederic Kaplan Pierre Dillenbourg

(12h05) Design of a Modular Architecture for Integrating Paper and Digital Document Management Systems
Matthew Jervis, Masood Masoodian

12h10 - 14h00 Lunch Break

(14h00) Storybook Augmentation through Portable Projectors, 3D Augmented Reality and Anaglyph
Fadi Chehimi

(14h05) ARcetate: Augmented Reality with Acetate Paper
Nicolas Nova

(14h10) Cloth-based Interfaces: Designing for Interactions with Textile Displays
Julian Lepinski, Roel Vertegaal

(14h15) Les éditions volumiques : Experimenting with Computational Paper
Bertrand Duplat, Etienne Mineur

(14h20) The NeverEndingStorytellingMachine: A Platform For Creative Collaboration Using a Sketchbook and Everyday Objects
Edwina Portocarrero, David Robert, Sean Follmer, Michelle Chung

(14h45) Interactive Paper: Past, Present and Future
Beat Signer, Moira C. Norrie

15h10-16h30 **Poster/Demo Session**

16h30-18h00 **Discussion: The Future of Paper Computing**

PaperComp 2010: First International Workshop on Paper Computing

Frédéric Kaplan
EPFL CRAFT, RLC,
Station 20, 1015 Lausanne,
Switzerland
frederic.kaplan@epfl.ch

Patrick Jermann
EPFL CRAFT, RLC,
Station 20, 1015 Lausanne,
Switzerland
patrick.jermann@epfl.ch

ABSTRACT

Paper is not dead. Despite the progress of e-ink screens, smartphones and tablet interfaces, printed paper stays a convenient, versatile and familiar support for reading and writing. Books, magazines and other printed materials can now be connected to the digital world, enriched with additional content and even transformed into interactive interfaces. Conversely, some of the screen-based interfaces we currently use to interact with digital data could benefit from being paper-based or make use of specially designed material as light and flexible as paper. Far from a paperless world, printed documents could become ubiquitous interfaces in our everyday interaction with digital information. This is the dawn of paper computing.

Author Keywords
Paper Computing, Paper-based Interfaces, Paper-like Interfaces

ACM Classification Keywords
H.5.2 User Interfaces (D.2.2, H.1.2, I.3.6) - Input devices and strategies (e.g., mouse, touchscreen)

General Terms
Design, Human Factors, Theory.

INTRODUCTION

Despite the recurrent prediction of a paperless future, paper documents are still widely used in our everyday activities. Various studies confirm paper has inherent advantage over electronic platforms [1,9,14,17,19,21]. Several studies confirm that reading is easier on paper. Paper is light, cheap, foldable, easy to use and ubiquitous in our society. Paper sheets can be easily organized to create large spatial layouts. They can easily be shared and exchanged. Pen-based drawing and annotation on paper is intuitive and flexible. For this reason, paper still plays a crucial role in many informal and creative activities (sketching, brainstorming, etc.). These fundamental characteristics justify the relevance of research projects that explore the role of paper documents as interfaces to our digital world. We call this emerging field: **Paper Computing**.

Although he didn't use the term, the first clear vision of the potential of **Paper Computing** was certainly Pierre Wellner's Digital Desk [21]. This 20 year old seminal system demonstrated a set of activities seamlessly integrating digital and paper documents, transforming the metaphorical desktop introduced in the Graphical User Interface paradigm into a real physical one. In this set-up, printed documents and paper sheets were tracked by an overhead camera system. A calibrated projector could then "augment" these printed documents by additional digital information. Wellner's envisioning opened several research avenues that revealed many difficult underlying technical challenges. Since the DigitalDesk, researchers have continued investigating how new interactive technology can enrich our existing working practices with paper. Their works, reviewed in the next section, support the idea that despite the progress of ebook reader and table interfaces, printed paper with some form of digital enhancement will continue to be an efficient medium to read and to write.

But paper can be more than a reading/writing medium. It can invade domains where screen-based interactivity was until recently the sole and only conceivable mode of interaction. Roughly in the same years when Wellner was investigating various scenarios to enrich our practices with paper document, Johnson and his coworkers also from Xerox PARC explored how paper sheets could be used as a new kind of user interface to interact with a computer [14]. Indeed, paper interfaces permit tangible and direct manipulation and offer the possibility to blend input and output in a uniform space: the paper sheet. Based on these intrinsic qualities, they could offer a cheap and flexible implementation of Weiser's now classical trinity of Ubiquitous Computing interfaces [22] : Boards (yard-scale interfaces like a blackboard), Pads (foot-scale interface like a paperpad) and Tabs (inch-scale like a Post-it note). This second research line holds one of most fascinating direction of paper computing: replacing screen-based interaction by

UbiComp'10, September 26–29, 2010, Copenhagen, Denmark.
Copyright 2010 ACM 978-1-4503-0283-8/10/09...$10.00.

paper. Among the researchers that explore this direction, some explore also the potential of paper-like interfaces based on flexible display technology or made out of paper embedding electronic wires and components: Paper sheets of a new kind.

These parallel research lines share a common view: Far from a paperless world, printed documents are likely to become ubiquitous interfaces in our everyday access to digital information. This is the dawn of Paper Computing.

The following paragraphs describe briefly the main research directions explored so far in this emerging field.

DIGITALLY ENHANCING PAPER DOCUMENTS

Linking paper documents with digital counterparts

Digital documents have their own intrinsic advantage. They can be animated, interactive and hyperlinked. They are easy to archive, search, duplicate and share. Linking physical printed documents with their digital counterparts permits to benefit from their mutual advantages. If such a link could be made readers of a printed document could access additional resources (e.g. attached multimedia and interactive content, other linked documents, reading history of the document), develop new kind of collaborative reading/writing activities (e.g sharing annotations) and more generally integrate their reading activities in the rest of their digital life (e.g. archiving/searching their own reading activity, informing other readers about what they read, etc.). To build such a bridge between paper documents and their digital counterparts various approaches are currently explored.

Documents with printed visual markers

Barcodes and fiducial markers (1D or 2D) such as QR Codes or Datamatrix provide a simple and robust way to link printed documents with digital data. Such markers can be read by optical scanners, like a smartphone or a camera. “Watermarking” approaches have been developed to embed other kinds of identifying information in the structure of the document in a way which is invisible or hardly perceivable to the human reader. The main disadvantage of this family of approaches is that the printed documents must be produced with such kind of application in mind and that it cannot be easily extended to the large amount of paper documents already in use.

Computer vision approach to document recognition

Some researchers try to tackle the challenging issue of identifying and recognizing paper documents solely based on their “natural” visual features. The challenges are somewhat different depending on the way the document is captured, for instance either scanned by a tabletop scanner, or perceived through the camera of a mobile phone (6). Research efforts have been focusing on various subchallenges of this complex task like identifying a paper document on a desk or efficiently comparing its visual signature to a database of other documents. For each of those, various methods have been investigated ranging from Optical Character Recognition methods to geometrical visual pattern matching [2,7,11,16,20].

Dedicated devices to digitally annotate physical paper documents

Another line of research consists in designing devices capable of inscribing digital notes to physical printed sheets. The commercial Anoto technology is one of most widely used dedicated device for interacting with real paper sheets (e.g. (8)). These pens that can be used exactly like ordinary pens are equipped with a built-in camera. The camera is used to decode the position of the pen by scanning a printed dot pattern directly encoded in specially printed paper sheets. The positional information is transfered in real time to a computer via Bluetooth or USB. Thus the Anoto pen can signal precisely its position and path both inside a given document part of a larger document space, provided that these documents are printed on the special paper. Alternative dedicated solutions include pen tracking with ultra-sound [15].

USING PAPER AS USER INTERFACES (PUI)

Paper-based interfaces

The research lines we discussed so far use paper as documents with some form of digital enhancement. A different approach consists of considering paper boards, sheets and cards as interfaces to computer applications. Affordances of paper permit intuitive and effective interactions. For instance, index cards can be used to control slides [12], facilitating their manipulation, on-the-fly reordering and enabling new forms of social exchanges. In the same way, Paper sheets can replace the “windows” of a Graphical User Interface [13]. Such paper interactive displays can be flipped, rubbed, stapled, spatially reorganized: a whole new vocabulary of interactive actions. Paper interfaces can be used in tabletop configuration with overhead projection but also on situations of mobility using wearable projectors [10]. Last but not least, paper interfaces can be cut, photocopied, faxed, organized in folders and books.

Paper-like interfaces

Paper-like interfaces do not use standard paper sheets but specially designed material which should ideally be as light and flexible as paper. One route to this goal is the one of flexible displays [5]. Despite the rapid progress in this area, many challenges remain to match the intrinsic quality of paper sheets. For this reason, several researchers are currently exploring another approach: embedding electronic components into paper structure [4,18]. Using in particular different types of inks and paints (conductive or non-conductive) it is possible to innerve papers with “wires” linking different computational elements (microcontrollers, speakers, leds, etc.). Such a kind of “paper computer” can be programmed to react to particular inputs and to create various textural effects on the paper surface instead of a screen display. This opens many new avenues for designing interactive paper products.

Cloud computing services associated with paper interfaces

The rise of paper interfaces is only made possible because of the parallel development of cloud computing architectures, enabling ubiquitous access to data and services. Although the importance of the server architecture associated with paper interfaces was already emphasized in early research in this domain [14], the massive and global shift currently happening in computer infrastructures [3] opens the possibility that such kind of interface could start to be used "in the wild", beyond controlled lab experiments .

THE EMERGENCE OF A NEW FIELD

PaperComp 2010 brings together researchers exploring the future of paper interfaces and associated practices. Submissions should cover all aspects of paper computing including its core technologies, the new interaction paradigm it enables and societal implications of these developments. Given the potential importance of this new research direction, we believe this emerging community of researchers will continue to grow, attracting researchers from neighboring fields to explore yet to be unveiled potential of paper computing.

REFERENCES

1. Adler, A. et al., A diary study of work-related reading: design implications for digital reading devices. In *Proc. CHI '98,* ACM Press (1998), p. 241-248.
2. Behera, A. Lalanne, D. and Ingold, R. Visual signature based identification of low-resolution document images, In *DocEng '04: Proc. of the 2004 ACM symposium on Document engineering*. ACM Press (2004), 178–187.
3. Carr, N. *The Big Switch: Rewiring the World, from Edison to Google*, W.W: Norton, USA, 2008
4. Coelho, M. et al. Pulp-Based Computing: A Framework for Building Computers Out of Paper. In *Proc. Ubicomp 2007*, ACM Press (2007).
5. Crawford, G. Flexible flat panel displays, *Wiley-SID Series in Display Technology* (2005).
6. Doermann, D. The indexing and retrieval of document images: A survey, *Computer Vision and Image Understanding*, 70 (1998), 287–298.
7. Erol, B. Ant´unez, E. and Hull, J. J. Hotpaper: multimedia interactionwith paper using mobile phones. In *MM '08: Proc. of the 16th ACM international conference on Multimedia*. ACM Press (2008), 399–408.
8. Guimbretière, F. Paper augmented digital documents. *Proc. UIST 03* (2003), 51-60.
9. Mackay,W. et al. The Missing Link: Augmenting Biology Laboratory Notebooks. In *Proc UIST 2002*, 41-50.
10. Mistry, P. Maes. P. SixthSense – A Wearable Gestural Interface. In the *Proc. of SIGGRAPH Asia 2009*, 2009
11. Nakai, T. Kise, K. and Iwamura, M. Use of affine invariants in locally likely arrangement hashing for camera-based document image retrieval, *Document Analysis Systems VII, vol. 3872* (2006), 541–552.
12. Nelson, L. Ichimura, S. Pedersen E. and Adams L. Palette: a paper interface for giving presentation. In *Proc. CHI 1999*, ACM Press (1999), 354-361
13. Holman, D. Vertegaal, R. Altosaar, M. Troje N. and Johns, D. Paper windows : interaction techniques for digital paper. *In Proc. of the SIGCHI conference on Human Factors in Computing Systems*, ACM Press (2005) 591-599.
14. Johnson, W. Jellinek, H. Klotz, J. Rao R.and Card S.K., Bridging the paper and lecornic worlds: the paper user interface In *Proc. CHI 93*, ACM Press (1993) 507-512.
15. Lai, W. Chao, P. and Chen, G. The Interactive Multimedia Textbook: Using A Digital Pen to Support Learning for Computer Programming. In *Proc. ICALT'07*, (2007).
16. Liu, X. and Doermann, D. Mobile retriever: Access to digital documents from their physical source, *International Journal on Document Analysis and Recognition*, 2008.
17. O'Hara K. and Sellen, A. A comparison of reading paper and on-line documents, In *Proc. CHI '97*, ACM Press (1997), 335-342.
18. Qi, J. and Buechley, L. Electronic popables: exploring paper-based computing through an interactive pop-up book. In *Proc. TEI'10*, ACM Press (2010), 121-128.
19. Sellen A.J. and Harper, R.H. *The Myth of the Paperless Office*, MIT Press, Cambridge, MA, USA, 2003.
20. Tan, C. L. Huang, W. Yu, Z. and Xu, Y. Imaged document text retrieval without ocr, *IEEE Trans. Pattern Anal. Mach. Intell.*, 24, 6, (2002) 838–844
21. Wellner, P. Interacting with paper on the DigitalDesk, *Communications of the ACM* (1993), 87-96.
22. Weiser, M. The computer of the 21st century, *Scientific American* (1991) 94-104

Direct Latitude/Longitude Identification of Paper Maps Using a Camera Phone

Genta Suzuki[1] Nobuyasu Yamaguchi[1] Mineki Takechi[2]
[1]FUJITSU LABORATORIES LTD. [2]FUJITSU LIMITED
Kanagawa, Japan Tokyo, Japan
{suzuki.genta, nobuyasu, takechi.mineki}@jp.fujitsu.com

ABSTRACT
In this paper we propose a novel scalable technique to embed geographical coordinates into paper maps and directly identify them using a camera phone. Existing paper map recognition systems match a camera image to stored map data. This tends to result in failed matching where the number of stored maps is huge. With this problem in mind, we developed methods to embed latitude/longitude geographic coordinates into paper maps invisibly and recognize the coordinates directly without having to match them with stored data. Yellow shaded patterns are used for embedding because despite being hardly visible, they are still printable by many printers and capturable on camera. The implementation provides real-time recognition via print maps and commercial mobile phones. In addition, from a simulation incorporating over 18,000 map images featuring various cities, scales, and designs, we confirmed our recognition method can identify the correct location without false recognition.

Author Keywords
Paper maps, mobile phone, georeference, steganography, camera recognition, image processing.

ACM Classification Keywords
H5.2. Multimedia information systems: Artificial, augmented, and virtual realities, hypertext navigation and maps

INTRODUCTION
Mobile mapping applications interacting with Geographic Information Systems (GIS) are commonly used nowadays for navigation, searching and viewing environmental representations of locations such as Google Maps for Mobile. In these applications, users can view personalized and dynamic information in a georeferenced manner, whereby the information is overlaid on digital maps.

However, the readability and accessibility of digital maps on mobile displays are lower than paper maps because the displays are compact and low resolution [4]. To deal with this, mobile systems which interact with paper maps have been proposed [3, 4, 6, 7, 9]. These systems recognize the location information of paper maps via physical input devices.

When it comes to improving the systems and popularizing the stage for practical use, the recognition systems of paper maps must meet the following requirements:

– **Using a mobile phone and paper maps:** It is preferable to use mobile phones as mobile devices because they are in such widespread use nowadays. Additionally, it is desirable that paper maps for the systems not only include commercial maps on books, etc., but also user-printed maps.

– **Providing scalable recognition:** The recognition systems would operate for many maps in practical use. The scalability of recognition means the system can provide accurate and prompt identification of map locations, even if there are huge numbers of maps with different scales and designs.

The contribution of this paper is in providing a map recognition system which meets both requirements. We have developed a system which embeds latitude/longitude geographic coordinates into paper maps invisibly and a map recognition system which detects the coordinates by a camera phone. The data embedding method uses a yellow shading pattern which is hardly discernible to the human eye, although printable by most existing printers. Camera phones can specify the geographic coordinates directly by decoding the shading pattern. The system facilitates georeference via mobile interactive applications with paper maps, which inter-translates between screen coordinates and latitude/longitude geographic coordinates.

The rest of the paper is organized as follows. The next section discusses related work, while the third section shows the concepts and algorithms of our system, followed by the implementation of the system and sample application in section four. Based on the results and performances of the implementation, we discuss the advantages and disadvantage of the system in the fifth section. Finally, the last section provides conclusions and future works.

UbiComp'10, September 26–29, 2010, Copenhagen, Denmark.
ACM 978-1-60558-843-8/10/09.

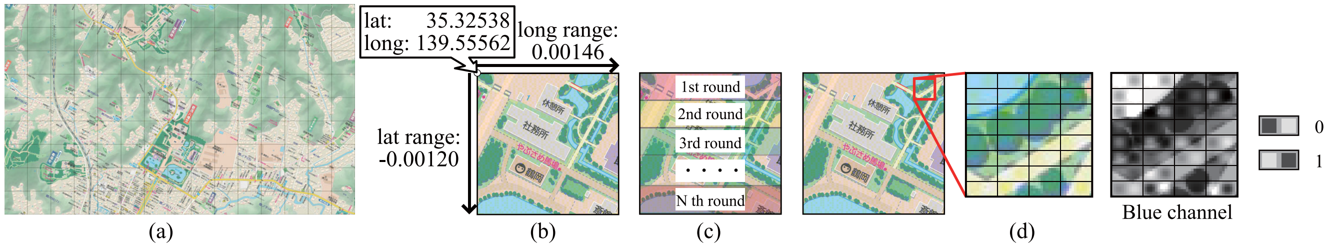


Figure 1. Latitude/longitude embedding method, (a) a map with latitudinal/longitudinal lines, (b) embedding data: latitude/longitude of top-left corner and ranges to other corners, (c) multiple embedding of data bit-strings, (d) embedding method using yellow shading.
Maps: Cartographic data Shobunsha Publications,inc.

RELATED WORK

There have been several approaches involving the recognition of paper maps using mobile devices. However, no systems currently meet both the aforementioned requirements.

The Magic Lens [7] tracks black dots composing grids and searches for the same image of the grids from a stored image patch. In [9], the map location is retrieved from a map database by analyzing the positional relationship of intersections. Although this research meets the hardware requirement, there is no mention of the scalability of recognition, hence it seems hard to guarantee the uniqueness of each pattern for all maps. Additionally, an increase in the stored data would reduce their recognition rate. With speed recognition in mind, if data are stored on a network server, it takes data transfer time.

Reilly et al. [6] use paper maps embedded with RFID (Radio Frequency Identification) tags and a mobile device with attached RFID reader to realize touch based interaction. Norrie et al. [3] adopt paper maps on which infrared-absorbing dots are printed invisibly and special digital pens incorporating a camera. The system retrieves the position of the map by pointing the pen. These works meet scalable recognition requirements because unique IDs can be assigned for each map. However, they use special reader device and special maps.

DIRECT LATITUDE/LONGITUDE IDENTIFICATION

Concept

The basic idea for the requirements involves embedding geographic coordinates invisibly into printed paper maps and recognizing them from the captured image of a camera phone. Our earlier work [2] and its commercial service called *FPcode* (Fine Picture code) [1] have successfully involved embedding 12 digit numerical codes into printed color images using yellow shading patterns. The method exploits the fact that while yellow shading is hardly recognized by the human eye, cameras remain sensitive to the same. Additionally, such shading can be printed by many types of existing color printers.

We aim to apply the method to paper maps and expand to embed latitude/longitude geographic coordinates, which are used in many GIS and GPS devices. This realizes direct latitude/longitude recognition without stored data and georeference of positions on the camera image. Applications can use this georeference to overlay contents such as text information, icons, and photos on the camera preview.

Embedding Latitude/Longitude into Paper Maps

The latitude/longitude embedding system for paper maps performs the following steps:

STEP 1) Dividing maps into grids

First, the given map image is divided into grids by black latitude and longitude lines (see Figure 1 (a)).

STEP 2) Generating data bit-strings

Next, data bit-strings, which include location information, are generated for each grid. Figure 1(b) shows 4 values of the location information: the latitude of the grid's top-left corner, the longitude of the same, the latitude range from top-left to bottom-left corner, and longitude range from the top-left corner to the top-right corner. These values are expressed as signed decimal degrees and to five decimal places (0.00001 is accurate to 1.11 meters at the equator) on a large scale map. After changing these values to binary-coded form, Bose-Chaudhuri-Hocquengham (BCH) error correction codes are encoded from the codes and added to the data bit-strings.

STEP 3) Embedding the data bit-strings into a map grid

The third step embeds the data bit-strings N times repeatedly into a blue channel of the map grid image. Figure 1(c) depicts the repetitive embedding. The blue channel complements that of yellow in the RGB color model. The system divides the grid into blocks and makes pairs of each two adjacent blocks. Subsequently, each data bit is assigned to each block pair. Finally, the gradation level of each block pair is sampled and changed to show the data bit, 0 (if left < right) or 1(if left > right), as shown in Figure 1(d). If the difference of the original gradation level exceeds a predefined threshold, the level will remain unchanged in order to retain image quality.

STEP 4) Printing map

Finally, a map image that is reconstructed by data embedded grids is printed using a color printer.

Recognizing Latitude/Longitude from Camera Images

The decoder works on camera phones and provides direct recognition of embedded latitude/longitude geographic coordinates from a camera preview image. The following show the decoding steps:

STEP 1) Detecting the map grid

In the first step, the decoder detects a grid from a camera image, while the Hough transform is used to detect black lines. If four lines are determined, the screen positions of the four vertices are calculated as grid corners.

STEP 2) Determining bit-strings by comparing blue levels

The second step acquires bit-strings from the grid. The decoder computes the screen positions of all embedded blocks from the grid corners and makes a list of blue levels for the same. Then, for each block pair, 0 (if left < right) or 1(if left > right) is specified by comparing the blue levels.

STEP 3) Determining data bit-strings by a majority

The decoder divides the bit-strings specified in the previous step into N parts and a majority vote of the same for each bit decides the data bit-strings in the third step.

STEP 4) Error detection and correction of the data bit-strings

This step attempts error-correction of the data bit-strings and finally acquires the embedded data. Errors in the data bit-strings are detected and corrected by decoding the BCH codes included in the data bit-strings. If the error correction succeeds, the original data, namely the latitudes and longitudes of the top-left corner, the latitude range from top-left to bottom-left corner, and the longitude range from the top-left corner to the top-right corner are decoded. From the results, the latitude/longitude geographic coordinates for all corners of the grid are decided. Where the error-correcting fails, the decoder reverts to the second step and tries to acquire data from another direction.

Georeference of Screen Positions

After decoding the embedded data, the decoder initializes API for georeference of screen positions from four pairs of the (x, y) and (latitude, longitude) of the grid corners. Homographic transformation is used for inter-translating between screen positions and latitudes/longitudes. In the translation, latitudes and longitudes are converted to screen positions with an accuracy of within 1 pixel. The scope of translation goes from screen positions to geographic coordinates, and the granularities are varied by the ranges of latitudes and longitudes embedded in the grid. The coordinates are calculated to be 1 digit smaller than the ranges. Therefore, if a large scale map is used, the latitude/longitude is determined to an accuracy of within 0.11 meter at the equator from the API.

IMPLEMENTATION

The Paper Map and Camera Phone

We develop the implementation using a Kamakura city map and the T004, which is a camera phone sold in Japan (see Figure 2(a)). The map is printed in A3 size using an EPSON PM-G4500 inkjet printer. The platform of the T004 is Qualcomm BREW [5] and its CPU is Snapdragon QSD8650 1GHz. The size of the camera preview image is VGA (480x640).

Application

We have implemented a latitude/longitude indicator and a contents browser to demonstrate our recognition system. Both applications overlay information on the real-time camera preview.

Latitude/longitude indicator for paper maps

The first application shows the latitude/longitude coordinates of a point on the screen. From every frame during the camera preview, the application fetches the latitude/longitude coordinates corresponding to the middle of the image by georeference. Figure 2(b) presents a snapshot of the application, with the green numbers denoting the latitude/longitude of a location marked by the green cross. In addition to displaying the coordinates, the application facilitates their transfer to other applications. When a user presses an “HP” button, the application is interrupted and an Internet browser shows the location on Google Maps.

Another useful aspect of this application is the ability to set a destination for navigation systems, such as in-car and personal navigation devices.

Contents browser for paper maps

The contents browser overlays photos and text information on areas focused on in the paper map. When the camera preview starts and the decoder recognizes the geographic coordinates of the map, the edge positions of the screen are translated into geographic coordinates. In the VGA size camera preview, (0,0), (480,0), (0,640), and (480,640) positions are translated. Subsequently, all data tagged in the geographic spatial are retrieved from GIS. Finally, the data are superimposed on the screen. Figure 2(c) shows the snapshot. Photos of Kamakura city pop up at the point where the photos are geotagged.

This application can be used to interact with GIS such as by searching shops and viewing sensor data. As for paper maps, it seems applicable for guide maps for tourist sites and amusement parks, exhibition floor plans, and so on.

DISCUSSION

Regarding the requirement for paper maps, we have evaluated that our embedding method can use a number of pairs of printings and papers, including commercial offset printing used for guidebooks and home inkjet printing. As for camera phones, we have confirmed that more than 300

(a) (b) (c)

Figure 2. Snapshots of the implementation, (a) the paper map and the mobile phone, (b) the latitude/longitude indicator shows the latitude/longitude coordinates (green number) of the center cross, (c) photos pop up over the camera image.

camera phones can decode FPcode that uses a uniform method to embed 12-digit numerical data. Based on the results, it is expected that a number of camera phones will be able to capture images, the image quality of which is sufficient to render them recognizable.

Towards the scalability requirement, our approach can distinguish maps where the drawing patterns are similar or the same but the locations differ (e.g. solid color grids of the sea). Concerning failed recognitions of embedded data, we have confirmed that there no false recognition occurred in the simulation. During the latter, attempts were made to decode 18,720 map grids from four directions (top, bottom, left and right). The breakdown of the grids, 104 grids (13x8) per map and a total of 180 maps featuring two cites (Tokyo and Hakone), 9 scale (from 1/500,000 to 1/1,000), and 10 design style are used.

The time spent on decoding is about 72ms in the implementation, while 58ms is required to detect the map grid, and 14ms for the rest. Although there is still room for improvement in the detection time, it seems faster than the time required to retrieve stored data from a network server because the decoder works on a standalone basis.

Our approach of direct latitude/longitude identification has the advantage that applications can operate with the same behavior for a number of maps, which show the same location in different designs by a different publisher. Additionally, the use of the reserved region, which is also included in the data bit-strings, makes the maps stand out. The reserved region is also used to represent the floor numbers of buildings.

One constraint of our system is the fact that pixel sizes of a target grid should retain a certain size in a captured image to ensure recognition succeeds. A minimum of around 200x200 pixels are needed to realize consistent performance. Since grids are about 3cm square, the maximum range of view is 7.2x9.6cm when the size of the camera image is VGA. This means our system is inadequate when attempting to photograph a whole map from separate points as with PhotoMap [8]. On the other hand, our system does lend itself to applications where the interaction model is "a magnifying glass", namely where a camera phone is used for pointing on a paper map.

CONCLUSION AND FUTURE WORK

In this paper we present a mobile recognition system for paper maps using yellow latitude/longitude embedding patterns. The system provides direct, real-time, and scalable latitude/longitude identification of paper maps.

As future work, we intend to develop the system for other mobile platforms such as Android and iPhone OS. We also aim to adopt it for maps on public display, since they can show our embedding patterns.

REFERENCES

1. FPcode Service. http://jp.fujitsu.com/solutions/fpcode/
2. Moroo, J., and Noda, T. Data embedding method on printed materials. *Proc. STEG'04*, 59-69.
3. Norrie, M., and Signer, B. Overlaying paper maps with digital information services for tourists, *Proc. ENTER 2005*, 23-33.
4. Paelke, V., and Sester, M. Design exploration of augmented paper maps. *Proc. ISPRS Workshop Visualization and Exploration of Geospatial Data*, 2007.
5. Qualcomm BREW. http://brew.qualcomm.com/
6. Reilly, D., et al. Just point and click? Using handhelds to interact with paper maps. *Proc. MobileHCI 2005*, 239-242.
7. Rohs, M., et al. Towards real-time markerless tracking of magic lenses on paper maps. *Adjunct Proc. Pervasive 2007, Late Breaking Results*, 69-72.
8. Schöning, J., et al. PhotoMap: using spontaneously taken images of public maps for pedestrian navigation tasks on mobile devices. *Proc. MobileHCI 2009*, 1-10.
9. Uchiyama, H., and Saito, H. AR GIS on a physical map based on map image retrieval using LLAH tracking. *Proc. MVA 2009*, 382-385.

Extraction and Classification of Handwritten Annotations

Andrea Mazzei
CRAFT - EPFL
Rolex Learning Center
Station 20
CH-1015 Lausanne
andrea.mazzei@epfl.ch

Frédéric Kaplan
CRAFT - EPFL
Rolex Learning Center
Station 20
CH-1015 Lausanne
frederic.kaplan@epfl.ch

Pierre Dillenbourg
CRAFT - EPFL
Rolex Learning Center
Station 20
CH-1015 Lausanne
pierre.dillenbourg@epfl.ch

ABSTRACT
This article describes a method for extracting and classifying handwritten annotations on printed documents using a simple camera integrated in a lamp or a mobile phone. The ambition of such a research is to offer a seamless integration of notes taken on printed paper in our daily interactions with digital documents. Existing studies propose a classification of annotations based on their form and function. We demonstrate a method for automating such a classification and report experimental results showing the classification accuracy.

Author Keywords
machine-printed and handwritten text separation, Document processing, annotation classification

ACM Classification Keywords
H.5.2 Information interfaces and presentation (e.g., HCI): Miscellaneous.

General Terms
Algorithms, Experimentation, Human Factors, Languages, Measurement, Performance, Reliability, Theory

INTRODUCTION

Annotating and taking notes on paper is a very common practice in our daily routines. Readers write comments in the margins of papers, underline important passages and use other various marking strategies. These practices help them to understand better what they read and, at a later stage, find back easier relevant passages. It plays also an important role for associative thinking and linking the content with other ideas and documents. Despite the efforts to transfer annotating practices to digital documents, annotating on paper has many advantages compared to any electronic equivalent (Kawase et al. [6]).

This article describes a method for extracting and classifying handwritten annotations on printed documents using a simple camera integrated in a lamp or a mobile phone. The ambition of such a research is to offer a seamless integration of notes, taken on printed paper in our daily interactions with digital documents. Handwritten annotations have different forms and functions (Marshall [8]). We highlight or underline words as attentional landmarks. We write short notes within the margins or between lines of text as interpretation cues. We use longer notes in blank spaces or near figures to elaborate with complementary information. The system, described in this article, aims at not only extracting handwritten annotations, but also classifying them in one of these three categories based on their spatial and colourimetric properties. This automatically generated classification could then be used to sort, organize and share annotations, for instance, in the context of collaborative reading applications.

The first part of the article reviews a number of systems that have been investigated in the last 20 years to tackle this issue. The second part presents our own contribution as original combination of a technique for extracting annotations, a clustering algorithm and a classification approach. To the best of our knowledge the method herein described has not been applied to this problem beforehand. We report the results of a preliminary study showing that handwritten annotations can be extracted and classified in a satisfactory manner using this technique.

MACHINE-PRINTED AND HANDWRITTEN TEXT CLASSIFICATION: A SHORT REVIEW

Discriminating machine-printed and handwritten text in textual images is a problem that has been intensely investigated in the last two decades. In 1990 Umeda and Kasuya [14] described their discriminator of English characters. Their patented invention is based on the strong assumption of uniformity of each block. The discrimination is performed by calculating the ratio between the number of slanted strokes and the sum of horizontal, vertical and slanted ones and by imposing a predetermined static threshold. Under these conditions they achieved a recognition rate of 95%.

Few years later two works focused on the classification at character level. Kunuke et al. [7] proposed a classification methodology based on the extraction of scale and rotation invariant features: the straightness of vertical and horizontal lines and the symmetry relative to the centre of gravity of the character. Their results showed a recognition rate of

UbiComp '10, Sep 26-Sep 29, 2010, Copenhagen, Denmark.
ACM 978-1-60558-843-8/10/09.

96.8% on a training set of 3632 and 78.5% on a test set of 1068 images; Fan et al. [2] used instead the character block layout variance. They reported a correctness rate above 85% tested on English and Japanese textual images: 25 images containing machine printed text and 25 containing handwritten ones. In 2000 Pal et al. [11] presented their method for Bangla and Devnagari; it relies on the analysis of some structural regularities of the alphabetic characters of these languages. Their method uses a hierarchy of three different features to perform the discrimination. The head line is the predominant feature, in fact it forms a peak in the horizontal projection profile of machine-printed text. Their recognition rate is attested on 98.6%. Guo et al. [3] suggested a method based on a hidden Markov model to classify typewritten and handwritten words based on vertical projection profiles of the word. They tested the algorithm on a test-set of 187 words, reaching a precision rate of 92.86% for the typewritten words and 72.19% for the handwritten ones.

More recently Zheng et al. [16] reported a work on a robust printed and handwritten text segmentation from extremely noisy document images. They used different classifiers such as k-nearest neighbours, support vector machine (SVM) and Fischer and different features such as pixel density, aspect ratio and Gabor filter. They achieved a segmentation accuracy of 78%. In the meanwhile Jang et al. [4] described an approach, specific for Korean text, based on the extraction of geometric features. They employed a multilayer perceptron classifier reaching an accuracy rate of 98.9% on a test-set of 3,147 images. On the other hand Kavallieratou [5] showed that a simple discriminant analysis on the vertical projection profiles performs comparably to many robust approaches.

One interesting application is the detection and matching of signatures proposed by Zhu et al. [17], a robust multilingual approach, in an unconstrained setting of translation, scale, and rotation invariant nonrigid shape matching. Peng et al. [12] suggested a novel approach based on three categories of word level feature and a k-means classifier associated with a relabelling post-procedure using Markov random field models; they achieved an overall recall of 96.33%. And finally in a more general scenario of sparse data and arbitrary rotation Chanda et al. [1] recently described their approach based on the SVM classifier and obtaining an accuracy of 96.9% on a set of 3958 images.

METHOD

We here present our approach for extracting and classifying handwritten annotations on machine printed documents. Figure 1 provides an overview of the processing pipeline. It consists of four steps. The first step takes in input the image containing the already extracted annotations and proceeds by clustering the pixels. Parallely the retrieved digital source of the document is processed in order to acquire an accurate estimation of the bounding boxes around the main text blocks present in the image. The set of classified annotations and the estimated bounding box are given in input to a decision tree classifier. A final step is responsible for evaluating the accuracy of the classification by comparing the average colour of each annotation with the predetermined ones.

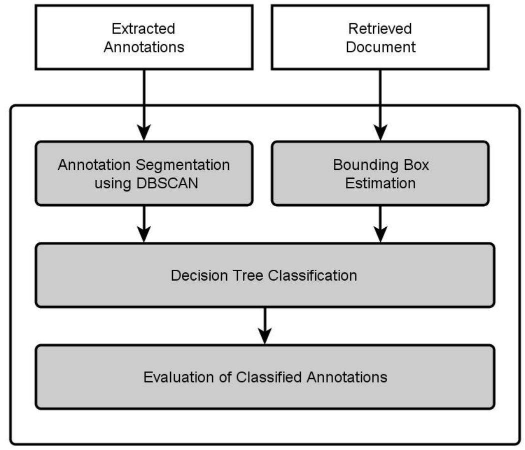


Figure 1. Processing pipeline

Annotation Extraction using Background Subtraction

A novel approach to separate handwritten annotations from machine-printed text is described by Nakai et al. [10]: they realized a method able to extract colour annotations from colour documents. Their method is based on two tasks: fast matching of document images based on local arrangement of features points and flexible background subtraction resistant to moderate misalignment. This method is more general than the above-mentioned ones, since it deals with any kind of annotation and printed document. Later improvements by the same authors [9] showed an accuracy rate of 85.59%. These results encouraged us to adopt their method.

Annotation Segmentation using DBSCAN

This module is responsible for grouping the colour pixels constituting the image containing the extracted annotations. To address this issue we decided to adopt the well known clustering algorithm DBSCAN (Density-Based Spatial Clustering of Application with Noise) for the following reasons:

- the pixels forming an annotation are subject to the conditions of spatial adjacency and colourimetric proximity
- the number of clusters is not known a priori: the number of annotations contained in a page is not predictable
- position, orientation, size and colour of an annotation are variable
- the algorithm should not have a bias toward a particular cluster shape and it should handle noise: the form of an annotation can vary from the rectangular highlighted region to the arbitrary handwritten mark
- the algorithm should distinguish adjacent or even self containing clusters: for instance the highlighted comments

Wu et al. recently reported significant improvements of the original DBSCAN algorithm in terms of time complexity [15]; they removed the original inadequacy in dealing with large-scale data. This allows us not to be bound up with low resolution images.

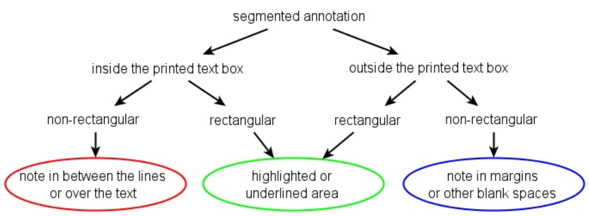


Figure 2. Decision tree classification

The input image containing the pre-extracted annotations is reprocessed. Each pixel is specified by 5 components:

$$p_i = (x_i, y_i, r_i, g_i, b_i) \tag{1}$$

the local position x_i and y_i, used as indexing terms, and the colour information r_i, g_i and b_i, which yields additional discriminative power. The output is obtained by partitioning this set of n pixels into a set of k clusters:

$$A = (A_1, A_2, ..., A_k) \tag{2}$$

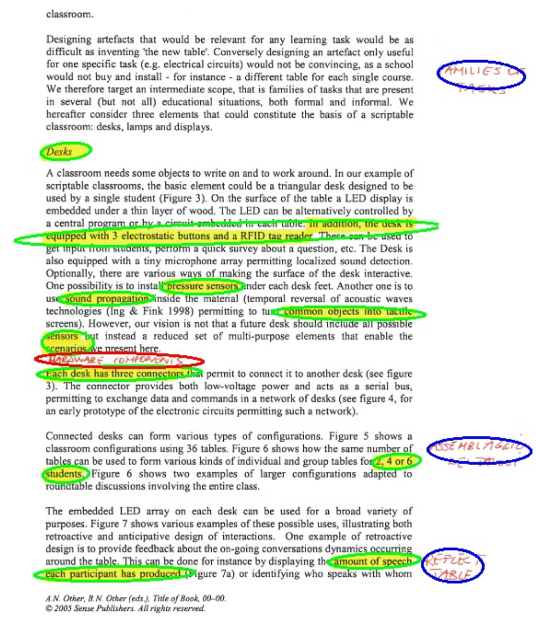
Each cluster corresponds to a correctly segmented annotation. The centroid contains the position of the centre of mass and the mean colour of the annotation. The algorithm is initialized by setting two radiuses, ϵ_{pos} for the spatial domain and ϵ_{rgb} for the colourimetric one and a minimum density $MinPts$ to discriminate all the pixels in core, density reachable and noise points.

Decision Tree Classification

A classification of different forms of annotation is analyzed by Marshall [8]; we regroup the discussed marking strategies by functionality: *memory recall* for underlined or highlighted elements, *interpretation cues* for symbols and short notes in between the lines or over the text, *contents elaboration* for notes in margins or other blank spaces.

We use a decision-tree-based classifier to map the clustered annotations into these categories. Figure 2 illustrates the structure of the decision tree and defines the annotation types in the leaf nodes. In the first level all the annotations are discriminated according to their local position on the page: annotations in between the lines or over text and annotations in the margins or other blank spaces. In the second level all the annotations are separated according to their rectangularity; some methods to compute this derived feature are proposed by Rosin [13]; these methods have desirable properties for our scenario such as rotation invariance and robustness to noise.

The rectangularity is calculated using the minimum bounding rectangle (MBR). More precisely the MBR can be calculated on the elliptical approximation of the shape of interest. Each value of rectangularity is then thresholded to separate more compact annotations such as highlighted areas from others with more complex boundaries such as notes and symbols. Figure 3 shows a satisfactory classification result. In this figure the red, green and blue ellipses contain the notes between the lines or over the text, highlighted passages and notes in the blank spaces respectively.



classroom.

Designing artefacts that would be relevant for any learning task would be as difficult as inventing 'the new table'. Conversely designing an artefact only useful for one specific task (e.g. electrical circuits) would not be convincing, as a school would not buy and install - for instance - a different table for each single course. We therefore target an intermediate scope, that is families of tasks that are present in several (but not all) educational situations, both formal and informal. We hereafter consider three elements that could constitute the basis of a scriptable classroom: desks, lamps and displays.

Desks

A classroom needs some objects to write on and to work around. In our example of scriptable classrooms, the basic element could be a triangular desk designed to be used by a single student (Figure 3). On the surface of the table a LED display is embedded under a thin layer of wood. The LED can be alternatively controlled by a central program or by a circuit embedded in each table. In addition, the desk is equipped with 3 electrostatic buttons and a RFID tag reader. These can be used to get input from students, perform a quick survey about a question, etc. The Desk is also equipped with a tiny microphone array permitting localized sound detection. Optionally, there are various ways of making the surface of the desk interactive. One possibility is to install pressure sensors under each desk feet. Another one is to use sound propagation inside the material (temporal reversal of acoustic waves technologies (Ing & Fink 1998) permitting to turn common objects into tactile screens). However, our vision is not that a future desk should include all possible sensors but instead a reduced set of multi-purpose elements that enable the scenarios we present here.

Each desk has three connectors that permit to connect it to another desk (see figure 3). The connector provides both low-voltage power and acts as a serial bus, permitting to exchange data and commands in a network of desks (see figure 4, for an early prototype of the electronic circuits permitting such a network).

Connected desks can form various types of configurations. Figure 5 shows a classroom configurations using 36 tables. Figure 6 shows how the same number of tables can be used to form various kinds of individual and group tables for 2, 4 or 6 students. Figure 6 shows two examples of larger configurations adapted to roundtable discussions involving the entire class.

The embedded LED array on each desk can be used for a broad variety of purposes. Figure 7 shows various examples of these possible uses, illustrating both retroactive and anticipative design of interactions. One example of retroactive design is to provide feedback about the on-going conversations dynamics occurring around the table. This can be done for instance by displaying the amount of speech each participant has produced (Figure 7a) or identifying who speaks with whom

A.N. Other, B.N. Other (eds.), Title of Book, 00–00.
© 2005 Sense Publishers. All rights reserved.

Figure 3. Annotation classification output

EXPERIMENTAL RESULTS

We have collected 33 annotated pages of scientific articles containing a total of 571 annotations produced by a culturally heterogeneous group of Master and PhD students. They produced the annotations in their own native languages and using their personal style. We set only one constraint: we asked them to use the same colours for each type of annotation within one page. This constraint is imposed only to automatically and objectively evaluate the accuracy of our approach. For each page we supervised the last step of the pipeline (Figure 1) indicating the corresponding function of each colour used for annotating. The experimental results show a classification accuracy of 84.47%.

Strengths and Weaknesses

We here report the observed strengths and weaknesses. The adopted method for extracting annotations from printed documents and the ones discussed in the literature review introduce noise in the separation. DBSCAN effectively identifies and handles these noise pixels. We now report some relevant cases of correct and robust classification and cases of failures. Figure 4(a) shows a difficult scenario in which our approach correctly classifies the annotations. An interline comment is between two highlighted words: in this specific case the spatial information is not discriminative enough to distinguish them: the colour information is determinant to perform the separation. Another strength is that our approach does not depend on a specific language. Figure 4(b) shows a case of correct classification of a note written in Iranian. Figure 4(c) shows a case of correct clusterization but incorrect classification. The big red ellipse contains a chain of bordering highlighted regions. This region is clustered as a set of homogeneous annotations but wrongly classified as interline note because of a wrong value of rectangularity.

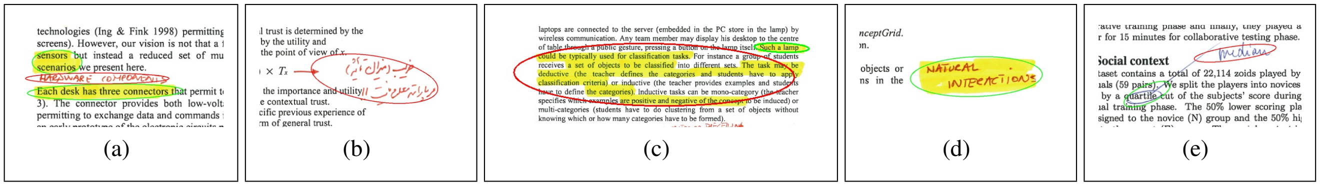


Figure 4. Cases of robust, poor and wrong classification

Figure 4(d) shows a case of self contained annotations. In this case the red ink diffuses into the highlighter ink creating a colour transition between them. This leads to a rough clusterization result. Lastly our approach is not well-suited to capture the notion of linking as shown in Figure 4(e).

CONCLUSION

In this paper we propose a system for clustering and classifying handwritten annotations extracted using already existing techniques. Although there is room for improvements using this approach, the results are promising enough to extend the investigation to a more accurate and granular classification.

REFERENCES

1. S. Chanda, K. Franke, and U. Pal. Structural handwritten and machine print classification for sparse content and arbitrary oriented document fragments. In *SAC '10: Proceedings of the 2010 ACM Symposium on Applied Computing*, pages 18–22. ACM, 2010.

2. K.-C. Fan, L.-S. Wang, and Y.-T. Tu. Classification of machine-printed and handwritten texts using character block layout variance. *Pattern Recognition*, 31(9):1275 – 1284, 1998.

3. J. K. Guo and M. Y. Ma. Separating handwritten material from machine printed text using hidden markov models. In *ICDAR '01: Proceedings of the Sixth International Conference on Document Analysis and Recognition*, page 439. IEEE Computer Society, 2001.

4. S. I. Jang, S. H. Jeong, and Y.-S. Nam. Classification of machine-printed and handwritten addresses on korean mail piece images using geometric features. In *ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 2*, pages 383–386. IEEE Computer Society, 2004.

5. E. Kavallieratou, S. Stamatatos, and H. Antonopoulou. Machine-printed from handwritten text discrimination. *Frontiers in Handwriting Recognition, International Workshop on*, 0:312–316, 2004.

6. R. Kawase, E. Herder, and W. Nejdl. A comparison of paper-based and online annotations in the workplace. In *EC-TEL '09: Proceedings of the 4th European Conference on Technology Enhanced Learning*, pages 240–253. Springer-Verlag, 2009.

7. K. Kuhnke, L. Simoncini, and Z. M. Kovacs-V. A system for machine-written and hand-written character distinction. In *ICDAR '95: Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 2)*, page 811, Washington, DC, USA, 1995. IEEE Computer Society.

8. C. C. Marshall. Annotation: from paper books to the digital library. In *DL '97: Proceedings of the second ACM international conference on Digital libraries*, pages 131–140. ACM, 1997.

9. T. Nakai, K. Iwata, and K. Kise. Accuracy improvement and objective evaluation of annotation extraction from printed documents. In *Document Analysis Systems, 2008. DAS '08. The Eighth IAPR International Workshop on*, pages 329–336, 2008.

10. T. Nakai, K. Kise, and M. Iwamura. A method of annotation extraction from paper documents using alignment based on local arrangements of feature points. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 1, pages 23–27, 2007.

11. U. Pal and B. B. Chaudhuri. Automatic separation of machine-printed and hand-written text lines. *Document Analysis and Recognition, International Conference on*, 0:645, 1999.

12. X. Peng, S. Setlur, V. Govindaraju, R. Sitaram, and K. Bhuvanagiri. Markov random field based text identification from annotated machine printed documents. In *ICDAR '09: Proceedings of the 2009 10th International Conference on Document Analysis and Recognition*, pages 431–435. IEEE Computer Society, 2009.

13. P. L. Rosin. Measuring shape: ellipticity, rectangularity, and triangularity. *Mach. Vision Appl.*, 14(3):172–184, 2003.

14. T. Umeda and S. Kasuya. Discriminator between handwritten and machine-printed characters, 1990.

15. Y.-P. Wu, J.-J. Guo, and X.-J. Zhang. A linear dbscan algorithm based on lsh. In *Machine Learning and Cybernetics, 2007 International Conference on*, volume 5, pages 2608 –2614, 2007.

16. Y. Zheng, S. Member, H. Li, and D. Doermann. Machine printed text and handwriting identification in noisy document images. *IEEE Trans. Pattern Analysis Machine Intelligence*, 26:2003, 2004.

17. G. Zhu, Y. Zheng, D. Doermann, and S. Jaeger. Signature detection and matching for document image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:2015–2031, 2009.

MobARDoc: Mobile Augmented Printed Documents

Tom Holland
The University of St. Andrews
tom.e.holland@gmail.com

Aaron Quigley
The University of St. Andrews
aquigley@cs.st-andrews.ac.uk

ABSTRACT

This paper presents work to date on the MobARDoc system for providing digital features for printed content through an augmented view on a mobile device. MobARDoc matches the printed text seen through the device's camera against the digital version of the same document to determine what content is visible and then overlay a touch based interface on top of the camera image. Our goal is to enable digital features for printed content (including academic papers, business documents, books, magazines and newspapers) which includes: the display of version changes, annotations, reference lookup, hyperlinks, content updates, character / author information, textual search, spelling, punctuation and grammar correction, text copy, dictionary and thesaurus support, pronunciation aid and translation, location extraction and mapping, date extraction and contextualisation and currency adaptation.

Author Keywords Augmented Reality; Paper interfaces.

ACM Classification Keywords H.5.1 [Multimedia Information Systems]: Artificial, augmented, and virtual realities

General Terms Algorithms, Experimentation, Design, Human Factors.

INTRODUCTION

Digital content has yet to completely replace printed content. While much of classical printed material comes from non-digital sources, today the majority of printed content comes originally from a digital source. If the printed content can be associated with its digital original, then we can provide some of the functionality of digital documents for their printed counterparts, using a mobile device to bridge this digital physical divide.

MobARDoc does not rely on Optical Character Recognition (OCR). It is instead based on a technique which matches features extracted from blocks of text within a mobile device's camera image against the original digital version of the document. For the purposes of this work "documents" is a generic term for any printed textual content including books, magazine and newspapers.

To enable augmentation to run in real time on the latest generation of smart phones we introduce methods to enhance the image processing phase to reduce the processing and memory costs.

UbiComp'10, September 26–29, 2010, Copenhagen, Denmark.
ACM 978-1-60558-843-8/10/09.

MOBARDOC

Our approach is broken down into a series of stages involving both the original digital content and the mobile device's camera image of the printed content, which form a pipeline.

Preprocessing the digital content

We use a commercial tool [1] to extract the textual content (including coordinates for individual word bounding boxes) from PDF files.

Earlier work [2] has explored the use of CUPS filter to automatically tag documents being printed with a barcode and to generate a PDF version (which can be pushed to a remote server for indexing as required). Commercially printed items (books, newspapers, magazines) are already printed with a visual identifier (a barcode) and come from a digital original (from which a PDF can be generated).

Document and page selection

When wishing to interact with printed content through the mobile device, the document must first be identified; either through decoding a printed visual identifier with the camera (such as a barcode) or through use of existing work in cover recognition ([3]).

Selecting a document loads the data profile for that document (which includes information such as the number of pages and the available augmentation layers).

To reduce the search space, the specific page must be identified. Currently this is a manual process, but existing work in page layout recognition ([4]) can be applied as an additional phase in the process.

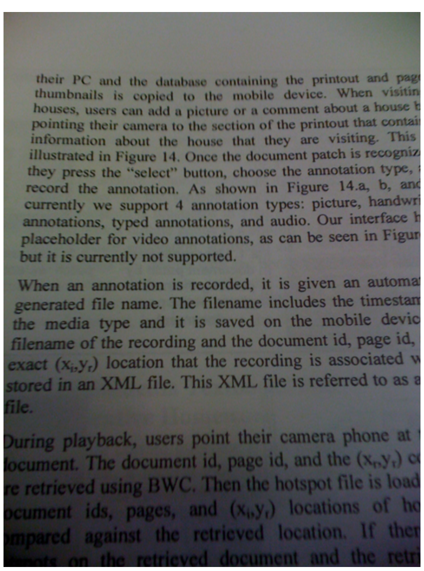
Once a page is selected, the specific data profile for that page is requested from the server. This profile contains a map of the page's content, which is based on features extracted from the PDF version. The page map contains the textual content and bounding box coordinates of each word (in the original digital document's coordinate system). This data may have the textual context excluded (but leaving the bounding box coordinates of each word), which can be preferable in circumstances where exposing the original textual content would not be desirable (such as with copyrighted content).

Binarisation and filtering of the camera image

To identify the pixels from the device's camera image which form the printed text, we first binarise the image (pixels which form the content we are interested in are classified as '1', pixels which form the background, in our case the remainder of the page, are classified as '0'). We convert the input image to greyscale (using the NTSC RGB ratios), resulting in a value between 0 and 255 for each pixel. Binarisation requires selecting a threshold, a value in the scale used to determine if a pixel is part of the content or background. Consider the situation of dark text on a light background, here values below the threshold are classified as content, while values above are background. A desirable threshold value varies, depending on the lighting and other external factors (e.g. shadows cast on the page, such as those by the mobile device being held over it). Global thresholding uses one threshold value for all pixels in the image. This threshold is calculated by first examining all pixels values in the image and uses the distribution to determine a suitable threshold value. Global thresholding is not robust to real-world conditions (due to differences in lighting across the image). Adaptive thresholding is capable of calculating different thresholds for pixels within the same image, by using the values of pixels surrounding the target pixel. While significantly increasing processing time, this does allow for lighting differences to be compensated for.

Our approach is to use different thresholds across the image based on a grid system to separate blocks of pixels. A histogram of the pixel greyscale values in each block is generated and the distribution of values within that block used to determine a suitable threshold. The histograms for blocks which contain text conform to similar patterns in distribution.

This approach is similar in its initial stages to [5] and [6]. Our grid block sizes are larger and their dimensions vary based on the text size in the source document. From a series of sample images of text in documents, books, newspapers and magazines, collected using a mobile device held in a realistic manner (an example can be found in Figure 1), we found that the average text size did not result in distinct


their PC and the database containing the printout and pag
thumbnails is copied to the mobile device. When visitin
houses, users can add a picture or a comment about a house b
pointing their camera to the section of the printout that contai
information about the house that they are visiting. This
illustrated in Figure 14. Once the document patch is recogniz
they press the "select" button, choose the annotation type,
record the annotation. As shown in Figure 14.a, b, and
currently we support 4 annotation types: picture, handwri
annotations, typed annotations, and audio. Our interface h
placeholder for video annotations, as can be seen in Figur
but it is currently not supported.

When an annotation is recorded, it is given an automa
generated file name. The filename includes the timestan
the media type and it is saved on the mobile devic
filename of the recording and the document id, page id,
exact (x_i,y_i) location that the recording is associated w
stored in an XML file. This XML file is referred to as a
file.

During playback, users point their camera phone at
locument. The document id, page id, and the (x_i,y_i) co
re retrieved using BWC. Then the hotspot file is load
ocument ids, pages, and (x_i,y_i) locations of ho
mpared against the retrieved location. If ther

Figure 1. Printed text content through a mobile device camera.

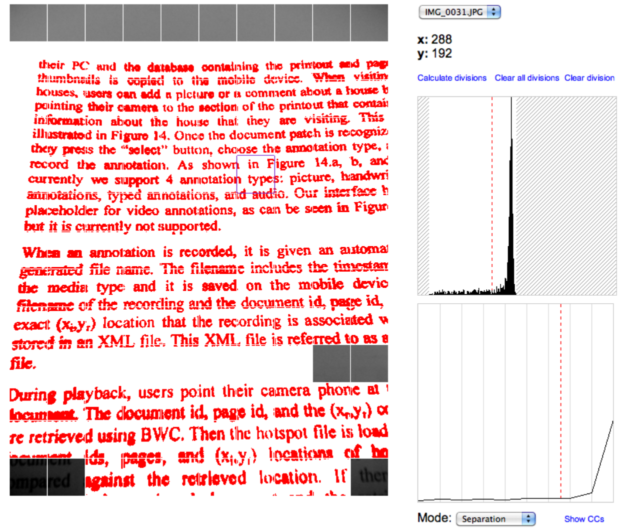


Figure 2. Sample data exploration interface showing pixel blocks, histogram generation, histogram value sampling, threshold calculation and resulting binarised output.

peaks in the greyscale range for content and background pixels (a peak representing the background was preceded with a series of low values across wide range). As a result, identifying bimodality in the form of two Gaussian distributions could only be used to detect and process large pieces of text (such as headings).

Using these sample images (not specifically the target page's or document's content), we pre-generate appropriate threshold values for different histogram distribution patterns (see Figure 2). This has the added benefit of filtering out blocks which do not contain text (or contain non-text items), since blocks not matching a known distribution pattern are ignored (this would require additional stages if standard binarisation techniques were employed).

Text histogram distribution patterns can vary, depending on lighting, the device used and the size of the text. The size of the blocks that compose the grid must also be determined. Although major differences between histograms generated from the digital original suggest that direct comparison cannot be made, the digital original can be used to determine the size of text within the page (and if the page contains pieces of text of different sizes, such as headings and paragraphs). These are used to determine the appropriate size of grid blocks and select suitable histogram distribution patterns from the general pre-generated set to match against. This information is supplied to the device when the data profile for a page is requested. Similarly, device differences which affect the histograms are compensated for by pre-generating sets of general matching data using that device; when a device requests the data profile for a page, the relevant details of the device are sent in the request, allowing specific sets of device customised matching data to be returned.

This technique is less accurate than adaptive binarisation techniques, the resulting classifications containing more noise. Camera images of printed text have the effect of increasing the greyscale range over which pixels are spread. From a digital original of black text on a white background, white becomes varying shades of grey. Black text also becomes shades of grey, but these greys are darker and spread over a narrower range than the white background (although with some overlap). What begins as a crisp edge between black content and white background in a digital file becomes a fuzzy grey transition once printed and viewed through the camera of a mobile device. Our technique results in some of this fuzzy transition being classified as content (causing letters to appear thicker and some to merge into one another). This is acceptable, since we are not performing standard OCR techniques on the content and does not result in an amount of error sufficient to prevent the features extracted from being correctly matched against the digital original.

Connected components and normalisation

Once the pixels have been classified as textual content or other, we use a standard 8 connected components technique to join neighbouring content pixels into 'blobs'. These blobs may contain a single character, multiple characters or whole words (the result of our binarisation is that some letters may have become merged with others). The connected blobs of pixels must now be formed into words (since the features we wish to extract for comparison to the digital original are based on words). The connected pixels are inconsistent; the same camera image may contain blobs of pixels that represent single or multiple letters (see Figure 3). We use a series of domain specific stages to assemble the blobs into words for feature extraction:

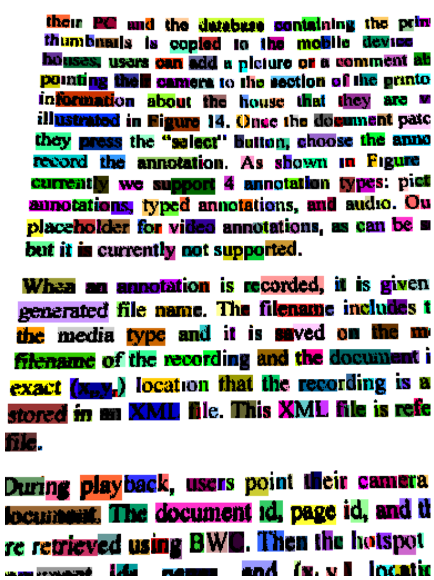


Figure 3. Bounding boxes of connected components after binarisation (visualised with a random background colour assigned to each 'blob' of joined pixels).

1. Calculate the centroid of each blob
2. Select the closest blob corner to each blob corner (excluding its own corners)
3. Using the blob centroids, calculate the bearing between each pair of 'nearby' blobs from the previous step
4. Using the distribution of all these bearings, discount those blocks which do not fit with a line pattern (this does not require that the text forms horizontally aligned lines in the camera image)
5. With the remaining pairs of nearby blobs, calculate the Euclidean distance between the blob edges. Eliminate nearby blobs above a threshold (calculated using the distribution of these distances)
6. Blobs that still have nearby blocks are merged into a single blob to become a word.

Feature extraction and comparison against digital content

We are experimenting with a variety of features that can be extracted from the pattern of words exposed through the previously described stages. Locally Likely Arrangement Hashing (LLAH), an extension of Geometric Hashing, has been built upon in a series of papers ([7], [8], [9]). The use of the ratio of triangle areas within small subsets of words has proven to be resilient against perspective differences between the digital original and the camera's view of the printed content, albeit at the cost of generating all possible combinations within a subset of 'close' words (measured by Euclidean distance, which varies depending on the perspective). Given that we are searching a reduced space (we wish to locate the words visible in the device's camera image within the known page) and existing approaches having been geared towards recognition within ten of thousands of pages, we are looking at less intensive methods which produce a more generalised signature for a block of words (we only require that a block of words is uniquely identifiable by pattern within a page).

Distortion calculation and interface overlay

Once the pattern of words visible to the device has been matched to words within the digital document page, the difference in perspective between the digital original and mobile device's view on the printed document must be calculated to enable the interface to be overlaid on the mobile device's view. Using the centroids of matched word bounding boxes as points on a plane, and using the matched points in the original digital document as the base, a homography is calculated. The visible portion of the relevant overlay interface (which is stored in the original digital document's perspective) can then be distorted accordingly to match the view through the mobile device.

User interaction and augmentation layer switching

Augmentation layers are the overlays available for a document. These are composed of content to be overlaid on top of the camera image and interface elements which can be triggered from this content (which may include the ability to display further content related to the items being interacted with).

Different types of documents will benefit from different augmentation layers. For a business document or research paper this may include differences between versions, coworker / coauthor annotations and hyperlinks for URLs or references. For books these may include annotations by the original author (similar to the concept of "DVD extras"), updates to the content (such as more recent information in a travel guide) or summaries attached to each character name when displayed in a novel, which become increasingly detailed the further into the book the page belongs, to help a reader keep track of characters and their history (without exposing as yet unrevealed plot details). For magazines: author profiles and hyperlinks; for newspapers: updated article content (such corrections and late additions to a story). Media (such as audio and video) can be attached to existing content with hyperlinks or indicated by iconography in the document borders.

Human authored augmentation layers by an authority (such as the author or editor) could be made available to all; some augmentation layers could be determined by the reader's social network (allowing a user to see a friend's / colleague's comments on an article if desired, allowing digital comments to be attached to printed articles).

Numerous automatically generated augmentation layers can also be leveraged. These could include: textual search, hyperlinks, spelling, punctuation and grammar corrections, text copy functionality, reference extraction, dictionary and thesaurus lookup, pronunciation and translation (using online services), location extraction (linking to maps), date extraction (what else happened then; add to my calendar) and monetary values (conversions between different currencies; value now in comparison to when printed).

The inclusion or exclusion of the original textual content in the data profile affects which augmentation layers can be made available. Augmentation layers based on a person adding content (such as annotations) are unaffected. Automatically generated augmentation layers (such as spelling, textual search, text copy, dictionary or thesaurus) require that the original textual content must be exposed to allow them to function.

CONCLUSIONS AND FUTURE WORK

We have presented our work in progress on the MobARDoc system to provide digital document functionality to printed digital documents through a mobile device. We are specifically using features extracted from text patterns, rather than OCR, to match the portion of content visible through the mobile device's camera within the original digital document page to allow a variety of functionality to be made available through an interface overlaid on the device's camera image. Our work can be differentiated from existing approaches in our specific targeting of the real-world use of mobile devices and adapting or replacing parts of the pipelines used in similar work to meet our goals.

We plan to continue to develop our pipeline and produce a functional prototype on a mobile device suitable for a mixture of automatic testing and human-centered experiments in two of our target domains: research papers and books.

REFERENCES

1. Pdflib text extraction toolkit. http://www.pdflib.com/products/tet/
2. Holland, T. and Quigley, A. Doctrack: automatic printed digital document tagging and remote retrieval. Late Breaking Results of the Sixth International Conference in Pervasive Computing, May 2008.
3. Tsai, S.S., Chen, D., Singh, J. and Girod, B. Rate Efficient Real Time CD Cover Recognition on a Camera Phone. In ACM Multimedia, Vancouver, Canada, October 2008.
4. Van Beusekom, J. Document Layout Analysis. Diploma Thesis, Image Understanding and Pattern Recognition Group, Technical University of Kaiserslautern, Kaiserslautern, Germany (under Prof. Dr. Thomas Breuel).
5. Chow, C.K. and Kaneko, T. Automatic detection of the left ventricle from cineangiograms. Computers and Biomedical Research, vol. 5, pp. 388-410, 1972.
6. Taxt, T., Flynn, P.J. and Jain, A.K. Segmentation of document images," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 11, no. 12, pp. 1,322-1,329, 1989.
7. Nakai, T., Kise, K. and Iwamura, M. Use of affine invariants in locally likely arrangement hashing for camera-based document image retrieval. Document Analysis Systems VII, pp. 541–552, 2006.
8. Nakai , T., Kise, K. and Iwamura, M.. Camera based document image retrieval with more time and memory efficient LLAH. In Proceedings of Second International Workshop on Camera- Based Document Analysis and Recognition (CBDAR2007), pp. 21–28, 2007.
9. Uchiyama, H., Pilet, J. and Saito, H. On-line document registering and retrieving system for AR annotation overlay. In Proceedings of the 1st Augmented Human International Conference (AH '10), pp. 1-4, 2010.

Towards Understanding Erasing-based Interactions: Adding Erasing Capabilities to Anoto Pens

Simon Olberding and Jürgen Steimle
Technische Universität Darmstadt
{olberding, steimle}@tk.informatik.tu-darmstadt.de

ABSTRACT
Anoto pens are a powerful technology for capturing contents written on paper. However, current pens do not support erasing contents. We show how to easily construct refills for Anoto pens that allow users to erase handwritten traces. Moreover, we discuss how to design software solutions that incorporate paper-based erasing as a first-order command.

Author Keywords Anoto, pen, deleting, digital paper, erasing, rubber.

ACM Classification Keywords H.5.2 User Interfaces.

General Terms Design, Human Factors.

INTRODUCTION
The Anoto [1] technology has considerably bridged the gap between computing and pen and paper, which still prevails in many contexts of use. Electronic Anoto pens are capable of capturing handwriting on plain paper and make this data available to computer systems. Anoto pens are used in many research prototypes and commercial applications, mainly for the purposes of form filling (e.g. Anoto Forms Processing) as well as for notetaking and annotation (e.g. [2],[3],[4],[5]). Notetaking solutions allow users to write and draw with an electronic pen on a paper notebook or on printed documents. The system then automatically creates a digital version of the document, which can be used for archival purposes, for full-text search within contents or for sharing contents with co-workers, to state only some examples.

Yet the use of Anoto pens entails a problem: while pens can be used for writing and drawing strokes, the reverse action is not possible, as they cannot be used for physically erasing strokes. Some systems allow the user to make a specific pen gesture for indicating that strokes should be deleted, e.g. a cross-out gesture preformed on the content to delete. This however decouples the physical representation from the digital representation. While the content is deleted from the digital representation, it is still visible in the physical one. This is particularly problematic if some contents are to be replaced by others, since it is not possible to write (or draw) over "erased" contents which are still visible. In contrast, digital graphics tablets allow for erasing contents with a specific erasing pen or with the backside of the pen.

The aim of this paper is to encourage the field to take physical deletion into account and to gain a deeper understanding of erasing-based interactions. We contribute to this by showing how to modify Anoto pens such that they can be used for physically (and electronically) erasing handwritten contents on paper. The prototype can be easily constructed and requires only cheap standard materials. We further discuss how software solutions should be designed for taking into account paper-based erasing as a first-level command.

ANOTO TECHNOLOGY
The Anoto technology depends on two components: A specific dot pattern which is printed on the sheets of paper encodes the unique position. The Anoto pen behaves like a traditional ballpoint pen and leaves visible ink traces on paper. During writing, it (see Fig. 7) uses a built-in camera to read the pattern and decodes its position. Moreover a sensor recognizes the force with which the pen tip is pressed onto paper while writing. This data is either streamed in real-time to a nearby computer or temporarily stored on the pen.

The ink reservoir and ballpoint unit can be easily replaced with a refill by just pulling the old one out and plugging the new one in. Current refills for Anoto pens come with standard ballpoint ink in blue color, which cannot be erased. Even if one found a material which is capable of erasing ballpoint ink, the erasing activity would not be electronically tracked.

BRIEF SURVEY OF CONVENTIONAL ERASING PENS
As a reference for our solution, we investigated on traditional solutions for erasing handwritten content.

One solution for removing ink of a fountain pen is to use an ink eraser. These have traditionally two tips. One for erasing the ink, the other one is for rewriting on the erased area. Whiteout overcomes the limitation of having to use a specific pen for rewriting. It can erase any content on a sheet of paper. Rewriting is possible with a large range of pens. However, whiteout needs time to dry, is permanently visible and moreover erases all contents, not only handwritten traces.

Another well known technology for erasing is using a rubber in combination with a pencil. The graphite ink of the

UbiComp'10, September 26–29, 2010, Copenhagen, Denmark.
ACM 978-1-60558-843-8/10/09.

pencil gets gradually deleted by rubbing over it with a piece of specific synthetic gummy or caoutchouc. The combination of pencil and rubber has the advantage that content can be modified in a large variety of ways. For example, the transparency of the content can be adjusted with a fine granularity. Moreover, the graphite can also be smeared with a finger on the paper, which is specifically interesting for artists. However, a pencil is not adequate for writing something durable (e.g. signing a contract or making notes during a lecture).

A more recent technology uses a specific ink which becomes transparent when heated. When the user rubs with an erasing unit over the ink, this generates heat and makes the ink disappear. If temperature is very low (e.g. in the freezer), the ink becomes visible again. The Pilot Frixion Ball uses this technology. It is available in different versions with 10 different ink colors.

Ink erasers and whiteout are not appropriate for use with the Anoto pen. In contrast, traditional pencil and rubber as well as the Frixion pen are promising technologies.

ADDING ERASING CAPABILITITES TO ANOTO PENS

In this section we formulate some general requirements for building a custom refill for an Anoto pen. Then, we show how to construct two different types of erasing pens. The first is a pencil and a rubber, the second one is based on the Frixion technology. For each type of erasing pen, we constructed a writing and an erasing refill. In the case of the pencil and the rubber, we built a refill with a pencil as tip and a refill with an eraser as tip. In the other case, we built a refill which allows to write with the specific ink used in Frixion pens and a refill with a rubber ball as tip which is used for erasing.

All refills are working with all common types of Anoto pens. We successfully tested them with Logitech io2, Nokia SU-1B and Anoto ADP 301 pens. For ease of construction, we decided to separate the erasing and writing capabilities and created a second pen specifically for erasing (as done by rubber pens or ink erasers). Future Anoto pens could include an eraser on the upper end (like pens of Tablet PCs). Nevertheless, separating erasing and writing into two pens reflects also common work practice of creative workers (e.g. industrial or graphic designers).

The shape of the pen and its technology form several general requirements for an eraser pen: 1) The ink and the eraser must preserve the Anoto pattern. 2) The refill must correctly activate the pressure sensor, while writing with the pen. Therefore, the refill must fit in the casing. 3) The tip of the pen must be small enough to ensure that the pattern remains visible to the camera.

Preparation

Before detailing of how we constructed both types of erasing pens, we list the tools and materials that we used: superglue, nipper pliers, a pair of scissors, a drill machine and a 2mm drill, a Frixion pen, a rubber, a 1.5cm long and 2mm wide graphite refill for drop action pencils, a medical syringe, a 12cm long and 2mm wide plastic pipe and two Anoto ballpoint refills.

Generally, each refill consists of a reservoir for the ink and a tip. We constructed two refills with a plastic pipe as reservoir, each with a length of 6 cm. Since the tip of the Frixion pen and the one based on graphite do not fit in the plastic pipe, we widened the opening of both plastic pipes with the 2mm drill for a length of 4mm each.

Pencil & Rubber

In order to connect the graphite lead to the widened plastic pipe, we added a drop of superglue on the lead and inserted it into the pipe. If the refill is inserted into the pen, it might not stick inside, since the pipe and the tip are too small in diameter. As workaround, we added a 5mm wide and 20 mm long adhesive tape around the upper part of the pipe. Figure 1 shows the completed refill that can be plugged into the pen.

Figure 1. Graphite refill.

Figure 2. Rubber refill.

In order to build the eraser, we took a standard Anoto ballpoint refill as reservoir and plugged a piece of rubber (6mm long and 3mm in diameter) on its tip. Therefore, we drilled a 4mm deep and 2mm wide hole into the rubber. Figure 2 shows the result.

Depending on the frequency of use, graphite and rubber might have to be replaced in regular intervals. If the rubber is worn down, it can be easily replaced on the refill. Our current construction of the graphite refill, however, does not allow to replace only the graphite lead, as the lead is glued to the refill. One has to use a new refill, but this is cheap and easy to construct.

Frixion & Rubber ball

We now describe how to built the refills for using the Frixion approach. The components for the writing refill consist of a widened plastic pipe as reservoir and the tip of the original Frixion Pen. The tip can be pulled out of the Frixion pen's reservoir by using nipper pliers. Then, we added a drop of superglue on the rear part of the tip and plugged it into the pipe. The result is shown in Figure 3.

Figure 3. Frixion writing refill.

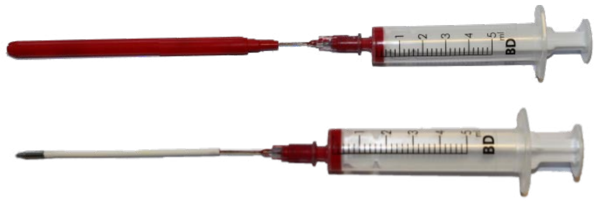


Figure 4. Filling the ink reservoir.

In order to transfer the ink from the Frixion pen into the refill, we used a medical syringe. First, we filled up the syringe with ink of the Frixion pen (see Fig. 4 upper part). Thereafter, we inserted the ink into the refill (lower part). It takes approximately five to ten minutes until the ink reaches the tip. Then the refill is ready to be plugged into the Anoto pen.

We also experimented with the M70 Lamy IT refill. This is a refill with a plastic tip and a stable reservoir of metal, which has the correct length and diameter to fit in the Anoto pen and to carry the tip of the Frixion pen. However, we found that it is highly difficult to create a stable connection between the tip and the reservoir. Both are made of metal, a material that is hard to glue together with standard materials.

In order to create the rubber, we used the standard Anoto refill as reservoir and constructed a tip using part of the Frixion pen's rubber ball. First, we pulled the rubber ball out of the casing of the Frixion pen. Next, we cut the eraser with a pair of scissors into two pieces (see Figure 5). Then we plugged the smaller part onto the tip of the Anoto refill. In order to avoid that the Anoto refill leaves ink traces, we added a drop of superglue on the tip of the refill. Figure 6 shows the completed Frixion eraser refill which can be used with a digital pen.

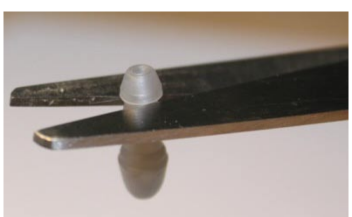


Figure 5. Frixion rubber ball.

Figure 6. Frixion eraser refill.

ERASING-BASED INTERACTIONS

In this section, we discuss how software solutions should be designed for taking into account paper-based erasing as a first-level command.

Facsimile of handwritten content. Many applications of paper-based computing show a facsimile of the handwritten content. Erasing-based interactions can be used to keep the digital facsimile in-sync with handwritten content on paper, even if content is erased.

Deletion on computers typically is an atomic action, the file being either deleted from its original location or left as it is. In contrast, erasing handwritten content is situated on a continuum. When rubbing once, content is deleted only to a certain extent. Only when rubbing over the same point several times, the content gets totally deleted. This allows different graduations of erasing.

From a physical perspective, the amount of deletion depends on the material of the sheet, the type of the pen and the type of the eraser. For example, it makes a difference whether to erase a stroke of a pencil with a high or low degree of hardness and whether it is written on a coarse or glossy page. These different graduations of deletion as well as the characteristics of the materials should be taken into account by applications. This allows to maintain the visual appearance of (partially and fully) erased ink traces on paper in-sync with the digital facsimile.

We developed a prototype application for notetaking and sketching. Users can take handwritten notes or make graphical sketches on paper and erase them on paper. Pen data is streamed to a computer via a Bluetooth connection. A digital facsimile is automatically made available in a software viewer. The application tracks not only the position information of writing and erasing traces, but also the pressure on the pen tip. If a trace is written with higher pressure, the user has to perform more pressure with the erasing pen or has to rub more often to erase it than if the trace is written with lower pressure. This creates a very authentic visual appearance of the facsimile and allows the user to maintain the expressiveness of traditional erasing techniques. Figure 7 shows an example of physical erasing and how the traces are visualized in the software viewer.

In order to keep the appearance of physically deleted content in-sync with its digital counterpart, a calibration step is necessary. The calibration depends on of the type of refill (e.g. graphite or ink), the surface of the paper and the material of the rubber. Currently, we manually calibrate the software, but we plan to implement a module for semi-automatic calibration in the near future.

Semantic interpretation. Besides ensuring a faithful facsimile, erasing-based interactions can also be leveraged for semantic interpretation by computers. This allows to attribute a semantic meaning to the grade of a note, which is situated on a continuum between fully visible and fully deleted. This opens up a design space for a novel class of very intuitive interactions. For example, slightly decreasing the note's visibility by rubbing over it could mean to lower its priority. In contrast, thickening up the traces of a stoke by writing or drawing them again could increase the priority.

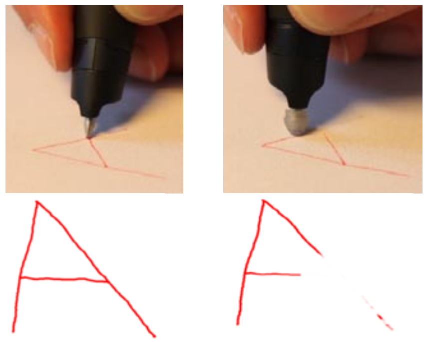


Figure 7. The user writes a letter (above left) and then gradually erases it (right). The visualization is shown below.

QUALITY TESTS AND EARLY USER FEEDBACK

In order to evaluate the quality of the solution presented, we performed several mechanical tests with common Anoto pens (Nokia SU-1B, Logitech io2, and Anoto ADP 301). We evaluated the following criteria: robustness of position tracking, robustness of pressure sensing, influence of strong mechanical forces, effects on the pattern, as well as wearout of the refills. Our reference was an Anoto pen with a standard Anoto refill.

The results show that the novel refills do not disturb the field-of-view of the camera, such that the pattern remains decodable. The applications receives correct data from the pen even if the pen is inclined up to 45 degrees from the vertical position. This is the same maximum angle as with a standard refill. Generally, the radius of the tip should be smaller than 3mm in order to not disturb not camera.

The next test verified if the pressure sensor in the pen works correctly. To simulate heavy use, we tapped with each refill successively 50 times in about 15-20 seconds. Even in this case, the pen correctly recognized all pen down and pen up events.

Then we tested use with strong mechanical forces, both in the horizontal and in the vertical dimensions. With each refill, we pressed the pen strongly onto the paper sheet and rubbed with very quick back-and-forth movements. The rubber refill, Frixion writing refill and Frixion erasing refill worked properly without any problems. The pencil graphite refill, however, could not resist to strong forces because then the graphite lead broke. In cases of normal use, without purposely applying strong forces, also this latter refill proved to be sufficiently stable. Moreover, we continuously shaked the Frixxion refill for three minutes and put it upside down for two hours. The ink remained inside the refill.

We also tested if the Anoto pattern remains intact after a large number of writing and erasing cycles. For both types of writing/erasing refills, we repeatedly wrote and completely erased contents at the same position. Even after 30 cycles, the pattern was decodable at the erasing position.

The graphite lead refill and the rubber refill get worn out by use and reduced in height. We measured the minimal and maximum length of the refill that still ensures that the pen works correctly. The minimum length is 6.2 cm and the maximum length is 7.2 cm. This tolerance of 1.0 cm is large enough for the reduction in height generated by longer use.

In order to verify, that the refills bear up the usage habits of actual users, we gathered early feedback from two members of our lab. Each of them used our prototype for 20 minutes, while being free to draw and erase whatever he wanted. He could see the digital facsimile at the same time. We encouraged the participants to use the pen naturally. After ten minutes we exchanged the erasing and writing refills.

Both participants were enthusiastic about the possibility to delete handwritten content. One participant even stated, that "this pencil feels more natural than the Anoto ballpoint pen itself". While both reported that the mapping of the pressure force of the pen to the digital version creates a natural facsimile with the graphite lead and rubber refills, they also stated that this mapping has to be improved when using the Frixion refill. Moreover, both stated that the Frixion eraser needs more pressure force than the rubber to erase contents. One participant repeatedly wrote and erased content with the Frixion refills. He was surprised, that even after the paper felt coarse, the pattern could still be read by the pen.

SUMMARY AND CONCLUSIONS

As a first step towards research on pen-and-paper interactions that involve erasing, we demonstrated how to construct two different types of erasing pens that work very reliably with standard Anoto pens. Furthermore, we gave an example of how erasing-based interactions can be used to create faithful and naturally looking digital facsimiles of handwritten content. Moreover, we gave an outlook on how erasing-based interactions can be semantically interpreted. In future research, we plan to investigate the design space of erasing-based interactions more deeply.

REFERENCES

[1] Anoto. Digital pen and paper technology. Anoto AB. http://www.anoto.com.

[2] F. Guimbretière, "Paper augmented digital documents," Proc. UIST 03, 2003.

[3] R. Yeh, C. Liao, S. Klemmer, F. Guimbretière, B. Lee, B. Kakaradov, J. Stamberger, and A. Paepcke, "ButterflyNet: a mobile capture and access system for field biology research," Proc. CHI 06, 2006.

[4] B. Signer and M.C. Norrie, "PaperPoint: a paper-based presentation and interactive paper prototyping tool," Proc TEI 07, 2007.

[5] J. Steimle, O. Brdiczka, and M. Mühlhäuser, "CoScribe: Integrating Paper and Digital Documents for Collaborative Knowledge Work," IEEE Trans. on Learning Technologies, 2(3), 2009.

Improving paper books: searchable books

Ming Ki Chong and Fahim Kawsar
Computing Department, Lancaster University, UK
{chong, f.kawsar}@comp.lancs.ac.uk

ABSTRACT

Much of today's information is digitised. Representation of information is increasingly becoming digital. Yet, paper books remain popular, as many readers prefer the reading experience that paper books provide, which digital interfaces cannot. In this paper, our aim is to improve users' reading experience by enhancing books with digital functionalities. We conducted a user survey study to identify features that users desire. The study highlights one specific feature – content searching within books. From this result, we discuss three design choices that can incorporate digital searching into paper books.

Author Keywords

Books, User Survey, Digital Search.

ACM Classification Keywords

H.5.2 Information interfaces and presentation (e.g., HCI): User Interfaces – *User-centered design*.

General Terms

Design, Human Factors.

INTRODUCTION

Since the dawn of books, inscribing text on papers has traditionally been the primary medium for conserving information. Historically, paper has been the preferred option because of its availability, lightweight, as well as its low-cost economic value and other benefits. Over many of the past centuries, countless number of writers have contributed immense volume of literatures on papers and the literatures have been passed down many generations. Throughout this extent of time, literature readers have developed proficient skills of comprehending paper-based information. Subsequently, the ability to read has become one of the common skills of many people today.

As we shift towards the new era of digitising information, the trend of conserving information is also changing. Already, a large amount of today's information is stored in and presented on digital devices, web pages for example. Also, device capacity nowadays allows the storage of over hundreds or thousands of digital books on a single device. Integrated with network capability, users can access millions of digital books online. Compare to paper-based literatures, access to digital materials is far quicker and easier. In summary, digital technology offers many possibilities and functionalities that paper interfaces cannot offer [5].

Both paper-based and digital books have individual inherent benefits. Paper books, as a traditional medium, offer tangibility as well as many affordances [6]. For instance, regardless of a book's content, when a reader picks up the book, he/she automatically knows that the content is presented in sequential pages (i.e. text that spans more than a page is subsequently continued on the following page) as well as how to access a desired page (by either simply leafing through and turning pages until the desired page or approximately opening the book to a page close to the desired one). Digital devices, on the other hand, has the benefit to download or copy e-books from other devices, without losing the quality as the content is represented in digital form. Each medium has benefits that the counterpart cannot offer, and yet, one medium's benefits cannot be adopted by the other. In this paper, our aim is to find ways to improve users' reading experience by incorporating digital functionalities into paper books.

Many researchers have suggested various methods and novel concepts to improve users' reading experience. For instance, Watanabe et al. [7] presented "BookiSheet", an interface that consists of bend sensors for scrolling through digital content, which provides the tangible sense of turning pages in a book by bending. They envisioned combining the interface with a flexible display to provide readers the impression of a real book. Similarly, Fujinami and Inagawa [2] suggested embedding sensors in paper books to detect page turning. When a user turns a page, the sensors detect the action and an external display show multimedia information according to the current page that the user is reading.

Numerous ways to improve users' reading experience with digital technology have been suggested in research. Instead of focusing on the technology, we adopt the *Human-Centered Design* (HCD) methodology (*Hear*, *Create* and *Deliver*) from IDEO [3]. In this paper, we present our results of the "Hear" phase, by understanding users' needs and investigate the dire issues that we must solve for the users. We

UbiComp '10, Sep 26-Sep 29, 2010, Copenhagen, Denmark.
ACM 978-1-60558-843-8/10/09.

thus conduct an informal survey with potential users (colleagues from our university) to identify the features that people want to have when paper interfaces are integrated with digital functionalities (cf. section: Survey Study). From the survey study, our interviewees helped us to identify that the functionality of content searching in paper books should be considered. We therefore focus on this feature and discuss three design choices to implement searchable books.

SURVEY STUDY: UNDERSTANDING USERS' NEEDS

As an initial study, we recruited seven colleagues (6M, 1F) from our university; six postgraduate students and one postdoctoral researcher. They all have background in computer science. The survey was conducted as informal discussions with the interviewees individually or in pairs. Each discussion lasted for about 15 to 20 minutes. We prepared three pre-selected questions (see below) to start the discussion. After asking each of the questions, based on the subjects' responses, we further inquired their reasoning by asking them open questions.

For the rest of this section, we summarise the interviewees' answers from the discussions (see figure 1 for the summary).

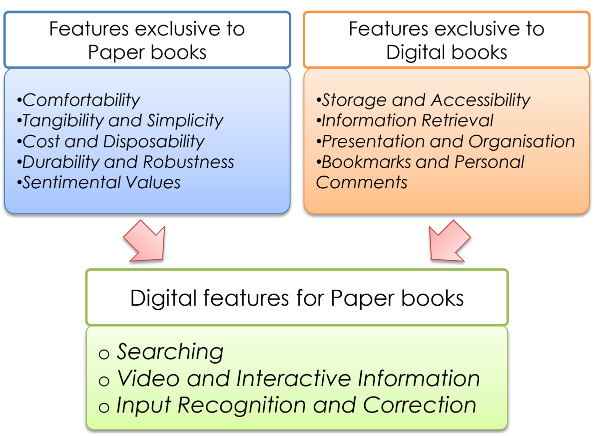


Figure 1. A summary diagram of our interviewees' responses

•***What features do paper books offer that digital books do not?***

Comfortability: Many the interviewees mentioned that reading information on paper is more comfortable because: (1) text on paper is illuminated from natural ambient light, thus easier to focus; while information on backlit displays are difficult to read for a long period. (2) Paper books are easy to hold since the material is lightweight and bendable, while digital devices are rigid and fixed in shape. (3) Paper has been predominant for many generations; many people prefer to read text from an interface that they are familiar with. Furthermore, one subject explicitly mentioned that since his work involves much around with technology, when he reads books during his free time, he prefers an interface that does not remind him of technology and work.

Tangibility and Simplicity: All paper interfaces are tangible, as information is presented on a physical surface. Due to the inherent characteristics of paper, the interface can only display static information; any content printed on paper is fixed and permanent. As a result, paper interface provides an affordance of what readers see on a page is what they get and expect no extra information (i.e. the concept of What You See Is All You Get). This subsequently simplifies readers' expectation, as there is no hidden sub-functions.

Cost and Disposability: The monetary value of a book depends on its prestigiousness. Many books are cheap and designed to be disposable. Magazines for instance, once they are read or outdated, they are recycled. Because of the low cost values of books, readers do not worry about misplacing the reading material. On the contrary, the current costs of electronic book readers are much more expensive than paper books; consequently, users often need to handle the devices with care. Furthermore, digital books induce two costs: a cost for an electronic book reader and another separate cost for the digital book content.

Durability and Robustness: The nature of paper allows books to be manhandle with carelessness. Paper books can be thrown, dropped, and smashed, as long as the action does not cause the printed information to fade. One of the subjects jokingly told us that he would not hesitate to pick up a book to hit a fly or a mosquito, but he would not do the same with an electronic device.

Sentimental values: Beside monetary values, people attach sentimental values to paper books. For example, books are often used as gifts; a book given by someone special is unique to the receiver. Thus, a physical book can be seen as a irreplaceable memento. On the contrary, people attach less sentimental values to digital books, since digital information exists virtually and it is easily duplicated and recovered.

•(*Vice versa of the previous question*) ***What features do digital books offer that paper books do not?***

Storage and Accessibility: Current mobile devices have immense data storage capacity. A single device can store over thousands of books. As a result, it provides users the convenience of able to access a library of books on one device, i.e. virtual content increases while the physical characteristics remain unchanged. Furthermore, as ubiquitous computing proliferates, many devices nowadays have network capability; this allows users to access online digital books. Via a mobile network connection (like HSDPA), users can access and download digital books from any location.

Information Retrieval: Results from our survey show that all of the interviewees recognised keyword searching on digital books is a prominent advantage that paper books do not offer. Traditionally, paper books have a list of indexes (usually at the end of the book) to help readers find information quickly and easily. Yet, this functionality is only useful if the search keywords are indexed. Whilst with digital books, since information is digitalised, indexing the entire content is possible. As a result, the list of indexes for digital books is much larger.

Presentation and Organisation: Digital content is separate from its presentation. For example, users can change the font and colour of the text without changing the book's content. As a result, users can personalise interfaces according to their preferences. Beside presentation, users can also personalise the ways of organising their libraries. For example, users can sort books according to genres, authors, or languages via a call of the sorting function.

Bookmarks and Personal Comments: With the suitable input interface, digital book readers can support functionalities that allow users to annotate paragraphs with personal comments or to leave bookmarks in any pages. These features can be hidden or removed without affecting the original content.

•If paper books could adopt digital books' characteristics, what features (or functionalities) should be included?

Searching: All of our interviewed subjects suggested searching for information by keywords is a fundamental feature, especially for reference books (like textbooks). People often want to retrieve text that was previously read or to search for specific information in a book without browsing many pages. Manual search is time consuming, as it requires flipping through numerous pages; instead, digital search is convenient, precise and accurate, as the system finds the exact locations of the information in the book. Furthermore, multiple search terms can be inserted at once for retrieving related information, and by matching all the terms, digital search can provide higher precision. However, since multiple results are often returned, designers must select an appropriate output to ensure results are presented in a readable manner. Other than using keywords, some interviewees also suggested the use of audio speech or drawing doodles to search for information in books. For example, a user scribbles a sketch, and then the system analyses the input sketch and returns figures that are related to the drawing.

Another suggested search feature was selecting text from the book and then search for additional information on the Internet (i.e. web search based on the information from the book). For example, while a user is reading a book, the user may want to search for definitions. The user can select a certain passage on a page and perform an online search (like a dictionary or Wikipedia) to find additional information about the selected text.

Videos and Interactive Information: The second feature our interviewees suggested was displaying videos on paper. Often, ideas can only be expressed in videos; nevertheless, this option is limited in paper books because of the nature of paper. Paper interfaces can only offer static information, like text and still images. Currently, to simulate the display of a video on paper, sample frames from the video are used to express the story. However, the experience of viewing sample images from a video is not the same as watching the entire video, as much of the information is lost. Web pages for example, when viewing a video on a webpage, we can play, pause, rewind or fast-forward the videos; however, if the webpage was printed on paper, we lose the controllability as well as the videos.

Following the idea of embedding motion pictures in books, our interviewees also suggested the adoption of interactive information in books. For example, additional information should only be shown on demand, like videos should only play on users' request. This gives readers the control of information flow and avoids disturbances by the extra content.

Input Recognition and Correction: Other than displaying information, books can also be used for capturing information. Notebooks for example, people often jot down ideas or draw figures on notebooks. Our interviewees suggested the idea of having an input interface that users can write text or draw figures on paper and the system can automatically correct any mistakes, like changing a freehand drawn circle into a perfect circle.

Besides identifying features to improve paper books with digital functionalities, other suggestions include improving the experience of reading electronic books. Physical form factors were suggested; for example making digital book readers more paper-book like by using bendable and lightweight components to manufacture the devices, incorporating the smell of books, etc.

Although many of the suggested features from study are still far-fetched with the current technology, one particular suggestion, information searching within books, is possible.

DESIGN CHOICES FOR SEARCHABLE BOOKS

Many (paper-based) books have an index section with a list of pre-selected indexed words. Traditionally, searching for desire information in a book requires the reader to find indexed terms and the corresponding page numbers from the back-of-the-book index. Nonetheless, if the search terms are not indexed, the alternative approach is readers must exhaustively browse through pages of content until the information is found, but there is no guarantee that the information will always be found. On the other hand, since information can be digitised, a digital reader can index the entire content. Similar to web search engines, an indexer converts the books' entire content into a full inverted index. Once content is indexed, users can enter search queries and retrieve related information at an instant speed. This method is convenient and timesaving as most of the laborious work (like manual searching) is done automatically.

Consequently, we discuss three related design choices for incorporating search facilities into users' reading experiences.

1. Electronic Book Readers : As the name implies, the straight forward approach to incorporate search function in books is to make books fully digital and use an electronic interface to support user input and information output. The idea of using an electronic device to view digital books has existed for many years. Earlier versions of electronic books use personal computer interfaces, like desktops or laptops, as a medium to output books' content.

As technology evolves, more people are reading digital books. In the recent years, mobile electronic book readers (also known as e-book readers or e-Readers) have gained much popularity in the book industry. The functionality of searching for information in e-book readers already exists commercially. For example, the built-in e-book application of Apple's iPad has an integrated search function for users to locate information in a book. By entering search terms, the system automatically retrieves related information from the book that the user is reading.

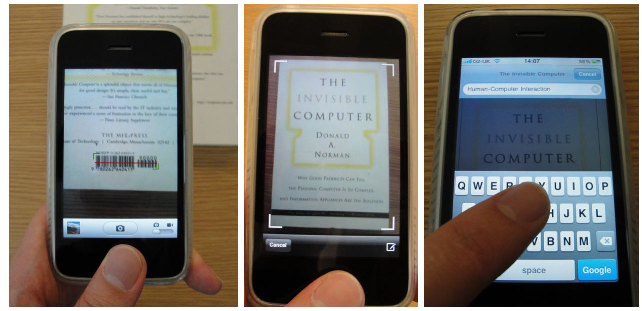
2. Paper Books + Embedded Interactive Interfaces: As digital devices become pervasive, everyday objects are embedded with computing components. Thus, an alternative approach is to embed digital interactive interfaces within paper books. By doing so, paper books keep most of their affordances, with the minimum attachment of digital components for searching. Unfortunately, the technology today is insufficient to implement this concept as a functional unit. The implementation of this concept requires components that do not affect the original book's form factors (like shape, size and weight). Moreover, the components must facilitate user input/output as well as data storage.

3. Paper Books + External Device: The final concept is having books remain paper-based; instead, an external device (like a mobile phone) is used as an interface to perform the information retrieval task. In other words, the external device acts as a module to bridge users' search queries, the books' digital contents, as well as the search results.

A PROTOTYPE OF PAPER BOOKS + EXTERNAL DEVICE

The ubiquity of feature rich mobile phones has afforded us to utilise them for a variety of purposes. As mentioned above, for our current context of augmenting a paper book with content searching facilities, such mobile phones pose simple yet powerful interaction experience. Users can maintain and enjoy all the physicality of a paper based book, yet receive the most desirable digital search feature by linking a book with its digital representation through a mobile phone. There are multiple choices to perform this linking operation, e.g., (2D barcode, NFC tags, Object Recognition, etc). Using one of these techniques, a user can submit book's information (like the title or the ISBN) and a search query via a mobile phone, and then the phone transmits the information to an online database for search results. Liu and Doermann [4] adopted this interaction technique for document retrieval. They suggested linking physical and digital documents by allowing users to submit a query picture of the text of a document and retrieve its electronic version from a database; whilst here, our aim is to search for the location of specific information in the document. For example, figure 2 shows a simple application (similar to the one presented by Enrol et al. [1]) where the camera can scan an ISBN barcode of a book to link it with its digital representation to offer digital searching feature. Alternatively, books can be embedded with NFC tags, thus users can use an NFC reader to identify books by holding them close to each other.

This approach of using an external device for searching al-



(a) (b) (c)

Figure 2. An illustration of book content search via a mobile phone. Figure 2(a) shows the application capturing the book's ISBN barcode, while figure 2(b) illustrates using the book's cover page. Either of the methods can be adopted to capture the identity of the book. Figure 2(c) illustrates a user entering a search query after the book has been identified.

lows books to remain fully paper-based, without the need of altering the original interface; thus, all paper-based affordances remain intact. Moreover, the search function is only employed when users demand it; hence, the functionality is optional and it is completely hidden if search is not needed. However, the utilisation of an external device has inherent disadvantages: the functionality depends on the external device to have the required application pre-installed, and it also requires the database to have an indexed copy of the book being searched.

CONCLUSION

In this paper, we have discussed the design spaces to improve reading experiences by augmenting digital functionalities into paper books. We have reported our findings of user needs implicated from an informal qualitative user study by following a human centred design methodology. Our study highlights one specific desirable feature - content searching within books to be available in all reading experience. Consequently, we discuss three design choices with varying degree of digitisation and augmentation. Finally, we present a simple yet practical solution out of these design choices to augment a paper book with digital search services through an intermediary device to ensure a seamless and compelling reading experience.

REFERENCES

1. B. Erol, E. Antúnez, and J. J. Hull. Hotpaper: multimedia interaction with paper using mobile phones. In *MM '08*, pages 399–408, 2008.

2. K. Fujinami and N. Inagawa. An augmented book and its application. In *DIPSO 2008*, pages 52–57, 2008.

3. IDEO. *Human-Centered Design Toolkit*. 2nd edition.

4. X. Liu and D. Doermann. Mobile retriever: access to digital documents from their physical source. *Int. J. Doc. Anal. Recognit.*, 11(1):19–27, 2008.

5. B. N. Schilit, G. Golovchinsky, and M. N. Price. Beyond paper: supporting active reading with free form digital ink annotations. In *CHI '98*, pages 249–256, 1998.

6. A. J. Sellen and R. H. Harper. *The Myth of the Paperless Office*. MIT Press, Cambridge, MA, USA, 2003.

7. J.-I. Watanabe, A. Mochizuki, and Y. Horry. Bookisheet: bendable device for browsing content using the metaphor of leafing through the pages. In *UbiComp '08*, pages 360–369, 2008.

What do U-Note? An Augmented Note Taking System for the Classroom

Thomas Pietrzak
Telecom ParisTech
LTCI, INFRES
pietrzak@enst.fr

Sylvain Malacria
Telecom ParisTech
LTCI, INFRES
malacria@enst.fr

Aurélien Tabard
IT University of
Copenhagen
auta@itu.dk

Éric Lecolinet
Telecom ParisTech
LTCI, INFRES
elc@enst.fr

ABSTRACT
We present U-Note, a new system that helps children to study their lessons. It links a paper notebook and digital documents in order to reconstruct the context of the class. This system makes it possible for students to browse the teacher's documents at the state it was when he wrote the words he is currently reading. The student can also add information he found on internet. We first discuss the interviews we had with teachers, that led to the design of the system. Then we describe the system itself, which consists of a capture system, and a browsing application.

Author Keywords
Augmented classroom, digital pen, Digital lecturing environment, capture and access

ACM Classification Keywords
H.5.2 Information Interfaces and Presentation: User Interfaces—*User-centered design*

INTRODUCTION
As teachers become more fluent in digital media and the cost of laptops, beamers and recording tools keeps decreasing, teachers may have to orchestrate classes that rely on both physical and digital tools and media. As a consequence, experiments with digital teaching that used to be feasible only in universities, can now be experimented with a younger population, in any high-school setting.

In addition to oral explanations and writings on whiteboards, high school teachers we interviewed use presentations, videos, web pages and specialized applications. On the other hand, students still use pen and paper to write their notes. There are many reasons for that: pen and paper is cheap, easy to use and not distractive [5, 6]. Yet, when students review what they wrote during the class in their notebooks, the rich and interactive multimedia experience from the classroom is lost.

Submitted for review to Papercomp workshop.

In this paper we present the initial stages of the design of U-Note, an application that aims at augmenting the students' paper notes by linking them with the documents used by the teacher in the class. We particularly focus on the granularity of the link between documents. For example our system allows to link a word in a notebook to a particular slide in a slide show. Students can also directly interact with digital documents and consult them along with the notes.

Our study began by interviewing secondary and high school teachers. This helped us to define our motivations and to focus on the real needs of users. After reviewing existing systems, we designed a new system called U-Note that provides features required according to our investigations.

INTERVIEWS WITH TEACHERS
In order to better understand how digital materials are used in classrooms nowadays, we visited a high school located in inner Paris. We interviewed three teachers with students ages, ranging from 11 to 21 years old: middle school (11-15), high school (15-18) and cram school (18-21). Each teacher was interviewed separately, for one hour. We focused on their use of paper and digital materials during 'normal' classes, practical classes, and outside of the classroom.

Uses of paper: we identified paper as the primary medium used by pupils for sharing, editing, and knowledge keeping.

Paper notes as a record: students used to write notes on notebooks in every class. In primary school they copy the lesson the teacher writes on the blackboard. In middle school the teacher dictates and write keywords on the blackboard. In high school students take notes upon the teacher's speech. However, teachers are also using modern multimedia equipment such as computers and beamers, but this is done more occasionally as a way to augment lectures with digital documents. These digital materials can be of different kinds, depending on the topic of the lesson: for instance videos for history or biology, or interactive demonstrations for Mathematics or Biology. Unfortunately, these materials are generally not available any longer after the class as they can hardly be printed out and distributed on paper by teachers. Alternatives consists in sending emails containing the multimedia resources, but this is not a widespread practice.

A striking finding of our study is that the main medium used for recording is still paper transcripts even when students do their work by using computers. For example, we could observe a case where students had to perform some exercises on a spreadsheet, but would eventually have to report results and how they solved the problem on paper. An interesting point here is that not only paper makes it easier for the teacher to go through the students' work, but it also makes it possible for the teacher to annotate the students' work and write comments and advices.

Sharing: we also identified sharing ease as an important motivation for using paper in the classroom. Because paper is relatively inexpensive and copy facilitates already exists in high schools, paper documents can be reproduced quickly and distributed extensively. For instance, teachers frequently distribute printouts to complement the teaching-book.

Edition: the ease of combining heterogeneous paper materials together makes it a good candidate for editing and putting together final versions of documents. Students use their notebooks or binders to attach paper documents to their notes. We noticed that teachers often use text tools such as Word for editing text. But this text was often printed and merged with other paper resources copied from an external resource such as books, web pages, or hand-written notes, before being distributed to students or used as an help during the lecture.

Implications for design

we are currently running more interviews to assess our initial observations. In particular we are setting-up questionnaires for pupils and teachers to gather more information on the use of digital tools. Nevertheless, this preliminary work highlighted that paper is still the central medium for organizing information in the class. As said before paper notes do not currently hold any type of digital information. In the following section, we present the system we propose for augmenting paper notes with digital information. Then we will present existing annotation systems that makes it possible to conjointly use paper and digital documents.

U-NOTE

Our first investigations led us to design a system that allows to combine student's notes with teacher's documents. We centered our design around several principles:

- The navigation is centered around the student's notes: the notebook remains his main source of information.
- The link must be as specific and precise as possible: scholar documents are full of information, the student must be able to locate easily the piece of information he is searching for.
- The student must be able to browse the documents: students that misunderstood the lecture may want to freely read them again.

The U-Note system allows the student to reconstruct the context of the class. He can use his paper notebook to browse all the documents shown by the teacher, and add his own data, typically web links, comments or text excerpts that he collected from the web. The system consists of two independent parts.

One part is used by the teacher. It records events occurring during the class. The other part is used by the student to access the documents. The student writes his course on dot-pattern paper with an ANOTO digital pen setup in batch mode. When he is at home he plugs the pen on the USB dock and the strokes are transmitted to the student application (figure 1) to be processed. This application downloads the history of the class from the teacher's server application. Timestamps of strokes and events allows to link them precisely. The student can also interact directly with his notebook with a digital pen in streaming mode. When he points on a stroke, he can access the documents shown by the teacher at this time. The same operation can be done in the student's application by clicking with the mouse.

The teacher's module

Nowadays teachers use multimedia files in their lectures, in particular in secondary school and high school. The purpose of this part of the system is to record the context of the class with high precision. It is composed of several programs and plugins.

We developed a Powerpoint extension that records information like presentations loads, unloads and slide changes. This plugin allows to know precisely which slide of which presentation was shown at a given time. The same functionality is offered for web pages by a Firefox extension. Finally teachers also use audio or video recordings in their lectures. We developed a multimedia player that records load, unload, play and pause actions on these files. All these extensions send events to a central server. This server generates a log file of the class, that is sent along with the documents to the student's application. Finally we also capture the teacher's oral explanations using an audio recorder software running on the teacher's PC. The program allows the teacher to stop the recording, typically when there is rampus in the class. Additionally the system can also take into account events generated by an interactive whiteboard, when available. The teacher's writings on the whiteboard is then made available to the student.

The student's module

The goal of the student's module of the system is to help studying lessons by reconstructing the context of the classroom. For instance when reading his notebook, the student may reach a part he did not understand. Clicking on the notebook in an appropriate way (described later), will make them possible to access all data related to this specific part of the course, *i.e.* (depending on what was actually recorded): the oral recording at this moment, the slide that was displayed at that time, the writing the teacher was making, etc.

Technology: in order to realize this we need a link between the handwritten notes and the events recorded by the teacher's module. Digital pens can not only be used for writing on paper, as usual pens but they are also able to store the handwritten strokes and to send them to a computer. Two

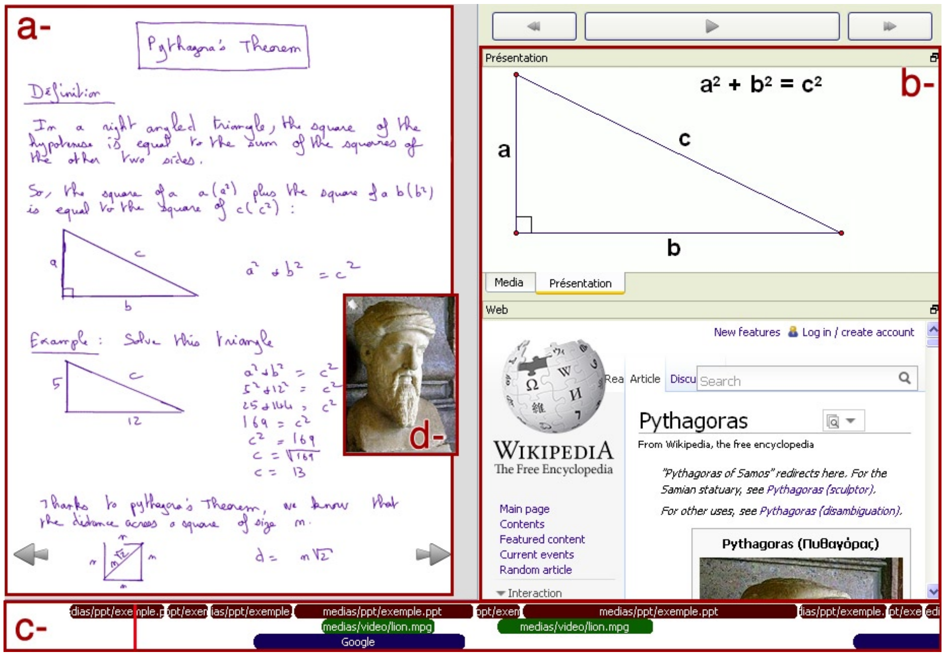


Figure 1. Screenshot of the student's application. a) notebook view. b) miniatures. c) timeline. d) post-it.

main kinds of technologies are currently available as described below.

Several models of digital pen rely on ultrasonic position detection (e.g. Epos[1] and Z-Pen[2]). The user must then clip an USB key on the top of the paper sheet, and use an electronic pen. Any kind of paper can be used as the device captures and stores every stroke relatively to the USB Key. However, the user has to re-clip the USB key every time he starts writing on a new page. This means that he cannot modify previously written pages. This strongly limits possibilities for interaction.

The ANOTO[3] technology allows to digitize strokes in a more natural way. The counterpart is that specific paper must be used together with a dedicated pen. More specifically a dot pattern must be printed on the paper sheets. ANOTO pens offer one or two operating modes depending on the model. In streaming mode the strokes are directly sent through bluetooth and processed on the fly. In batch mode, strokes are recorded in an intern memory, and processed later when the pen is plugged on an USB dock. The important idea is that these strokes not only have a spatial value, but also a temporal value. The temporal value is absolute, so it can be used to link the teacher's events to the student's strokes. In our study we used a Nokia SU-1B model that allows both modes. The student uses the pen in batch mode in the class, and in streaming mode when he is back at home.

The notebook view: the main component is a view of the notebook (figure 1-a). It contains the strokes recorded with the digital pen. This view is useful in the situations when the student's notebook is not with reach. The student will probably prefer to read notes on his notebook otherwise, so that the view can be hidden on demand.

The miniatures: the part of the GUI display makes it possible to explore the teacher's documents while reading the notes. Four kinds of components can currently be displayed in this area (figure 1-b). The first component allows to view a miniature of a Powerpoint presentation. The user can browse the miniatures of the slides, and open the file in Powerpoint. The second one is a web view. The third one is a multimedia player, that can read audio and video files. The fourth one is a viewer for interactive whiteboard.

The timeline: the purpose of the timeline is to provide a visual link between the notebook and the digital documents (figure 1-c). Using the timeline, the user can explore the chronology of events that occurred during the class, and see which slides, webpages and videos were shown and when. The user can click on the items to open a miniature of the corresponding document in the same state than during the class. Pupils can also tap on their notebook with the ANOTO pen setup in streaming mode to open documents that were displayed when they wrote the word under the pen tip.

If the dynamic of the class is important, the student can replay the class from a given point. A red dot moves over the strokes to show what was written and when. The miniatures are updated to show the exact part of the documents shown at this time in the class.

Post-its: while doing his homework or studying his lessons, the student will sometimes search for additional informa-

1 `http://www.epos-ps.com`
2 `http://www.danedigital.com/6-Zpen/`
3 `http://www.anoto.com`

tion on the Web. When he finds some useful information he might want to keep it and paste it into his digital notebook to make a link between the webpage and his lesson. We developed a tool that allows to cut pieces of a webpage and stick them in the digital notebook (figure 1-d).

The Web page capture is based on a Firefox extension. The user selects the piece of webpage with a drag selection. The collected information is sent to the student's module by a Firefox extension. The pieces appear in the student's module as post-it windows. The student can then stick them on one page of his notebook or trash some of them. The page is still active inside, and can be opened in a web browser.

RELATED WORK

Various systems have been proposed for allowing later access to captured live experiences. Ubiquitous Presenter [11] is a classroom presentation tool allowing an instructor to annotate slides with a tablet PC while he shows them and gives the lecture. Students can view the live presentation with narration and digital ink with non-Tablet PCs. The captured presentation can be saved to a Web server and later reviewed as a video by students online. Recap [4] enables the capture of the lecture, slides and teacher's annotations with a higher granularity than Ubiquitous Presenter. Students can access this capture after the class via any activeX enabled web browser. However, neither Ubiquitous Presenter nor Recap provide the capability to link teachers' multimedia datas with student's note, whatever they produced on paper on any type of device.

Classroom2000 [1], which later became eClass [2], is a classroom presentation tool allowing an instructor to annotate slides on an interactive whiteboard. It then links these annotations with a video and an audio recording of the class, and web links opened during the lecture. A longitudinal evaluation of this tool showed the usefulness of links between documents and audio recordings. It also underlined the fact that students take fewer notes when using this system. This is not surprising since the teacher provides his notes. The advantage of this is that students are more concentrated on the explanations. However taking notes has an important role in the memorization process [3] and they integrated a note taking system with pen-based video tablets [9]. Hence, unlike U-Note, these systems require specific equipment and are not adapted in secondary and high school where notebooks are still widely used.

In CoScribe [7], the teacher starts his lesson by giving printouts of his slides. Students are then able to directly create handwritten annotation on teacher's printouts with a specific pen. They also can structure and tag their annotations for a later retrieval. Finally, they can collaborate with other students by sharing their annotations. However, CoScribe uses the printouts as a central media and does not associate the teacher's materials with the notes in student's notebooks.

Finally, various works have been proposed to link digital documents and materials with physical notes taken in personal notebooks. Yeh *et al.* designed a notebook for field biologists [12]. They associated handwritten notes with GPS coordinates and photos they could shot or samples they could find on the field. West *et al.* designed a similar system for scrapbooking [10]. Their system allows to combine handwritten notes with media documents such as photos, videos and sounds using explicit gestures. Finally, Tabard *et al.* proposed Prism [8], a hybrid notebook aggregating streams of digital resources (webpages, e-mails, etc.) with Biologists' notebooks. It long term deployment showed that from all the streams aggregated, users relied on one as their master reference (usually the paper notebook). These works and CoScribe are based on the ANOTO technology.

CONCLUSION

We presented a study about new technologies for classrooms. We began with an interview with secondary and high school teachers. It helped us to identify the usages and the needs. Paper appeared to be still widely used by students to take notes, while teachers use multimedia content in their lectures. The fact is that students have difficulties to link their notes with their teachers' materials. Hence, they cannot get all the advantages of digital contents when studying at home. We presented U-Note, a system that uses the student's notebook as a central media in order to go through captured lessons. It allows the student to access the teacher's materials at the exact state it was when he wrote his lesson.

REFERENCES

1. G. D. Abowd. Classroom 2000: an experiment with the instrumentation of a living educational environment. *IBM Systems Journal*, 38(4):508–530, Dec. 1999.
2. J. A. Brotherton and G. D. Abowd. Lessons learned from eclass: Assessing automated capture and access in the classroom. *ACM Transactions on Computer-Human Interaction*, 11(2):121–155, 2004.
3. K. A. Kiewra. Note taking ad review: the research and its implications. *Journal of Instructional Science*, 16:233–249, 1987.
4. C. K. Kong and J. K. Muppala. Recap: a tool for automated capture and generation of synchronized audio, powerpoint and digital ink presentation. In *Proceedings of CATE '07*, pages 323–328, 2007.
5. S. Oviatt, A. Arthur, and J. Cohen. Quiet interfaces that help students think. In *Proceedings of UIST '06*, pages 191–200, New York, NY, USA, 2006. ACM.
6. A. J. Sellen and R. H. Harper. *The Myth of the Paperless Office*. MIT Press, Cambridge, MA, USA, 2003.
7. J. Steimle, O. Brdiczka, and M. Mühlhäuser. Coscribe: Using paper for collaborative annotations in lectures. In *Proceedings of ICALT '08*, pages 306–310. IEEE Computer Society, 2008.
8. A. Tabard, W. E. Mackay, and E. Eastmond. From individual to collaborative: the evolution of prism, a hybrid laboratory notebook. In *Proceedings of CSCW '08*, pages 569–578. ACM, 2008.
9. K. N. Truong, G. D. Abowd, and J. A. Brotherton. Personalizing the capture of public experiences. In *Proceedings of UIST '99*, pages 121–130, New York, NY, USA, 1999. ACM.
10. D. West, A. Quigley, and J. Kay. Memento: a digital-physical scrapbook for memory sharing. *Personal Ubiquitous Computing*, 11(4):313–328, 2007.
11. M. Wilkerson, W. G. Griswold, and B. Simon. Ubiquitous presenter: increasing student access and control in a digital lecturing environment. In *Proceedings of SIGCSE '05*, pages 116–120, New York, NY, USA, 2005. ACM.
12. R. Yeh, C. Liao, S. Klemmer, F. Guimbretière, B. Lee, B. Kakaradov, J. Stamberger, and A. Paepcke. Butterflynet: a mobile capture and access system for field biology research. In *Proceedings of CHI '06*, pages 571–580, New York, NY, USA, 2006. ACM.

A Tabletop System for supporting Paper Prototyping of Mobile Interfaces

Benjamin Bähr, Sven Kratz, Michael Rohs
baehr@cs.tu-berlin.de, sven.kratz@telekom.de, michael.rohs@telekom.de

Deutsche Telekom Laboratories
TU Berlin
Ernst-Reuter-Platz 7
10587 Berlin, Germany

ABSTRACT

We present a tabletop-based system that supports rapid paper-based prototyping for mobile applications. Our system combines the possibility of manually sketching interface screens on paper with the ability to define dynamic interface behavior through actions on the tabletop. This not only allows designers to digitize interface sketches for paper prototypes, but also enables the generation of prototype applications able to run on target devices. By making physical and virtual interface sketches interchangeable, our system greatly enhances and speeds up the development of mobile applications early in the interface design process.

ACM Classification: H5.2 [Information Interfaces and Presentation]: User Interfaces. – Prototyping, User-Centered- Design.

General terms: Design, Human Factors

Keywords: Paper prototyping, mobile user interface, tabletop, mobile device, participatory design, evaluation

INTRODUCTION

The growing use of mobile devices and the advances in their technical capabilities have opened up a large potential market for mobile applications. This development is promoted by now well-established, professional and centralized distribution channels realized through specific application stores. These stores create vivid competition, dominated by short development cycles.

The development of mobile applications, in comparison to development for stationary use, bears a number of specific challenges, principally rooted in their interface design. Mobile devices are designed for a myriad of circumstances and usage contexts and thereby have to adapt to unsteady user attention and changing environmental conditions. The additionally limiting hardware constraints of the devices and an insufficient control over these factors can lead to strong usability losses, which can jeopardize the application's overall success and performance. User evaluations of interface prototypes are an important strategy in reducing this risk. The user assessments should be preferably conducted during the early design phases to evaluate the key implications of the interface layout in a subsequent implementation.

In the following, we describe the implementation of a specific unobtrusive tabletop system [5] that is developed to simplify and accelerate the design of such early-design-stage paper-based prototypes (PBP). Through minimizing the implementation effort, the system enables designers to collaboratively sketch a number of prototype variants, which can be automatically translated into executable prototypes that are executable and examinable on the respective mobile devices. The approach aims to fulfill the following design principle: Of utmost importance is to provide simplicity and speed to the development process, thereby enabling a creative support environment, which promotes collaborative work. This paper discusses the decisions we made in our design approach, shortly describes the required hard- and software components, and finally details a number of specific interaction factors we realized to support and fulfill these factors.

DESIGN APPROACH

The implemented system offers assistance for the development of prototypes early in the development process, which stands in accordance to the well-established paper-based prototyping approach, in which the evaluation is conducted by test users who interact with paper-sketch representations of the real interface. Our system enables a design team to collaboratively design interface sketches on prepared paper sheets and to virtually add functionality to those, in the form of user interaction elements, for example defining the interrelation of the screens. As a result, the surface system is able to generate a virtual representation of the interface, which can be used by a player installed on a mobile device, to execute and use a functional interface.

Our design approach is based on the concepts of paper prototyping, findings on mobile prototyping, and tabletop-based interaction systems, which we discuss in the following.

Use of a Paper Prototyping Approach

The PBP approach, described in detail in [12], has become an established prototyping paradigm, in which users are asked to fulfill given realistic interaction tasks with paper sketch representations of the real interface. We chose to follow this technique because it promotes our described interaction goals in several ways.

Sketching onto paper is an inherent method to examine and communicate thoughts of any kind and therefore is perceived to be fast, intuitive and uncomplicated, especially if these thoughts are elaborated in a collaborative way. Furthermore, it imposes relatively few constraints on a design and thereby does not impede creativity flow in the way other tools do, aiding project designers to create layouts that are feasible to implement. Part of the design team should have appropriate experience of interface design as well as knowledge of the constraints of the target user interface toolkit, but the PBP method remains very helpful to integrate valuable designers with a less profound technical background into the team through the commonly explored communication base of paper and pen. We adopted this approach to enable fast results from the design process, since every sketch, with an understanding of its functionality, can be used as a testable interface version. Therefore the approach encourages the team to develop a number of further interface alternatives which are then comparatively tested.

It is in a sketch's nature to eliminate details and to take the focus on certain carefully chosen aspects and thereby to create an abstraction of reality. [11] refers to the "less is more" philosophy of many architects to prefer a simple diagram drawing over realistic models to discuss and examine certain aspects rather than to confuse people with complexity and perfection. Generally, a polished interface increases the users' hesitation to critically communicate their experiences. Especially people with limited technical experience generally assess their own personal weaknesses rather than analyzing the interface design in question and therefore are shy to describe their problems and issues with the software. These people will more likely to discuss their opinions, when they are presented with a simple diagram, or even childish looking paper-based sketch interface representation.

Mobile Prototyping

In principal the setup described could be used to prototype any kind of digital interface. We focused our work on mobile applications prototypes, since they are preferable tested on the devices themselves in mobile conditions, rather than in a laboratory surrounding, whereas stationary use applications can be sufficiently tested with the established paper based prototyping scenarios.

The heterogeneity of mobile applications use contexts is pointed out in the work of [9], which introduces a design space including factors such as Locations and Settings of use, as well as the Movement and Posture of the users. Further dimensions are described in specific schemes about the Devices and their Usages, the degree of Workloads, Distractions and Activities and Users and Personas.

To take these specific variables into account, [9] and [13] follow a consequent continuation of the paper prototyping approach in carrying it over to a mobile surrounding. They used dummy versions of the devices in question and built reproductions in accordance to the original's size, weight, and buttons, substituting the screen with interchangeable paper interface representations, letting the dummy device react to the user's behavior. However, some problems of such low-fidelity prototypes are obvious. The reproduced device will lack some properties, which might be important to the evaluation, such as the brightness, reflections or shadings on the screen. In addition, a high extent of Wizard-of-Oz supervision is necessary to react according to the users' behavior and monitor their interactions for the later review.

For these reasons our approach uses mixed-fidelity prototypes, which directly display the designer's sketches, but in a virtual way, running on the specific device. This gives the test users the opportunity to integrate the application use in their regular habits. Thereby the experiments do not only clarify the picture of how users interact with the application in specific use contexts, but are moreover able to identify these fields.

Tabletop-Supported Prototyping

Setting up a tabletop environment in combination with paper sheets as the core interaction element in our system differs from approaches followed in [7] or [10] who use computers and other digital devices to directly sketch prototypes. This is motivated on the one hand by the native use of paper and pen to create sketches, on the other hand by a better support of collaborative design. Gathering around a table to work on and discuss ideas of any kind is an accustomed attitude. This is especially true for design tasks since they demand a high level of communication. A detailed examination on how tabletop interface systems are able to support collaborative work and to mediate group dynamics is done by [6]. These thoughts are followed by [2] in a tabletop application that supports users in collaborative brainstorming sessions. A collaborative design-specific examination of a setup which uses an overhead projector is done by [1], implying the use of paper as a combined input and output channel.

Following a tabletop approach additionally creates the possibility to carefully retain the appearance of the system within the development process. A standard (obtrusive) tabletop system behavior risks limiting the creativity flow of the design team through over-exposure of its hardware [3]. Therefore our technical setup is embedded into a usual working desk environment, allowing its appearance and interaction level to be largely controlled by its users.

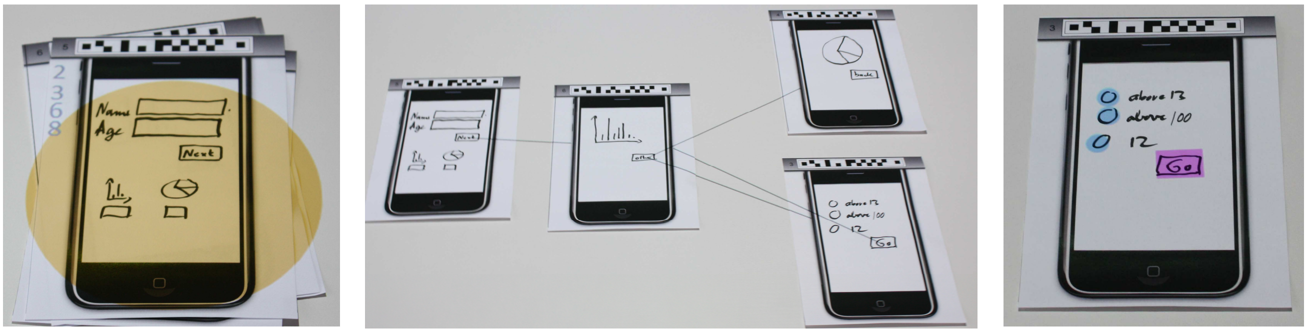


Figure 1: Heap Building Tool(left); Display of Screen Interrelation(middle); Adding Functionality to the physical content(right)

HARDWARE AND SOFTWARE IMPLEMENTATION

We used an unobtrusive tabletop setup, which is composed of video projector, located in a central position above a regular meeting table and three cameras, two firewire CS-Mount digital cameras and a DSLR, which are used to implement the user's input channel for interaction. A Java application running on a PC controls all cameras. One firewire camera is used for infrared finger tracking [8], another for barcode marker recognition [4]. The DSLR is used to take high resolution images of the tabletop surface. Additionally, a printer is included in the setup, which is used to print out physical representations of the user interface sketches.

KEY INTERACTION FACTORS

Due to space constraints there is not enough room to describe the whole paper prototyping tabletop system and all aspects of its usage process in detail. To give an overview, we summarize the design process. The tabletop system supports designers in the early stages of the design process, by allowing them to draw sketches on paper and to specify links between widgets, such as buttons, on paper to other paper sketches. The tabletop system automatically captures sketches as they evolve. It identifies screens by scanning a barcode marker that is located on the top of each sheet (Fig. 1). The borders of the device display are pre-printed on the sheet. There is some white-space around this border to allow writing down notes – either for the designers themselves or for communication with developers in later stages of the design process. All additional semantics, such as links between paper sheets, are projected onto the table. The complete interface can thus be laid out onto the tabletop with the virtual semantics projected on top. From the paper sheets captured at high resolution and the additional semantics specified by the designer, the system then creates a virtualized paper prototype, in which each paper sheet is shown on the mobile device display and user actions on the virtualized prototype can be activated immediately by the test users. The paper sketches are thus useful for designers as well as test users. The prototype can be used on actual mobile devices out in the field and it is thus possible to evaluate the prototype in realistic usage contexts.

The following subsections highlight the key interaction factors, which are all focused on achieving the design principle stated in the introduction.

Focus on physical paper sheets as the interaction center

The center of development consists of ready-made paper sheets in the slightly enlarged shape of the display, on which the paper-sketch representations of the interface can be drawn and edited. Every paper sheet is equipped with a barcode-marker, which allows the system to steadily determine the sheet's current position and rotation on the table. This allows the system to project virtual information onto the paper which is aligned relatively with the paper sheet.

As most prototyping projects will embrace a substantial number of interface screens, design teams will often face limitations of tabletop space. Designers can intuitively handle this problem by arranging spontaneous groups of screens which are then aggregated to heaps. Here the system supplies a navigation aid in monitoring the arrangement of groups and projection their constituent parts, as shown in Figure 1.

Adding functionality to paper

To be able to actually run and test the interface prototype in a later project phase, the single screens have to be equipped with certain user-controls and other forms of functionality.

Controls which trigger the activation of a different screen are of central interest to the development process. Designers are able to define these relations, such that the system optionally marks the connection as a projected visual representation on the table. The projection is updated in real time as the designer moves the paper sheets containing the interface sketches. This permits all the interaction paths to be shown on the table's surface, keeping the interface's storyboard traceable throughout the design process.

Merging physical and virtual interface representations

The content of the individual interface screens is displayed using two methods. Virtually, through the projection of virtual contents on paper and physically, as a result of the treatment of the paper by pen. The two media, pen drawing and projection, are integrated and overlaid to facilitate an unobtrusive and liberated design process. The contents can be merged with each other. A high resolution digital SLR (DSLR) camera photographs the physical interface. This high resolution image is subsequently translated into virtual space. This allows the virtual representation (after potential modifications by the designer) to be retranslated to a physical form by printing it out onto a new sheet of paper. In order to carry out such a virtualization of the interface sketch; the DSLR captures one or more frames. Using the 2D barcode marker as a reference, the image frames are perspectively corrected and cropped to contain only the interface elements. After an interface sketches have been virtualized, the system prompts the user to swap the photographed versions with an according blank paper canvas, on which the virtual representations remain as projections.

The created virtual representations can still be physically edited by pen sketches, which can be layered and integrated into the virtual version. If the system receives a capturing command for a frame which already comes with a digital representation, it blends out this digital content for the image capturing instant, so that an addition of the existing and new frame of the interface can be added to the merged version without quality losses.As already stated, the system can convert the virtual interface back to a physical paper format using printouts. The printed interface representations can then be immediately reintegrated into the design process.

Easy editing through virtualization

The interface's virtualization is not just oriented towards later export onto mobile devices; it additionally creates a number of advantages for the design team, which lie within the digital editing of the content. In a way similar to the operations available in most computer graphics applications, the system allows virtual content to be marked, copied, moved, and pasted.

Thereby highly annoying editing-procedures of can be avoided, such as the subsequent reallocation of interface elements, or the drawing of structures that occur time and again.

FUTURE WORK

Current Mobile Applications increasingly take use of interaction paradigms which reach beyond the use of standard UI- controls such as checkboxes or buttons. The recognition of gestural input via multitouch screens has especially developed to a standard control principle. We plan to apply functionality to the described tabletop system with its infrared finger tracking, allowing the developers to practically define finger gestures on the table surface, which can then be introduced as controls within the running prototype.

CONCLUSION

We have described a tabletop-based system that focuses primarily on paper sheets as an input medium for sketching interface prototypes for paper prototyping. The system can digitize the contents of the paper sheets and permits designers to specify dynamic behavior in their paper prototypes. The system therefore helps interface designers in the process of creating paper prototypes, for instance by making it possible to track changes and recall earlier versions of sketched paper prototypes or to reproduce physical versions of previous instances of sketched paper prototypes. Furthermore, our system enables designers to generate software prototypes for on-device evaluation of interfaces previously sketched as paper prototypes at a very early stage of the interface design cycle. By following a paradigm of unobtrusiveness, our tabletop system is designed such that it does not restrain the creative flow of the designer and consequently the productivity of the interface design process in general.

REFERENCES

1. Hartmann B., Morris M., Benko H., Wilson A.: Pictionaire: Supporting Collaborative Design Work by Integrating Physical and Digital Artifacts. *In Proc. of* CSCW'10, ACM
2. Hunter S. and Mess P.: WordPlay: A Table-Top Interface for Collaborative Brainstorming and Decision Making, IEEE 2008
3. Klemmer S., Everitt M., Landay J.: Integrating Physical and Digital Interactions on Walls for Fluid Design Collaboration. Human-Computer Interaction, 2008, Volume 23 pp. 138-213
4. Kray C., Rohs M., Hook J., Kratz S.: Group Coordination and Negotiation through Spatial Proximity Regions around Mobile Devices on Augmented Tabletops. *In Proc.* of IEEE Tabletop 2008, pp. 3-10.
5. Kratz S., Rohs M.: Unobtrusive Tabletops: Linking Personal Devices with Regular Tables (2009)
6. Morris M., Cassanego A., Paepcke A., Winograd T.: Mediating Group Dynamics through Tabletop Interface Design, IEEE 2006
7. Newmann M., Lin J., Hong J., Landay J.: DENIM: a informal web site design tool inspired by observations of practice. Human-Computer Interaction 2003, Volume 18, pp. 259-324
8. Rekimoto, J. 2008.: Brightshadow: shadow sensing with synchronous illuminations for robust gesture recognition. In CHI '08 Extended Abstracts on Human Factors in Computing Systems, ACM
9. de Sá M. and Carrico L.: Lessons from Earls Stages Design of Mobile Applications, *In Proc. of* Mobile HCI 2008 ACM
10. de Sá M., Carrico L, Duarte L., Reis T..: A Mixed Fidelity Prototyping Tool for Mobile Devices, *In Proc. of* AVI 2008, ACM
11. Schumann J., Strothatte T., Raab A., Laser S.: Assessing the Effect of Non-Photorealistic Images in CAD. *In Proc. of* CHI 96 ACM
12. Snyder C. Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces (Interactive Technologies). Morgan Kaufmann, 1st edition 2003
13. Svanaes D. and Seland G.: Putting the Users Center Stage: Role Playing and Low-fi prototyping enable end Users to Design Mobile Systems. *In Proc. of* SIGCHI 2004 ACM

Using Tangible Symbols with people with profound learning disabilities to access computer based media

NickWeldin **Gosia Kwiatkowska**
The Rix Centre
University of East London
Docklands Campus
University Way
London E16 2RD
nick@rixcentre.org G.Mkwiatkowska@uel.ac.uk

Karen Bunning
School of Allied Health Professionals
University of East Anglia
Norwich,
NR4 7TJ
k.bunning@uea.ac.uk

ABSTRACT
This project was an exploratory pilot study with 4 single cases. The aim was to explore the potential for using paper based machine readable symbols as a method for accessing media on a computer with people with profound and multiple learning disabilities. We were interested in the features of user engagement during a session, and what changes occurred over the 4 recorded sessions.

Author Keywords
Tangible symbols, Learning disabilities, choice, fiducial, Multimedia Advocacy.

ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

General Terms
Design, Human Factors.

INTRODUCTION
The people we worked with in this study were in the range of people with severe learning disabilities [1], and people with profound and multiple learning disabilities [2, 3]. Many people in these groups have a lot of labels applied to them, and while these will never adequately define an individual, we need a collective way to talk about some of the issues they face. It is important to remember that these descriptions are not medical conditions or definitions but a combination of different individual factors for each person. There are no universally agreed definitions for this group, and individuals within it are as diverse and individual as any group of people.

For most people with profound learning disabilities using the computer through a mouse, keyboard and mainstream operating system is not appropriate.

The most common starting point for computer access is using a single switch of some description through a switch box [4] to control some aspect of what is happening on the computer. This could replicate a mouse click to sequence through a simple visual PowerPoint presentation, picture slideshow, or to play games aimed at learning cause and effect.

Once someone understands the cause and effect nature of a single switch, a second switch will usually be introduced. The aim then is to develop the idea of choice between things, but it is often difficult (and expensive) to move beyond a couple of switches to introduce a wider choice. There are a number of ways to navigate choices using one or two switches in software, but they tend to be too complex for people with profound learning disabilities to navigate.

Touch screens are often used, and while their cost is coming down, it is still significant. The lack of physical tactile feedback about where on screen buttons are, and their location changing can make then difficult to use.

BUILDING ON WHAT IS ALREADY THERE
Symbols and pictures on cards are already used with many people in this group away from the computer to encourage and support understanding and communication. These range from home made icon systems, to more formally structured symbol systems that include methodologies about the interactions used with them. An example of a symbol system that includes these is Picture Exchange Communication System (PECS). [5]

Symbols and pictures can be used at a basic level to indicate something that is about to happen, to choose between to indicate a preference, or in more complex combinations to represent more complex ideas if it is appropriate for that person.

As this population face significant issues in learning new things we wanted to leverage any skills they may already have from working with this kind of system in the real

UbiComp'10, September 26–29, 2010, Copenhagen, Denmark.
ACM 978-1-60558-843-8/10/09.

world, and extend them to control and make things happen on a computer.

We wanted to make a system to play media on the computer in response to symbol cards. The system needed to be free, or very cheap (for someone who already had a computer) so that if it worked for someone they could have it, and continue to use it after the study, as well as take it home if they were using it at school.

In order for this to be possible the hardware requirements were restricted to commonly available things, so that a parent or carer could use it for little or no additional cost if they already had a computer

The symbols that the computer recognised needed to work quickly, whatever their orientation and distance (within reason).

WHAT WE USED

We used the widely adopted reactivision computer vision framework [6] for recognising symbols on the backs of cards. Although other symbol recognition systems offered potential advantages in encoding information within the symbol they all had weakness in the speed and ease of recognition for the software in the free implementations. This made them impractical for this group. We also considered using RFID tags, but the lack of availability of over the counter domestic style readers at time was a barrier, and the potential issues of getting the correct replacements for lost tags made their continued use less likely.

We did not propose to use symbols in the usual table based system, as although it has some advantages it has significant access issues for people who use wheelchairs, and failed our criteria of cheap and readily available hardware. The default operation was simply holding up a card so a webcam could "see" them. If you are facing the computer, then holding up the card so you can see the symbol you recognise means the symbol the computer recognises on the other side is visible to a webcam built into a laptop, or positioned above or below the monitor on a desktop computer.

The program used in the study plays a video based on which symbol is shown. The program can play video full screen, or in a window so you can also have the reactivision webcam window visible. That helped some people to learn the process when it was first introduced. The program plays video files stored on the machine, each reactivision symbol has a number associated with it – the program simply plays video 1 if symbol 1 is seen.

We made some prototypes that played web based content, but the delay between showing the symbol and the media appearing and starting to play was significant and variable, making the cause and effect nature of the symbols much more difficult to discern, and so it was not used in the study.

We worked together with staff to identify appropriate content that would motivate people and look at ways that the symbols should be presented for the person they were working with.

They carried out regular weekly session over four months with the person taking part in the study, and once a month we videoed the session and analysed the video. The number of symbols used and the length of the sessions varied depending on the capacity of the individual participants to engage with the activity.

DATA COLLECTION AND ANALYSIS

The staff filled in evaluation sheets for the sessions they ran unobserved. We used two video cameras to record the observed sessions. They were positioned to capture the participant and staff member's expressions and interaction, and the interactions with computer hardware and screen display.

The two videos from each session were synchronised, put side by side, and sampled in 60 second segments with 60 second gaps.

The video samples were transcribed using recommended conventions from discourse analysis, and then coded using interactional/discourse structure (turns and moves). [7]

Repeat coding was carried out by a second person on 30% of the data, and Cohen's Kappa Coefficient applied as a statistical measure degree of reliability.

INITIAL OBSERVATIONS

We are still analysing the video shot at the sites, but have initial anecdotal observations.

The staff working with people were very enthusiastic and excited about the possibilities this offered, and enthusiastically set up and started using the software.

Music videos were used in most settings as they provided strong motivation for students. It also meant that if they presented a card when they weren't looking at the screen they got feedback from the sound.

In one of the school settings symbol cards were already being used away from the computer.

They cut a hole in a desk the size of sheet of A4 paper, stuck a sheet of frosted Perspex under the hole and put the webcam under the table. The students needed to put the cards onto the Perspex to have them recognised. This produced something similar to the way the cards were already being used in the school, but made it less easy for the use to transfer to the home setting from the school. Cards were used with photos of stills from the videos that would play. Both students were able use the system independently after 2-4 sessions. They used six cards, one of which was a control that would play a video of grass, with no sound. The grass clip was never deliberately played.

One student developed an interesting technique for manipulating the symbols, leaving all (except the grass one) lying on the Perspex, and then shuffled them around on the Perspex until the one she wanted triggered the video.

The teacher is now looking at how to use it with the other students in the class, and is planning to use it with the two current users in their review meeting at the school to show videos of what they have been doing this year.

The teacher is keen to develop the system with us to do other things, but for now feels that the enjoyment they get from using it as it is should be left for a while.

In other places people were using one card to understand the cause and effect relationship, but the staff were excited about this being a significant step for the people involved.

Staff needed some support to understand they needed to run two programs at once (the software to detect the symbols and the software to play the videos). Although they were familiar with more than one program running, they were less familiar with the idea of the interdependence of two programs.

People who have less profound learning disabilities who tried the software have more awareness that they are showing something to the computer and tended to turn the card around so the picture they recognise is shown to the computer, rather than the fiducial symbol. They are able to learn to hold it the other way round, but a system that recognised the same symbol that they do would be more simple for them to use, and has some advantages, but is at this stage much more computationally intensive for the computer.

Physically picking up the symbols was difficult for some people, and this may not be the most appropriate system for them, but the low tech nature of the tangible element of the system meant that the person working with them was able to try and implement ways of picking up and selecting the symbols with that person, rather than needing support for that.

The ability to increase or decrease the number of symbols during a session enabled the person offering support to change the degree of complexity (how many choices) without needing to reconfigure the computer, enabling a fluent and confident use by staff who don't feel they are technically inclined. It also enabled them to work in a way they were familiar with the participant to enable them to choose things, rather than the technology overwhelming them.

CONCLUSION

We are unable to draw firm conclusions as this was a pilot study, and the video analysis is not yet complete, but the anecdotal evidence at this stage is promising. The system was used in a range of settings with different people. The staff supporting the people in the study felt that it had been successful and are keen to continue using the system with people and develop other use case scenarios. It was used by people in the study at a cause and effect level, and also to choose what was played. At one site two participants started to use the symbols together to watch videos as a joint activity, an exciting development for the staff working with them as well as the participants.

ACKNOWLEDGMENTS

This work was made possible by funding from the Esmée Fairbairn Foundation

REFERENCES

1. Inclusive technologies description of severe learning disability. http://www.inclusive.co.uk/infosite/severe.shtml
2. Inclusive technologies description of PMLD http://www.inclusive.co.uk/infosite/pmld2.shtml
3. http://www.pmldnetwork.org/what_do_we_want/who_are_we_campaigning_for.htm
4. Example of commercial switch box and switch. http://www.inclusive.co.uk/catalogue/acatalog/inclusive_multiswitch.html
5. http://www.pecsusa.com/pecs.php
6. http://reactivision.sourceforge.net/
7. Jean Carletta, Amy Isard, Stephen Isard, Jacqueline Kowtko, Gwnyeth Doherty-Sneddon, & Anne Anderson HCRC Dialogue Structure Coding Manual (June 1996; 27 pages) Ref. No. HCRC/TR-82

Adapting the TinkerSheet Augmented Papers to Facilitate Student's Reflection

Son Do-Lenh, Patrick Jermann, Guillaume Zufferey, and Pierre Dillenbourg
CRAFT, Ecole Polytechnique Federale de Lausanne (EPFL)
1015 Lausanne, Switzerland
{son.dolenh, patrick.jermann,
guillaume.zufferey, pierre.dillenbourg}@epfl.ch

ABSTRACT

In this paper, we describe how learning scenarios in vocational schools can be enriched by the addition of *reflection sheets*, augmented paper sheets designed to facilitate reflection. This interface, called TinkerSheet was initially developed as a component of a tangible learning environment, enabling apprentices to see visual feedbacks and to control parameters of the system. We reflect on the shortcomings of TinkerSheet to support reflection and how to redesign it for this purpose.

INTRODUCTION

The interface, called TinkerSheet, is developed within the context of a tangible learning environment, the TinkerLamp (Figure 1). It has been used in several classes of logistics apprentices. While the current design of TinkerSheets has been working well as a control interface, our field observations showed that in general it does not provide many opportunities for the apprentices to perform reflection, which is a central mental process in learning.

REFLECTION FOR TINKERSHEETS

We redesign the TinkerSheets to orient the apprentices' efforts into exersising reflection by asking them to compare different layouts, to associate causes and effects of warehouse elements, and to transform computer feedbacks into other types of representation for deeper reflection. These mechanisms "force" them to see the commonalities of different pieces of information, and hence understand beyond the specificities of a particular situation.

We present five dimensions along which a teacher or educational practitioner should consider when designing a TinkerSheet for reflection: location, collaboration, feedback, manipulation and reflection mechanism. These five dimensions are mutually independent of each other and their combinations form the whole design space for different reflections sheets with their own usage scenarios.

UbiComp '10, Sep 26-Sep 29, 2010, Copenhagen, Denmark.

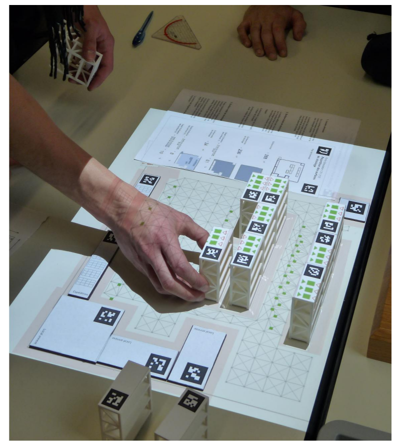


Figure 1. Building a warehouse model with the TinkerLamp. A TinkerSheet is at the far side of the model.

LEARNING SCENARIOS

The TinkerSheets can be adapted based on the design space and used in different learning scenarios across different settings: individual, group, class at vocational school and at workplace. For example, a scenario, called "Fieldwork Reflection by Comparing" is demonstrated as follows.

Current practice: Apprentices are taught about different types of surfaces that are used to design a warehouse, e.g. raw and net surface. They build a warehouse model using the TinkerLamp and explore more about what constitutes each type of surfaces and their impact on work effciency. Everything takes place at school.

Reflection practice: To enable apprentices to relate to those concepts at their workplace, multiple TinkerSheets are printed out at the end of the class, with multiple choice questions and blank radio buttons to answer. The apprentices are asked to bring the sheets to their own workplace, and *compare* e.g. two layouts they build at school according to some certain criteria, and choose the more similar one with their real warehouse using a pen. They bring the sheets back to the next class, discussing with their class.

ACKNOWLEDGEMENTS

This research is part of the leading house "Dual-T". We thank the teachers and logistics apprentices for their support.

A Paper Interface for Code Exploration

Quentin Bonnard
CRAFT - EPFL
Rolex Learning Center
Station 20
CH-1015 Lausanne
quentin.bonnard@epfl.ch

Frédéric Kaplan
CRAFT - EPFL
Rolex Learning Center
Station 20
CH-1015 Lausanne
frederic.kaplan@epfl.ch

Pierre Dillenbourg
CRAFT - EPFL
Rolex Learning Center
Station 20
CH-1015 Lausanne
pierre.dillenbourg@epfl.ch

ABSTRACT
We describe Paper Code Explorer, a paper based interface for code exploration. This augmented reality system is designed to offer active exploration tools for programmers confronted with the problem of getting familiar with a large codebase. We first present an initial qualitative study that proved to be useful for informing the design of this system and then describe its main characteristics. As discussed in the conclusion, paper has many intrinsic advantages for our application.

Author Keywords
Paper interface, code exploration, augmented paper, tangible interface

ACM Classification Keywords
H.5.1 Information Interfaces and Presentation: Artificial, augmented, and virtual realities; H.5.2 Information Interfaces and Presentation: Training, help, and documentation

General Terms
Design, Human Factors.

CONTEXT AND MOTIVATION
Programmers are often confronted with the problem of getting familiar with a large codebase. In a typical scenario, new programmers coming to an institution have to learn about a project in order to start their own contribution. This is usually a difficult challenge. Different strategies are commonly used. The new programmer can browse the documentation, when it exists. This gives a broader overview, but does not allow for in-depth exploration. In a complementary manner, he can follow goal-oriented tutorials. He may also actively learn about the code by fixing bugs or developing unit tests. While usually more motivating for a programmer and useful for the project, this solution does not give a good overview of the overall architecture.

In this article, we present a novel tool for code exploration, based on a paper interface. Paper interfaces intend to provide computing abilities to paper, while trying to keep its simplicity and advantages over electronic devices. They can be used to augment traditional documents, but also create music [2], design warehouses [7], etc. However, programming appears as one of the tasks least adapted for paper interfaces, as it is one of the few intellectual activities that did not exist before the digital age. Indeed, very mature software exist for coding, and text input only makes paper a very poor competitor to a keyboard/mouse/screen system. Nevertheless we believe that paper has a rich set of properties that fits particularly well in the specific scenario we just described.

We describe hereafter Paper Code Explorer, a system designed to take the most of paper for code exploration. This is a Digital Desk-like system [5] and follows in a long tradition of using tagged paper for interaction control [1]. This tool focuses on code understanding and is not meant to be used for testing, debugging or developing. Nevertheless, code exploration plays an important role in any of such programming activities. Several techniques are commonly used to navigate between the portions of code displayed on the rather limited screen real estate: bookmarks, hyperlinks between definitions and occurrences, hierarchical index of the components of the workspace, outline of the displayed resource, tabs, etc. The contexts in which such active exploration has positive learning outcomes have been the subjects of many studies. One findings is the crucial need of a global map/representation to make the best use of hypertextual navigation: allowing flexible in-depth exploration while not getting lost in the process [4]. We believe that paper interfaces are good candidates to offer alternative solution to this classical problem.

In the next section, we describe an initial study involving a new programmer getting familiar with a code library using a mock-up of Paper Code Explorer. Informed by this initial study, we then present the main characteristics and components of our code exploration system.

AN INITIAL STUDY
Before designing the system, we set up a mock up of Paper Code Explorer and used the opportunity of a new colleague joining our team. A member of the team (the expert) was to explain him (the novice) how to start working in our codebase. Naturally, the expert would have walked through the code, and the novice would have asked question as they ar-

UbiComp '10, Sep 26-Sep 29, 2010, Copenhagen, Denmark.
ACM 978-1-60558-843-8/10/09.

Figure 1. Paper sheets and a computer mocked up Paper Code Explorer.

rive. We asked them to use a simple mock-up of Paper Code Explorer, to identify some flaws and validate some ideas. The mockup, shown on Figure 1, consists of small sheets of paper for each of the classes selected beforehand by the expert. Each class was represented by a box containing three boxes: one for the name of the class, one for the list of the name of the fields of the class, and the list of the name of the methods of the class. Both lists were ordered by decreasing visibility of the member (public, then protected, then private). Common paper related tools were placed on the table: sticky notes, stapler, pens, scissors, tape and paper clips. A large sheet of paper was placed as a support. The screen of the expert's computer was projected in front of them, and a regular keyboard/mouse controlled the computer. We interrupted the session several time to ask the novice some questions. This experience was not exactly what Paper Code Explorer is intended for: the expert guided the novice similarly to a tutorial documentation, as opposed to the novice exploring the code alone to build his own representation. This is linked to the fact that the novice only needed to know a small subset of the framework, as he would only work on a restricted aspect (improving a path finding algorithm); the novice does not need to learn everything about the software, only where to contribute.

This informal experiment is not an evaluation, but a first step in the design, which made possible following observations:

- *Paper representation of classes supports large overview.* Before even starting to discuss the architecture, the novice identified a design flow almost immediately when looking at the sheets in front of him: "There is a GetInstance() method everywhere so it's probably useless". (In fact, it corresponds to an abuse of the singleton design pattern for convenience reasons.)
- *Paper interfaces for code exploration should support a flexible navigation system adapted to the different granularity levels of the codebase.* The vertical navigation (from package to line of code and vice-versa) is at least as important as the horizontal navigation (from a class to another). Most notably, the expert started by drawing main components of the software and their relationship to each other, or walked through the main method block of code by block of code. Moreover, the expert navigated mostly using `show definition of` commands.
- *Paper representation of classes should include the visual characteristics of the corresponding source code.* The novice noted that the plain list of members does not give a feeling of the size of the file (in number of lines) as the size of the scrollbar cursor does on the computer. Furthermore, the list of methods does not show how big each method is, which is an important data. The novice further suggested that the ordering of the members by visibility is not very helpful; it would be more interesting to group private functions with the public function calling them. These comments inspired the design of the flash cards shown on Figure 2.
- *Code understanding tools should support visual, active exploration.* An interview with the novice the day after the experiment revealed that he was remembering the size of the classes better than their names. Even if short questions during the experiment showed that the expert's explanations were clear, the novice did not remember most of them the day after, which is another example that passive learning as in a walk through is not effective.

DESIGNING PAPER CODE EXPLORER

This section describes the design choices for the on-going implementation of Paper Code Explorer.

Base Components

Paper Code Explorer is a software meant to be used with an *augmented lamp*, i.e. a projector and a camera above a desk. It uses the ARTag fiducial markers system [1] to track and project an augmentation paper sheets of various sizes and forms. It is integrated in the Eclipse environment[2], which provides a very mature framework to handle code, and is easily extensible and customizable. Java is hence a good candidate as the language of the codebase to explore, as it is a broadly used language, and big open source projects in Java are not hard to find.

Paper Classes for a Broad Overview

Paper Code Explorer uses papers sheets in two ways: as support for objects and as ways of triggering contextual commands. We print flash cards containing the name of the class, the list of its members, and a tag allowing Paper Code Explorer to map the paper to its logical counterpart. Using classes as the unit of paper objects is a good compromise for our code understanding objective: printing line does not scale to big codebases, which is our target, and packages are too coarse for a deep enough understanding. These flash cards could be made out of cardboard or paper, depending on the relation we want to build between the user and the objects. We prefer using cheap, easily duplicable paper flashcards which can be cut, annotated or thrown away without

[1] `http://www.artag.net/`
[2] `http://www.eclipse.org/`

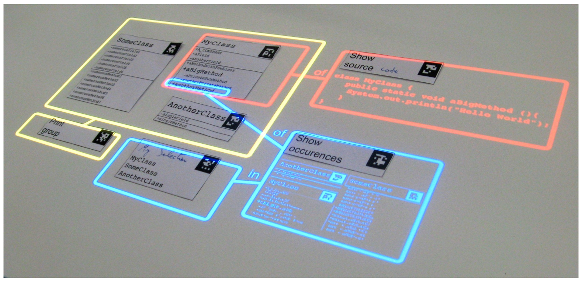


Figure 2. Three classes (MyClass, SomeClass and AnotherClass) are used for this illustration. The paper command `show source` is applied on MyClass, and projects its result in red. The paper command `print group` has been used to produce the paper object representing the three classes circled by the yellow box. The surface taken into account is delimited by a yellow projection. The paper paper command `show occurrences` is applied on one of the methods of MyClass, and the resulting set of class is restricted to the previously mentioned group. The results are projected in blue.

consequence and therefore should allow for more flexible usages.

Using paper classes on a desktop rather than a class diagram on a screen has the main advantage of providing a clearly bigger surface, which is important for such a layout based visualization. This allows for a navigation in the layout that does not require scrolling, which takes full advantage of spatial memory. Paper is tangible, so it is more natural to manipulate than a window with a cursor: moving is easy, several objects (close to each other) can be selected and moved at once. The motivation here is to allow the user to layout the classes according to relationships that make sense to her, and offload the working memory of these relationship onto the spatial organization.

Augmenting the Paper for In-depth Exploration

The list of the members of a class is usually not sufficient to understand it; we need to be able to show the code of the class or its documentation. The lamp can provide for such an augmentation of the objects, as shown on Figure 2 The scenario is comparable to a menu-based interface: the user selects selects an object and a command to apply on it. In our case, we put a command paper sheet corresponding for instance to `show the documentation of` or `show the source code of` a nearby object. A pop-up is then projected with the corresponding content, which can be scrolled by moving up and down the object paper relatively to a fixed, projected reference point.

On this aspect, the interaction zone is limited to the area covered by the lamp, which is comparable to the one of a screen. However, the advantage of a paper interface is that the menu can exist outside of this area. The user does not have to find a compromise between accessibility of various actions in menu and the space allocated to objects. More over, the same manipulability advantage of paper applies to actions: they can be organized and grabbed easily. Command results being linked spatially to them can be moved in an equally easy manner.

Manipulable Queries

The chosen granularity of the classes does not mean that command inputs can not have a finer granularity. Let us consider the command `show occurrences of` a member of a class (a method or field). This command is augmented with a pointer projected at a fixed position relatively to the paper. This precise pointer can be manipulated as easily as the paper, and allows a communication with Paper Code Explorer as precise as a mouse. It can also be interesting to apply actions on augmented content, e.g. `show definition of` a variable in the projected source code of a class.

Actions are not limited to unary operators. For example, it is important in our scenario to show the relationships between several classes: inheritance, aggregation, function calls, etc. Code can be considered as a semi-structured database which can be queried [3], for example on private methods returning a String and using a given member. We make such queries tangible and manipulable: the user can modify them easily (changing the input or the parameters) and observe the changes in real time. For example, a query on all occurrences of a method can give too many results, so the user can restrict them to occurrences within a given class. If this is too restrictive, the occurrences can be restricted to a wider set of classes. All these modifications on the query happen by incrementally adding and removing paper objects as a feedback to the result.

Also, some queries can have a result mentioning classes that are not in the augmented area. In this case, we use the fact that printed paper have a fixed text layout which can be easily remember. To be more concrete, let us consider the command `find classes using` a given member. When used, it projects thumbnails of the corresponding flashcards rather than the full flashcard in order to spare the display area. The user is maintaining a spatial arrangement of the printed classes on the side. This way she can match the form of the projected thumbnails with the one of the classes displayed on the side. Of course, other techniques can replace thumbnails if they are not adapted to the size of the codebase. The goal is not to remember the fixed layout of the whole codebase, but rather offload the working memory using features of the human vision.

Active Reading Using the Paper Interface as Information Support

A musician or a writer, for example, annotate heavily the document they are working on. This behavior can be found in most, if not all processes involving documents. On this topic, source code does not appear as a document: it is the same before and after spending time to understand it. Code can be commented, but these comments are usually not personal understanding notes.

Compared to screen-based exploration, it is clear that paper interfaces offer a much larger variety of tools to read in an active manner. In our scenario it is very easy to annotate the printed code in much the same way one would annotate a printed article. The flash cards corresponding the classes can be underlined or highlighted in any color in a natural way, free text can be written simply (consider for example writing the mathematical formula computed by a method), free forms can be drawn, etc. In addition, a reader can bookmark, highlight, link elements, underline or circle. The issue of extracting annotations have been addressed already [6], and the extracted data can easily be associated to the digital content used to generate the paper object.

To go one step further, one can save a layout by pasting the papers on a sheet. This creates a new composite object, that in turn can be annotated, named for a faster recall, linked with one another, etc. Alternatively, the paper classes can be stacked, folded, cut or teared apart.

A Printer on the Desktop for Memory Cycles

There are 3 layers on Paper Code Explorer. The printed layer forms the base of the interface, the augmentation displays digital information on it, and the user writes on the printed layer, possibly using the augmented information. The difference between the printed and written information is that the printed information can be duplicated easily. The interesting point is that these three layers can be merged into a printed or an augmented layer. This allows for a physical or virtual snapshot of the interface/memory, respectively. Such snapshots are useful for versioning the exploration work: they allow the user to save the state of the interface, e.g. save a grouping of the paper classes in case the new one is not as good. They are also useful for recovering from interruption. Moreover, the physical snapshot are paper objects themselves, and can be annotated too: giving it a title for example helps the interruption recovery furthermore.

To integrate the Print-Augment-Write (PAW) iterations in the workflow, it should be as easy as possible to create a new paper object. These new objects have to be usable by the user and by the Paper Code Explorer. To do so, they are assigned a tag so that the system can map the paper object to its information. A printer allows such a controlled creation process. Receipt printers are very adapted to our case: they are relatively small, and can be placed on the desktop, making them reachable but not too invasive. They can achieve sufficient speed (e.g. 7 inch per second) and receipts are not valuable per se (they are valuable if they prove the payment of something expensive, but are discarded in all other cases), removing the restraints a user could have to print and use temporary documents.

Practically, Paper Code Explorer allows to create a paper object representing a group of other paper objects. This group can be used as an alias of its content in a manipulable query (e.g. `show the relationships between` this group), or as a manipulable abstraction (e.g. a module to relate to other modules). The creation of new paper objects can be triggered with paper commands, such as `print` one of the classes shown in the result set displayed by a manipulable query. Paper commands can be duplicated too, or even combined, allowing a user defined menu of commands. The printer also allows to give more freedom in the starting point of the exploration: printing all the classes would not be very scalable for example; it is more interesting to start with an overview in which the user *zooms* by printing the details of chosen elements.

RELEVANCE OF PAPER-BASED INTERFACE

Although the definitive implementation of the Paper Code Explorer is still on going, we can already discuss the relevance of paper-based interfaces in this context. The manipulability of paper is an excellent way to navigate in a complex system such as a software architecture. The fact that paper remains visible outside the interaction zone eases the access and organization of commands. Complex objects such as queries can be built and modified intuitively. Code can be annotated freely with a pen, which, among other things, helps a lot the activity of reading for understanding.

One of the main objectives of paper and tangibles interfaces consists of augmenting the functionalities of a physical object without reducing its simplicity of usage and original advantages. In Paper Code Explorer, most of the fundamental advantages of paper are preserved. However, the augmentations do not offer the same level of interactivity than a traditional computer interface. Our system focuses on code exploration and does not support the input of code. The resolution of the projected augmentation or of the pointers is not as high as those of a screen or mouse, respectively. Nevertheless, we believe that the intrinsic benefits of a paper-based interface for learning justify this compromise at the interactivity level. In order to evaluate this choice more thoroughly, we intend to perform a comparative study with a tabletop display interface.

REFERENCES

1. T. Arai, D. Aust, and S. Hudson. PaperLink: a technique for hyperlinking from real paper to electronic content. In *Proc. SIGCHI conference on Human factors in computing systems*, pages 327–334. ACM, 1997.
2. E. Costanza, M. Giaccone, O. Kueng, S. Shelley, and J. Huang. Tangible interfaces for download: initial observations from users' everyday environments. In *Proc. CHI EA 2010*, pages 2765–2774.
3. E. McCormick and K. De Volder. Jquery: finding your way through tangled code. In *Proc. OOPSLA 2004*.
4. J. Rouet and A. Tricot. Task and activity models in hypertext usage. In *Cognitive aspects of electronic text processing*, pages 239–264. 1996.
5. P. Wellner. Interacting with paper on the DigitalDesk. *Communications of the ACM*, 36(7):87–96, 1993.
6. Y. Zheng, H. Li, and D. Doermann. Machine printed text and handwriting identification in noisy document images. *IEEE transactions on pattern analysis and machine intelligence*, 26(3):337–353, 2004.
7. G. Zufferey, P. Jermann, A. Lucchi, and P. Dillenbourg. Tinkersheets: using paper forms to control and visualize tangible simulations. In *Proc. TEI 2009*, pages 377–384.

Design of a Modular Architecture for Integrating Paper and Digital Document Management Systems

Matthew Jervis
Department of Computer Science
The University of Waikato
Hamilton, New Zealand
mjervis@cs.waikato.ac.nz

Masood Masoodian
Department of Computer Science
The University of Waikato
Hamilton, New Zealand
masood@cs.waikato.ac.nz

ABSTRACT
Paper documents continue to play an important role in many of today's offices. This leads to the coexistence of both paper and digital documents, with each type typically being managed using completely separate systems. There is therefore a need for systems that bridge the gap between the two realms of paper and digital document management. We have developed a paper document management system, called SOPHYA, which provides mechanisms for a seamless integration with conventional digital document management systems. This paper describes the modular architecture of SOPHYA, and demonstrates its flexibility in supporting development of different types of client applications.

Author Keywords
Document management, paper documents, digital documents.

ACM Classification Keywords
H4.1 Information systems applications: Office Automation;
H3.6 Information storage and retrieval: Library Automation

General Terms
Design, Human Factors, Management.

INTRODUCTION
Paper documents have a number of affordances that are not provided by digital documents; such as fast and flexible navigation, ability to read over multiple documents at once, etc. [9]. Until digital alternatives to paper can provide these affordances, it is likely that paper documents will continue to be used along with digital documents. It is therefore crucial to develop document management systems (DMS) which integrate organisation of digital and physical documents in a seamless manner to support better document workflow.

Various systems have been developed for tracking of physical documents, while providing some degree of connectivity with existing digital DMS. Most of these systems use RFID tags to augment real-world artefacts. One such system by Arregui et al. [2] uses RFID readers at various locations around the office to scan and track tagged documents. This system has been deployed in a patent office by O'Neill et al. [6]. Hark et al. [3] also describe a similar system, in which RFID tags are printed directly onto documents using e-ink technology. Other examples include research by AbuSafiya and Mazumdar [1] who propose a model that incorporates paper documents into a digital DMS using RFID, and Raskar et al. [7] who use active RFID tags, augmented with photosensing capability, in conjunction with a handheld projector that when aimed at the tags is able to determine their relative location and project visualisations onto them.

We have developed several alternative versions of a DMS called SOPHYA, which has been more fully described elsewhere [8, 4, 5]. SOPHYA is a technology for augmenting paper DMS to allow them to be integrated with digital systems. Unlike other systems referred to here, SOPHYA utilises wired communication rather than RFID for tracking physical documents. A wired system allows power to be supplied to electronic components attached to physical artefacts, so that they can provide more advanced functionality. This is clearly not possible with passive RFID systems, and active RFID systems require the use of batteries.

A full discussion of the implementation details of SOPHYA is beyond the scope of this paper. Here we describe the modular architecture of SOPHYA, and demonstrate its effectiveness in providing the necessary mechanisms for development of client applications which can be connected to SOPHYA to allow management of physical documents, while also integrating with conventional digital DMS used for organisation of electronic documents.

ARCHITECTURE OF SOPHYA
Figure 1 shows the architecture of SOPHYA, which is split into five layers: three of them belonging to SOPHYA (hardware, firmware and middleware components), and two that are application specific (digital DMS server, and the clients).

The hardware component manages the *containers*, which hold a collection of documents (e.g. folder, archival box) rather than the individual documents themselves. Containers can be placed in *physical storage locations* (e.g. filing cabinets, shelves, in-trays), which can be either ordered [5], or unordered [4]. Both the container and physical storage lo-

UbiComp '10, Sep 26-Sep 29, 2010, Copenhagen, Denmark.
ACM 978-1-60558-843-8/10/09.

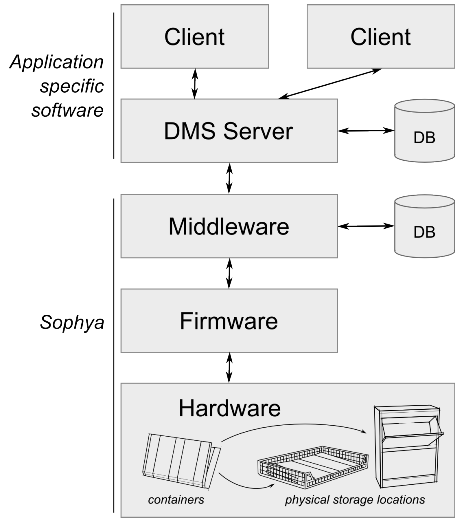


Figure 1. Overview of the architecture of SOPHYA.

cation are augmented with electronic circuitry. These components will be different depending on the type of SOPHYA technology used (e.g. ordered or unordered). However, each container would have a unique ID, and may have optional user interface components such as LEDs.

The firmware is the software embedded into each physical storage location which gives it a unique ID, and allows it to communicate with the containers (e.g. read their IDs and control their user interfaces), as well as communicating this type of information to the middleware.

The middleware, on the other hand, is responsible for dealing with data coming from different physical storage locations attached to SOPHYA, and presents this information to any application specific DMS server using it. The middleware maintains its own database to keep track of information related to individual containers (e.g. their IDs, location).

It is important to note that within the internal layers of SOPHYA there is no concept of digital documents. All that these layers are aware of is that there are a series of containers and locations, each with their own unique ID, and where each container is located. This type of information can, however, be queried by the application specific software which is then responsible for mapping it to digital content it is dealing with. This separation of the physical management of documents from the digital management of content associated with those documents makes it possible for a range of client applications to be developed. The separation also makes it possible for the internal components of SOPHYA to be modified without requiring the modification of the application specific software. We have already developed two versions of SOPHYA, ordered [5] and unordered [4], which can interact with application specific software seamlessly.

SOFTWARE INTEGRATION

As mentioned earlier, SOPHYA has been designed to facilitate integration of application specific DMS and clients with its middleware component. DMS software can connect to the middleware to receive information about the physical storage locations and containers. Software connecting to the middleware can opt to subscribe and receive events (e.g. containers added or removed), as well as being able to query for specific information when required.

The interface between the middleware and the application is, by design, abstracted from the hardware for reason of modularity. The type of SOPHYA hardware used in a particular setting would be dependent on the physical document management requirements of that specific setting. For instance paper documents storage and retrieval needs of a small law firm are radically different from the requirements of a library that needs to handle a large number of books; and as such the type of SOPHYA hardware used in each of these settings would be different. In a law firm it would be sufficient to know where a document is, and so an unordered system [4] might be all that is needed. In a library, on the other hand, it is also important to know the physical order of the books on shelves, what is before and after a book, etc., so a more advanced ordered system [5] would be more suitable.

Therefore, the information that can be queried from the middleware may not be possible with all hardware configurations, and as new hardware platforms are developed more queries may be supported by the appropriate middleware. However, an important requirement is that all different types of middleware support a set of basic functionality (i.e. event notification and queries) and they degrade gracefully when a given query is not supported by a specific hardware. The following sections describe these basic events and query types.

Events

There are a number of cases where the DMS software may want to be notified of events by SOPHYA. For example, when a folder is placed in a user's physical in-tray an alert could be sent to their email if they are out of the office. Currently there are two possible events, addition or removal of containers, to which a DMS can subscribe. The application can receive notification when a container is added to, or removed from, specific locations or across the whole system.

Queries

To allow the application to get information about the locations and containers, the middleware provides a virtual database which the application software is able to query. This is divided into three virtual tables, one for locations, one for containers and one for event history. These are virtual in that they do not exist in an static sense, but rather the information is gathered dynamically when a query is received. Table 1 lists the fields provided by each table.

This virtual database gives access to all of the information SOPHYA is currently able to provide about the physical artefacts it manages. By querying this database it is possible to answer questions such as: *Where is container x located? Which containers are at location y? Which containers are not currently present at a location? Which containers have been moved since time z? How long has container m been at*

Physical Storage Locations	
ID	the unique ID of the location.
Type	the type of location (e.g. filing cabinet, document tray, etc.).
Last Event	the ID of the most recent event at this location.
Last Accessed	the time of the most recent event at this location.
Containers	
ID	the unique ID of the container.
Type	the type of container (e.g. folder).
Location	the ID of the current location of the container (or null if it is not currently present in the system).
Position	the current position of the container in its present location (if available).
Last Event	the ID of the most recent event this container was involved in.
Last Accessed	the time of the most recent event this container was involved in.
Events	
EventID	the unique ID of the event.
Type	the type of event (e.g. added or removed).
Time	the time at which the event occurred.
ContainerID	the container which this event involved.
LocationID	the location which this event involved.

Table 1. Virtual tables provided by the SOPHYA middleware.

its current location? Which containers are in the vicinity of container n? Where has container i been?

EXAMPLE APPLICATIONS

SOPHYA provides low level information about document containers (e.g. folders, books) and their location, which then needs to be processed by application specific software designed to support the requirements of a particular setting. As mentioned earlier, such application software is likely to integrate with conventional digital DMS. To demonstrate SOPHYA's flexibility in this respect, we provide two demonstrative examples of such applications.

Library Application

Libraries are a good example of a scenario where management of physical artefacts (e.g. books, journals) can benefit from integration with digital systems. Most libraries have a combination of digital cataloguing and lending systems, which are separate from the mechanism used for storage and retrieval of items; the only link being the cataloguing label attached to individual items. Because of the lack of systems for digital tracking of artefacts, or the lack of sufficient integration between the digital and physical management systems, most large libraries often have problems with items being misplaced or lost permanently.

In this particular case an integrated digital library system can be developed which would rely on an ordered version of SOPHYA [5] for physical tracking of library items in terms of location, ordering on shelves, etc., while being linked to the cataloguing and lending systems. Figure 2 shows the architecture of such a system. On the hardware layer, items such as books are augmented with container circuitry, and shelves are augmented with physical storage location circuitry. Their firmware then allows them to communicate with the SOPHYA middleware, which maintains its internal database as described previously.

The library management system, on the other hand, is responsible for getting items' location information from SOPHYA, and cataloguing and lending information from the existing library system, and making them available to client software. The *loan database*, keeps tracks of items that are on loan, the *catalogue database* provides catalogue information about the items (e.g. title, author, etc.), and SOPHYA's middleware provides dynamically changing information about the current location of items in the library.

Clients access integrated library collection information through the library management software. There may be different clients for different purposes, allowing different levels of access. For example, the librarian client would allow librarians to access loan information for all library patrons and add entries to the catalogue database, while the patron client would only be able to view information about books and loan information for the logged-in user. Other clients could include on-shelf visualisations, remote browsing, etc.

Design Office Application

Another example of a typical kind of office document workflow that could benefit from the integration of physical and digital document management systems is demonstrated using the following fictitious scenario, which is actually based on our observation of an existing graphic design office:

The workflow begins with sales representatives in the field getting jobs from clients. A "job sheet" is then filled in for each job, and is brought back to the office along with any related hardcopy material. All material relating to a job is placed in a "job-bag". Information from the job sheet is then entered into the job management software and the job is assigned a unique ID. The job-bag goes to whoever is working on the job, and may get passed around if more than one person needs to work on it, though only one person can have it at a time. Job-bags for jobs that are incomplete, but not currently being worked on are stored on shelves at the centre of the office. These shelves provide a quick visual indicator of how much work remains to be done.

This workflow process could be supported using an unordered version of SOPHYA [4], as shown in Figure 3. The physical in-trays and desktops of the designers are augmented with physical storage location circuitry, and the job-bags become containers. Other than the physical artefacts being managed, SOPHYA would function in the same way as in the library example.

The differences appear on the application specific software side. In this case SOPHYA is integrated with existing job management system currently used by the example design

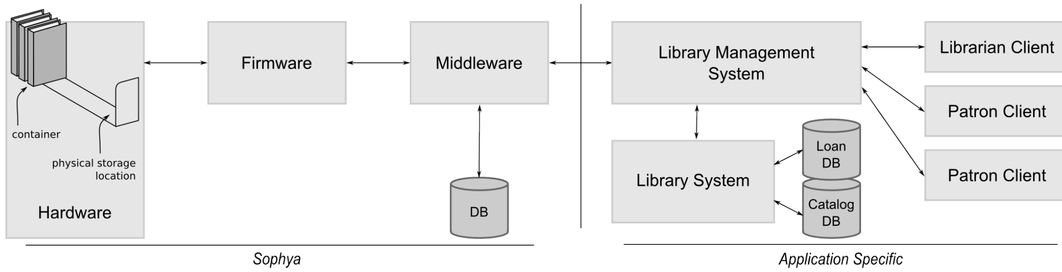


Figure 2. Library application.

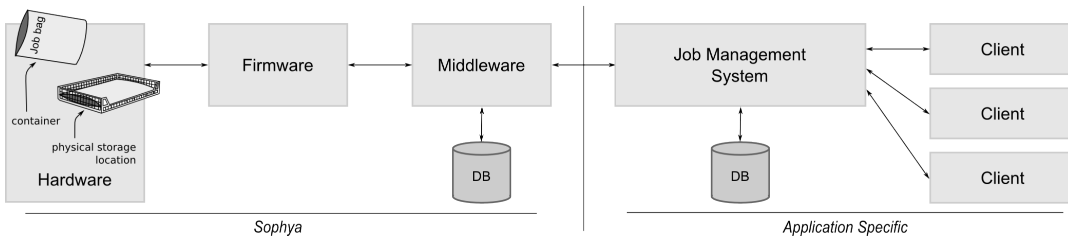


Figure 3. Design office application.

office. As each job (and thus job-bag) is already assigned a unique ID, it is simply a matter of creating a mapping between the job ID and the container ID of the augmented job-bag. When entering the digital information about a job into the job management system, the sales representative would place the job-bag on a container reader (similar to an in-tray) to create a mapping between its job ID and container ID.

A more advanced job management system and client software can then be developed to provide valuable information on tracking job-bags as they are processed and moved between different people, making it possible to dynamically view job-bag location, history, digital content associated or needed, etc.

CONCLUSIONS

This paper described the modular architecture of SOPHYA, which has been designed to facilitate seamless integration with conventional digital DMS. The demonstrative application software discussed in this paper illustrated how this integration can be achieved in two radically different case scenarios. We are currently in the process of developing a prototype library management and visualisation client software.

REFERENCES

1. M. AbuSafiya and S. Mazumdar. Accommodating paper in document databases. In *DocEng '04: Proceedings of the 2004 ACM symposium on Document engineering*, pages 155–162, New York, NY, USA, 2004. ACM.
2. D. Arregui, C. Fernstrom, F. Pacull, G. Rondeau, J. Willamowski, E. Crochon, and F. Favre-Reguillon. Paper-based communicating objects in the future office. In *Proc. Smart Object conf., Grenoble, France*, 2003.
3. B. S. Hark, J. H. Choi, and C. S. Leem. A database design of rfid document management system with e-ink technology. In *NCM '08: Proceedings of the 2008 Fourth International Conference on Networked Computing and Advanced Information Management*, pages 14–17, Washington, DC, USA, 2008. IEEE Computer Society.
4. M. Jervis and M. Masoodian. Digital management and retrieval of physical documents. In *TEI '09: Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*, pages 47–54, New York, NY, USA, 2009. ACM.
5. M. G. Jervis and M. Masoodian. Sophya: a system for digital management of ordered physical document collections. In *TEI '10: Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction*, pages 33–40, New York, NY, USA, 2010. ACM.
6. J. O'Neill, A. Grasso, and J. Willamowski. Rfid: enhancing paper documents with electronic properties. In *CSCW, Computer Supported Cooperative Work, Workshop Collaborating Over Paper and Digital Documents*, 2006.
7. R. Raskar, P. Beardsley, P. Dietz, and J. van Baar. Photosensing wireless tags for geometric procedures. *Commun. ACM*, 48(9):46–51, 2005.
8. T. Seifried, M. Jervis, M. Haller, M. Masoodian, and N. Villar. Integration of virtual and real document organization. In *TEI '08: Proceedings of the 2nd international conference on Tangible and embedded interaction*, pages 81–88, New York, NY, USA, 2008. ACM.
9. A. J. Sellen and R. H. Harper. *The Myth of the Paperless Office*. MIT Press, Cambridge, MA, USA, 2003.

Storybook Augmentation through Portable Projectors, 3D Augmented Reality and Anaglyph

Fadi Chehimi
Computing Department
Lancaster University, Lancaster LA1 4XH, UK
chehimi@comp.lancs.ac.uk

ABSTRACT
Many approaches have been proposed to enhance storybooks with some digital artifacts. Some used RFID tags, others used sounds, and many utilized augmented reality. This paper introduces a conceptual system that would augment storybooks with 3D and anaglyph content that is oriented according to the reader's posture and registered right on the paper rather than through intermediary displays by utilizing portable projectors. The system allows direct finger interaction with the projected graphics to create an immersive and entertaining user experience during reading

Author Keywords Paper computing, portable projectors, augmented reality, anaglyph, touch-based interaction

ACM Classification Keywords H5.m. Information interfaces and presentation

General Terms Design, Human Factors.

NTRODUCTION

For centuries, print has been the most reliable medium for documenting information, acquiring knowledge, and sharing thoughts [1]. With the advent of digital technologies however this role has started to drift. Although this is the case, reading printed books is still more preferable to reading digital ones, especially for recreational reading [11]. While digitized books often offer faster content retrieval/search, provide immediate dictionary look up, and encapsulate volumes of information with negligible footprint, they have notably failed to address the affordances printed books naturally deliver [11]. They require dedicated hardware which is generally expensive, not robust enough, and suffers from poor resolution or color contrast which makes reading from screen hard for long periods [10]. Physical books on the other hand are tangible with interactive feedbacks (e.g. turning a page), allow people to scribble on, are easier to navigate through, provide physical reference to depth of content, and so on [12,10]. They involve multiple sensory cues simultaneously (vision and touch) and this is what helps them retain their preference amongst the clutter of electronic alternatives.

For these reasons many trials to "completely replace" the physical artifacts of books with computer-based solutions, such as e-books, have failed so far [11,10]. As a result, the paramount consideration to keep in mind when designing electronic forms of books would be to enhance readers' experience with "real books" rather than to fully replace them digitally [6].

Many researchers have noted this and developed prototypes to overcome the issue. This paper extends their work and contributes to the current state-of-the-art of paper computing with a system that can augment storybooks, or paper in general, with 3D augmented reality and anaglyph[1] content registered and directly projected on the material in hand, rather than through wearable head-mounted displays (HMD) or on side monitors. Additionally, it allows for direct touch-based interactions with its projected content on the pages. The rest of this paper introduces the system we are proposing and highlights its features, the interaction concepts being designed, the possible implementation, and the future work to follow.

RELATED WORK

There are many research projects, and even commercial products, that augment storybooks with some sort of digital enhancements. Some have used spacial audio to create effective imagery and sense of place around books [10]. LeapFrog's LeapPad has utilized digital pens to interact with printed graphics on page [9]. Others have used mounted projectors to project content around books or on blank white paper for users to interact with [8,14] and many experimented with augmented reality (AR) for innovative storytelling approaches. The Magic Book [3] is the first prototype for this. It augments 3D content on markers printed on storybook pages through an AR handheld display or adjacent screen, and it provides a paddle with special marker as a mean to interact with this augmented content. Little Red [12] and The Haunted Book [13] are other two examples of this with the later augmenting 2D graphics instead of 3D that merge with the background illustrations on the marker-less pages of their concept storybook.

PAPER AUGMENTATION

The problems with using AR in its familiar form to digitally enhance physical storybooks can be summarized in the following:

[1] Anaglyph is still or moving pictures where the red and blue channels have been split and then reassembled in slightly different perspectives so that the image appears three-dimensional when viewed through 3D glasses with red and blue lenses

UbiComp'10, September 26–29, 2010, Copenhagen, Denmark.
ACM 978-1-60558-843-8/10/09.

- Using an HMDs or AR portable displays will limit the use of the system to one person at a time. This would eliminate the collaborative experience book reading usually offers.
- Showing the augmented visualization through a side monitor solves the above problem but introduces a flow in the interaction context: the reader's attention is either completely diverted onto the screen making the book void, or it constantly switches between the two spaces interrupting as a result concentration during reading.
- A paddle with a mounted marker is needed to interact with the augmented objects. No direct interaction with fingers is supported.
- The augmented content on the display occludes what is between the reader and the book. This is experienced for instance when the user tries to touch or point at a page or an augmented object but his fingers get covered by the augmentation.

Having considered these limitations, we present in this paper a possible alternative that uses projection, through portable projectors, projector phones or even future wearable projection accessories, to display 3D and anaglyph content directly on book pages rather than through intermediary displays.

3D and anaglyph AR

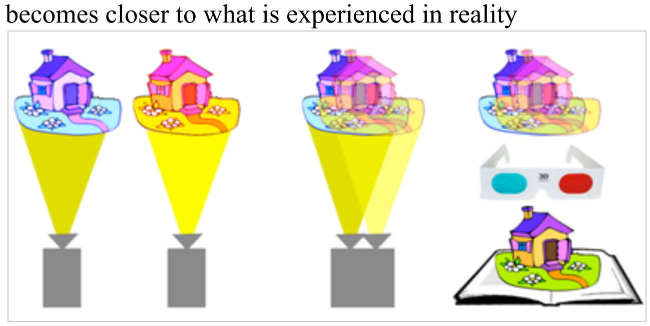
The idea of this is to have the 3D AR content projected straight on the piece of paper or book page that the user is looking at. This makes the augmentation visible on the exact surface of interaction and facilitates multi-viewer collaboration whilst reading. No need for HMDs or goggle-like displays to peek through at the augmented objects. Figure 1 demonstrates the possible look and feel of this technology with the aid of a wearable projector clipped to the ear of the reader.

Figure 1. Story 3D virtual content projected directly on the page in the orientation of the reader through an ear worn conceptual projector

All AR features are preserved here: markers are used to determine the orientation of the augmented content relative to the reader's posture; all virtual objects are registered on the paper in the correct orientation relative to his line of sight; objects can be static or animated and users may interact with them if the system allows; and users' experience with the medium in hand is enhanced by the use of augmentation.

By modifying the implementation of the 3D content being projected, it is possible to create anaglyphs that make the 3D content virtually come out of the page, creating thus a hologram-like effect. The OpenGL code responsible of the 3D environment can define two look-at views for each eye with one missing the red channel, Figure 2a, and the other missing the blue one, Figure 2b. Both then get blended into the final image with a distance corresponding to that between the eyes (approximately 4cm) applied, Figure 2c. The red-blue anaglyph glasses a user would wear for this will then compensate for the missing channels and feed the merged imaged to the eyes creating a realistic pop-up feel for the objects, Figure 2d.

With this we take the AR approach of enhancing books into another dimension that would add more entertaining factors into the reading experience. Also given that the reader will be able to interact with these contents with his fingers, as shall be discussed more later, the whole interaction becomes closer to what is experienced in reality



(a) (b) (c) (d)

Figure 2. Anaglyph implementation; a) left eye view without red channel; b) right eye view without blue channel; c) both views combined with horizontal offset; d) the hologram effect with 3D anaglyph glasses

Paper requirements

To achieve the experience sought after from our system the pages of the storybook we aim to design will be engineered to allow the system to detect its orientation and project content appropriately. The first thing will be to add markers at the upper left and right corners of the unfolded pages to allow the system to identify 1) the reference page and thus the material to project on it, and 2) the posture of the projector (or user) in relation to the book to project content in the right direction. Marker detection and pose estimation are very well established topics in the field of AR and thus we will be using the reliable approaches implemented in ARToolKit [2]. But instead of using black and white markers we will use colored ones to fit the artistic context of storybooks as Figure 3a shows. We will use dark colors (like red, blue, green, dark grey, etc) which would naturally convert to black during the binarization phase of the marker detection pipeline [4]. This will automatically generate the black boundary needed to identify markers, Figure 3b. Light colors like yellow and cyan would drop under the threshold value of the binarization algorithm and thus will be avoided.

The other requirement for pages is to have blank block spaces on them along with text to act as projection boards where the 3D content will display. The content will not necessarily be projected on the markers on page but on this blank space. If it is not present the projection will still take place but will be distorted by the clutter of text and graphical illustrations printed on the page.

(a) (b)

Figure 3. Page layout; a) colored markers and blank space on pages; b) binarized image of what the camera sees with the colored markers converted to black

Having markers printed on pages might look obstructing to their layout coherence and therefore we could explore the possibility of utilizing natural feature detection as in [13] for page recognition and 3D registration. The authors have reported very reliable performance results when using this technique for the purpose of book digital augmentation.

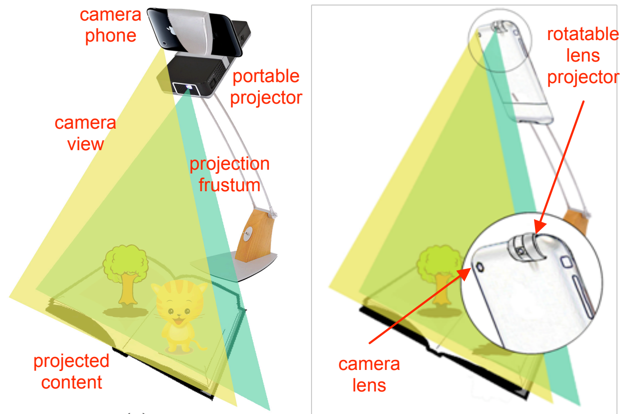
Facilitating hardware

Given that portable and pico projectors are still in their infancy and their capabilities are still limited many researchers have developed substitute prototype devices [14] and even proposed futuristic mobile phone projection accessories [7] to do the job projectors are envisioned to claim. Here we show the prototype we would develop and two other future implementations.

Desk lamp stand

This is our potential hardware implementation and the one possible with the available technologies. It is composed of a lamp stand altered to hold a portable projector and a mobile phone as in Figure 4a, (this could be used in place of a normal reading light). The phone is placed so that its camera points at the surface the projector is pointing at. It then can be used for marker detection and touch tracking on the book pages. Also the accelerometer sensor on the phone could be utilized to help determining the orientation of the projector relative to the book, and hence to the reader.

This design was inspired by the work in [13] but the difference here is that we are embedding the projector and the camera (on the phone) onto the lamp stand. When technology permits it will be possible to replace the two separate pieces of hardware with a composite phone that has an adjustable throw angle projector and a camera both built-in as Figure 4b demonstrates. A potential implementation of this has been introduced in [5].



(a) (b)

Figure 4. Hardware assembly; a) implementable prototype consisting of light stand holding a portable projector and a mobile phone; b) a visionary prototype with a phone embedding a rotatable lens projector and a camera

Ear-clip form factor

Another future design would be having the projector and camera in the form of a wearable ear-set device (as described in [7]), similar to Bluetooth hands-free sets, as Figure 5 shows. The device would (wirelessly) connect to the wearer's mobile phone which would play the role of the computer processing camera images and track fingers and printed markers. The phone also generates the 3D content to project onto paper and communicates it to the wearable device's pico projector for projection. The advantage of this design would be that interaction models based on such projection would become ubiquitous in the future.

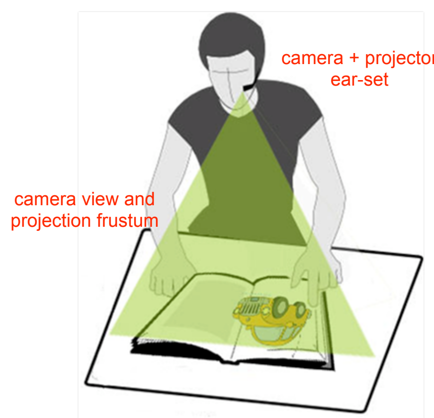


Figure 5. Ear-clip projector/camera form factor [7]

INTERACTION

Unlike existing solutions for AR-based storybooks our system will support direct finger-based interaction with the projected content. The user will be able to select an object for example or move it and drag it from page to page. This is achieved through tracking changes in shadow appearances as fingers move and point. The shadow shape can identify the position, direction and height of fingers and inherently touches. These shadows are naturally enforced by the light emitted by the projector and their varying sizes would determine the distance of the finger from the surface it is aiming at. This is a common technique in projection vision-based systems such as [14] which is found to be very relative to the detection and tracking methods we are planning to implement.

Given that the projector would be either mounted on a stand or worn by the user, hands will be free for gestural interactions on book pages. The fact that shadow-based tracking is capable of identifying multiple fingers at once makes multi touch interactions with projected content possible. As a result the user may pinch a 3D model with two fingers to zoom it in/out, can touch multiple objects simultaneously, and can collaborate with multiple users around.

CONCLUSION AND FUTURE WORK

The concepts presented so far are based on thorough inspection of the portable projection, mobile augmented reality and interactive surfaces research domains. It has been concluded that with the currently available technologies it is possible to implement and deliver most of the ideas introduced in this paper. And once implemented, we believe that it would considerably contribute to AR and paper computing as using animated 3D and anaglyph AR for paper enhancement up to our knowledge has not been discussed before.

Applying this to storybooks is a great way to demonstrate the potentials of the technology and its possible imagination stimulation. As discussed, physical books still dominate when it comes to reading for entertainment and therefore our approach emphasizes their physical characteristics and does not aim to fully transition the reader to a digital reading environment.

It is expected that some challenging issues will arise related to implementing the interactions, to the robustness of the shadow tracking system, to assembling the hardware,, or even to designing the book (having lots of blank space and the need to wear 3D glasses for instance). Usability of the system will be judged and evaluated when implemented and tested. But generally, it is possible to envision the potentials and usefulness of such prototype when used to augment the reading experience.

ACKNOWLEDGMENTS

This work is supported by the NoE INTERMEDIA funded by the European Commission (NoE 038419).

REFERENCES

1. Ahonen, T. and Moore, A., *Communities Dominate Brands*, futuretext Limited, London, UK, 2003.
2. ARToolKit, http://www.hitl.washington.edu/artoolkit
3. Billinghurst, M., Kato, H., and Poupyrev, I., The Magic Book–Moving seamlessly between reality and virtuality, *IEEE Computer Grapics and Applications 21,1*(2001), 6-8.
4. Billinghurst, M., Poupyrev, I., Kato, H. and May, R., Mixing realities in shared space: An augmented reality interface for collaborative computing, *Proc. International Conference on Multimedia and Expo,* (2000).
5. Cauchard, J. R., Fraser, M., Alexander, J. and Subramanian, S., Offsetting Displays on Mobile Projector Phones. *Proc. International Workshop on Personal Projection UbiProjection* (2010).
6. Grasset, R., Dünser, A. and Billinghurst, M., Edutainment with a mixed reality book: a visually augmented illustrative childrens' book, *Proc. International Conference on Advances in Computer Entertainment Technology*, ACM Press (2008), 292-295.
7. Greaves, A. and Rukzio, R., Interactive phone call: exploring interactions during phone calls using projector phones. *Workshop on Ensembles of On-Body Devices at Mobile HCI* (2010).
8. Koike, H., Sato, Y., Kobayashi, Y., Tobita, H. and Kobayashi, M., Interactive textbook and interactive Venn diagram: natural and intuitive interfaces on augmented desk system, *Proc. SIGCHI conference on Human Factors in Computing Systems* (2000), 121-128.
9. LeapFrog, http://www.leapfrog.com
10. Maribeth Back, M., Cohen, J., Gold, R., Harrison, S. and Minneman, S., Listen reader: an electronically augmented paper-based book, *Proc. SIGCHI conference on Human Factors in Computing Systems*, ACM Press (2001) 23-29.
11. Marshall, C., Reading and Interactivity in the Digital Library: Creating an Experience that Transcends Paper. *Proc. of CLIR/Kanazawa Institute of Technology Roundtable,* 2005.
12. Saso, T., Iguchi, K. and Inakage, M., Little red: storytelling in mixed reality, Abstracts SIGGRAPH, ACM Press (2003), 1-1.
13. Scherrer, C., Pilet, J., Fua, P. and Lepetit, V., The haunted book, *Ext. Abstracts nternational Symposium on Mixed and Augmented Reality* (2008), 163-164.
14. Wilson, A. D., PlayAnywhere: a compact interactive tabletop projection-vision system, Proc. symposium on User Interface Software and Technology, ACM Press (2005), 83-92.

ARcetate: Augmented Reality with Acetate Paper

Nicolas Nova
Lift / HEAD-Geneva
nicolas@liftlab.com

ABSTRACT

After fifteen years of academic research, the notion of "Augmented Reality" (Caudell and Mizell, 1992) has caught the public attention and attracted lots of interest from media and mobile phone companies. So much so that augmenting anything now refers to overlaying computer-generated graphics on top of the physical environment.

Interestingly, most of the projects about Augmented Reality (AR in the remainder of this document) are rarely built upon existing techniques to augment and modify people's perception of the environment. Techniques such as distorted glasses, mirrors or kaleidoscopes have for instance enable to produce curious ways to engage people with visual observations of the physical environment. In all those cases, artifacts have different affordances that can lead to a various interactions with the environment. Think about how a kaleidoscope, being used with one eye, can foster a different experience of augmentation.

Back to the workshop theme, we became curious about how a non-digital technology such as paper could be employed to produce Augmented Reality experiences. Adopting a similar approach as Looser's "Magic Lens" notion (2007), we wondered how a technique as basic as paper could enable AR? In order to investigate this issue, we built a physical prototype in the form of a location-based game called ARcetate. It corresponds to a deck of A5 acetate cards represented on Figure 1. Each card is made of three elements:

- A frame that correspond to the main features of the physical environment (building shapes, window frame, etc.). This is meant to help the user positioning the transparent overlay.
- GPS coordinates (latitude and longitude) and conventional bearings that indicate the direction to the North. This information indicate where the user is supposed to stand in terms of position and direction so that he or she is able to position the card correctly on top of the physical environment.
- An arrow that points to a specific feature in the environment that the user should notice to participate in the game.

The game mechanic is similar to geo-caching where participants employ positioning techniques such as GPS to hide and seek artifacts hidden anywhere in the physical environment. A deck of card is created for a specific city and the purpose for each player consists in wandering around the city and locate the places where to use each card. When at a specific location indicated by GPS coordinates, the user holds the acetate overlay in the direction indicated by the bearings and make the card coincide with environmental features represented on the acetate overlay. The goal for the player is then to notice a cue pointed by the arrow (a word, a peculiar object, a specific color) to compose a rebus. Collecting all the cues indicated by the arrows then enable players to complete the game by finding the rebus made of each of the elements pointed by the arrows.

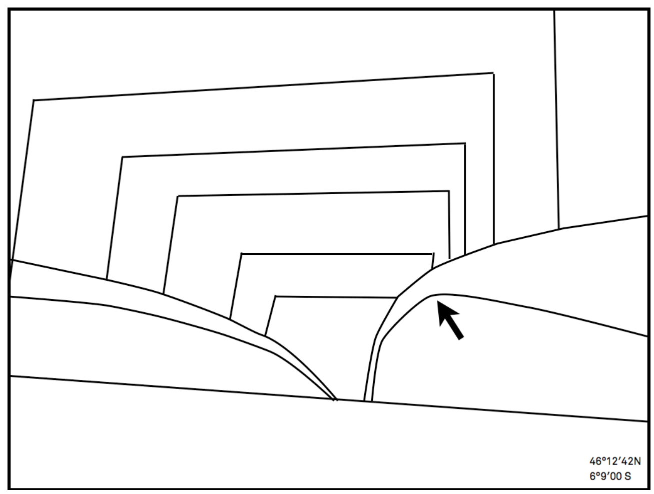


Figure 1. Example of the acetate sheet with the Augmented Elements.

The prototype is currently under development and will be tested empirically to understand how participants can appropriate this kind of activity and how this sort of cards can expand the role of maps in urban environments.

REFERENCES

1. Caudell, T. P. and Mizell, D. W. (1992). Augmented Reality: An Application of Heads-Up Display Technology to Manual Manufacturing Processes. Proceedings of 1992 IEEE Hawaii International Conference on Systems Sciences, 1992, pp 659-669.
2. Looser, J. (2007). AR magic lenses: Addressing the challenge of focus and context in augmented reality, Master's thesis, University of Canterbury, 2007.

UbiComp'10, September 26–29, 2010, Copenhagen, Denmark.
ACM 978-1-60558-843-8/10/09.

Cloth-based Interfaces: Designing for Interactions with Textile Displays

Julian Lepinski
Human Media Lab
Queen's University,
Kingston, Ontario, Canada
lepinski@gmail.com

Roel Vertegaal
Human Media Lab
Queen's University
Kingston, Ontario, Canada
roel@cs.queensu.ca

ABSTRACT

In this paper, we present a user interface for a textile computer display. It allows users to interact in ways that flow from the natural properties of cloth, with an interface that physically conforms to the shape of the object on which a task is performed. While recent work on flexible interfaces has shown promising results, physical properties such as the rigidity of the display remain a barrier to interaction scenarios that are truly physically flexible. We discuss interaction techniques for our cloth user interface, which include gestures such as pinching, draping, stretching and squeezing. Our interaction techniques employ the unique physical characteristics of cloth, including flexibility and shape-taking. We reflect on the system and examine potential directions for future work.

Author Keywords

Textile User Interface, Organic User Interface, Flexible Displays.

ACM Classification Keywords

H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

INTRODUCTION

Traditional user interfaces, such as graphical user interfaces (GUIs), are usually designed for surfaces that are rigid and inflexible. Recent work, such as Siftables [8], and commercial products like the Optimus OLED keyboard, design for the use of small and thin displays embedded into appliances. However, physical properties, such as the rigidity of the display, remain a barrier to flexible form factor scenarios. Interactive cloth presents a relatively recent direction in user interface design. In its flexibility and lightness, interactive cloth has the potential to fit organic form factors in ways that provide more flexible and lightweight interactions with objects of different shapes or forms.

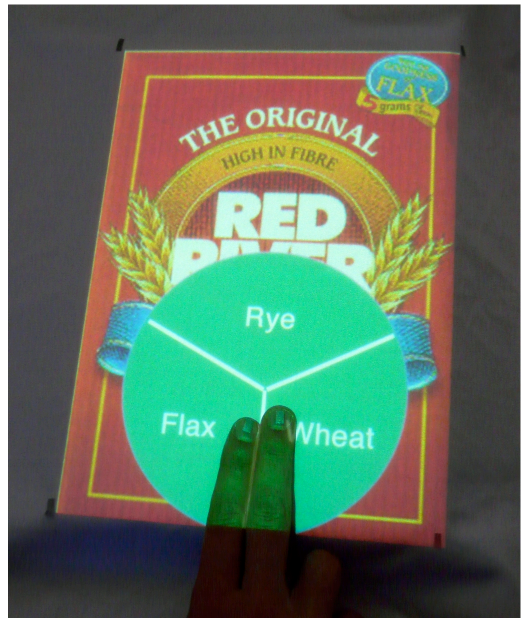


Figure 1. Cloth draped over a cereal box provides an interactive menu to its ingredients in a language of choice.

Rather than enforcing its own shape, cloth takes the shape of underlying objects, allowing it to assume a wide range of forms. Interaction with a piece of cloth draped over an object provides a tight coupling with the object itself. In this paper, we explore how textile displays could be draped over real objects that do not have a display, or that have a form that does not fit traditional displays. As cloth takes the form of an object, affordances provided by the object's shape become part of the available interaction vocabulary. These properties make cloth an ideal interface for interacting with live objects, such as humans.

Physical interactions with, and the properties of, cloth are natural and well understood by users; cloth may be folded, draped, stretched and touched. Interaction styles, such deformations, have logical effects that are easy to comprehend: a piece of cloth grows larger when stretched, and takes the shape of a rigid object when draped over it. This means that a users pre-existing knowledge of the interactive properties of cloth can be of use in developing a cloth interface.

RELATED WORK

Technological Developments

Current research on textile displays, such as Lumalive [12], and Flexible Organic LED displays (FOLED) may provide the back-bone for a future cloth computing device. FOLED displays are thin, light and flexible, and currently provide visual fidelity comparable to a computer display, while having many of the physical characteristics of a thick piece of paper. A review conducted by Sweatman [16] found rapid improvements in the performance of OLED technologies over the past two decades, with decreasing costs. Further work on e-ink, which has seen real commercial release, is another area from which the necessary interface technology may be drawn.

Textiles in Computing

Current research on interactions with physical textiles has largely focused on fabric coloured with thermochromic ink, and has often used the textiles in clothing. Wakita and Shibutani [18] describe a wearable ambient display created with a textile that uses thermochromic ink to change colours based on touch interactions. Berzowska describes her concept of memory rich clothing, by which she means clothing that exhibits its history; in this case, its touch history [2]. These systems evince the universality of textiles and their potential for interactivity in everyday scenarios.

Bergelin discusses using textiles as an interactive surface for a toy known as Spookies [1]. While the first iteration of the Spookies toy uses the textiles to simply cover physical LEDs and buttons, a later version uses woven thermochromic ink coupled with electronics to generate heat and change colours and shadings on the toy programmatically. Similar to Bergelins work, Nack et al. developed small interactive throw pillows [9].

Current textile interfaces largely seek to hide the computing elements of the device and allow users to interact directly with the cloth and objects they envelop. We extend this work by providing a general textile interface that may envelop numerous objects and shapes, rather than a single object.

Paper Interfaces

The Digital Desk system, described by Newman and Wellner [10] was one of the first systems to couple physical paper with graphics projections, blurring the line between virtual and physical, as users interacted with physical paper to perform regular computing tasks. Similar to Digital Desk, in PaperWindows, Holman et al. [7] use standard sheets of paper, augmented with motion-capture markers, to display projected data. This allows for a paper display that can be folded in all three dimensions. The sheets are interacted with using simple hand gestures, such as a rubbing motion used to transfer images from one sheet of paper to another. Guimbretiere [4] developed systems that closed the loop between digital and physical documents, capturing physical input to annotate and modify associated digital versions of these documents.

Finally, work by Usuda [17] and Schwesig et al. [15] have discussed the use of tactile input, such as folding and bending, to interact with augmented physical paper and paper-like devices. Schwesig et al. describe their Gummi interface as a bendable computer. Made up of sandwiched flexible electronic components, users interact with Gummi by two-dimensional positioning and bending of the interface.

Gestured Interaction

Work on contact gestured interaction often focuses on gestured control of touch sensitive surfaces, such as the Smart-skin system described by Rekimoto [14]. Further work includes Han's on a low-cost multi-touch interface [5]. These systems demonstrate the power of coupling input and output, and use generalized gestures that may be broadly applied to numerous application scenarios.

In [13] Piper, Ratti and Ishii discuss the Illuminating Clay system. This system uses a laser scanner to track the topography of clay on a table and projects an interface directly on the clay. Users interact with the system by shaping and molding the clay, and view the system output on the same. The authors argue that their system provides an improved interface for tasks, such as creating and modifying the topography of a space; they operate directly on the topography through the clay, rather than indirectly through devices such as a mouse or keyboard and CAD software.

IMPLEMENTATION

Our system was implemented by optically tracking retro-reflective points on the cloth and on the users hands, and projecting imagery on the cloth from above, using off-the-shelf hardware and custom software.

Tracking

The hardware used for tracking both our cloth and the user's interaction and gestures consists of two eyebox2 cameras [19]. These provide a stereoscopic image of the interaction area from which three-dimensional tracking data is calculated. The eyebox2 is a 1.3 megapixel video camera fitted with an infrared filter over the lens. In the housing that surrounds the camera lens is a matrix of infrared emitting LEDs.

We developed a stereo vision algorithm for image capture and point recognition in the system. The vision algorithm detects bright clusters within the camera image, each of which is a single reflective square.

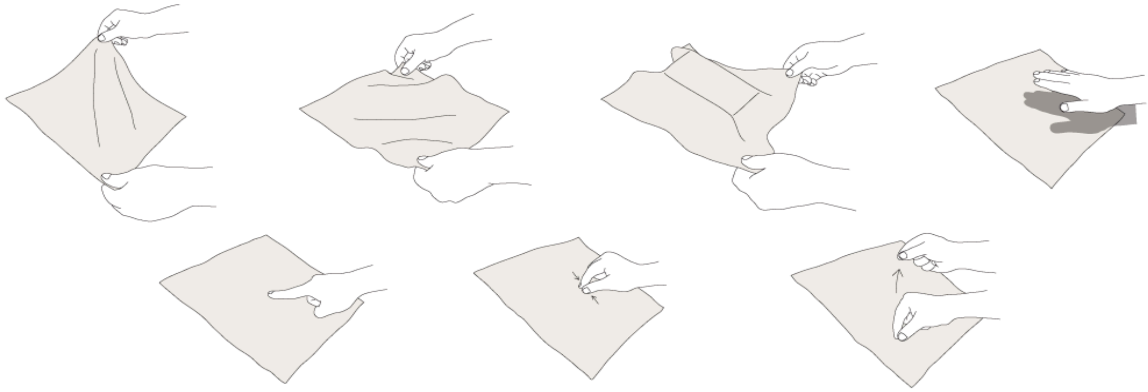


Figure 2: Our generalized set of gestures. Clockwise from upper left: Stretch, squeeze, drape, hover, touch, pinch, and peel.

The individual points tracked on each camera are then used to calculate the height of each point in space. For the purposes of our work it was necessary that the height be sufficiently accurate to detect changes in the physical shape of the cloth, to differentiate between gestures above and physical contact with the cloth. Our height tracking is accurate to approximately 2cm within a range of 30cm above the work area (50cm by 50cm).

Projection System and Physics Engine

While our system tracks numerous points on the surface of the cloth, the actual surface is continuous. For this reason, the physics engine models the surface of the cloth and interpolates to determine the position of points between the tracked points. The physics engine itself models the surface of the cloth as a series of weighted balls, connected by springs. By varying the mass and the spring constants of nodes, we obtained a model that closely approximated our actual cloth. This model deforms naturally, can fold over itself, and can be stretched, returning to its original shape once released. This physics engine was also necessary for simulating virtual cloth responses to user actions, such as the pinch and peel gesture. The hardware used for the display system consists of a BENQ DLP projector, operating at a resolution of 1024x768, mounted above the workspace.

The Gesture Engine

The gesture engine operates on data received when a user interacts with the cloth. Since both the physical cloth and the users fingers are augmented with infrared markers, the 3D locations of all interaction elements are known to the system. Based on predefined gesture routines, the gesture engine monitors user interactions and changes the system state as the user interacts with the cloth. These routines are defined as a set of display states, with transitions between these states triggered by specific gestures.

DESIGN

We will now discuss guiding principles for our interaction design. The development of our interaction techniques drew heavily from the principles of Organic and Tangible Interfaces [6]. Our design principles are as follows:

1. Interactions flow from the cloth metaphor: Users have preexisting ideas about how they interact with cloth, and these should be adhered to as much as possible.

2. Input equals output: Our system should avoid splitting user attention between input device and the display [6].

3. Form follows function: Shape provides valuable information to users as well as system on possible functionalities of the underlying object.

4. Physical or gestural interaction: There are cases where physical touch is not desired, for example, for reasons of hygiene or registration. We therefore included remote gestures, which can be executed while hovering above the cloth.

Gestures

We developed the following six interaction techniques to manipulate information presented on the cloth display:

Stretch: Cloth is stretched from opposing corners, or from two points on the cloth. As a stretch gesture physically increases the size of the cloth, this gesture is typically used to zoom or enlarge graphics on the cloth display. The user's region of interest may be defined by the points held by the users when they executed the gesture.

Squeeze: To perform a squeeze, cloth is pushed together from opposing corners, or from two points on the surface of the cloth. The natural counterpoint for the stretch gesture, squeezing the cloth reduces the visible surface area, often reversing the effect of the stretch gesture.

Drape: Here, the cloth is placed over an object, taking the object's shape. In the drape gesture, the cloth becomes aware of the underlying topography of the surface upon which it sits. The cloth display may adapt appropriately to this contextual information, whether by displaying information about the underlying object or adapting the display area in accordance with the available surfaces.

Hover: A finger or hand is held above the surface of the cloth. Such a virtual gesture allows interaction without disturbing the cloth itself, and may be used in situations where touch may not be appropriate or desired due to the situation of the cloth or underlying objects. The hover gesture often precedes a touch gesture, and indicates attention or interest. Hover gestures may elicit additional information about the area beneath the hover (with a fisheye lens for pictures, for example).

Touch: When touching the cloth, pressure is applied to one or more points on the cloth surface. It is detected by sensing the distance of the finger to the cloth. In simple implementations, touch (along with hover) may duplicate the functions of a standard GUI, and act as a mouse click. This allows a cloth interface to include all functionality presented by a standard GUI.

Pinch & Peel: Fingers are pinched together on or above the surface of the cloth. A pinch indicates that the user is grasping information or elements displayed on the surface of the cloth. A pinch may be focusing action, selection, or the physical grasping of graphical objects on the display. A peel gesture follows a pinch, and may be used to rearrange displayed information or elements, or to remove a layer.

APPLICATION SCENARIOS

Our system has been applied in a number of scenarios that are in the realm of in-situ augmented reality. In these scenarios, our cloth computer acts as a thin contextual display interface to interact with other objects.

Medical Imaging. Our first application is in operating rooms. Here, a surgical drape augmented with markers is projected on to display information, such as X-Rays, directly on the topography of the body. Surgeons can interact with scan data over the body by using the hover and peel gestures to navigate layers of imaging information.

Augmented Objects. Fig 1. demonstrates our cloth display as a reusable interactive display for augmenting everyday objects with high-resolution contextual information. After draping the cloth over the object, our camera determines the type of object through pattern matching its contours. The system then projects a graphical interface onto the object. In this example, users browse the contents and origin of the ingredients of a type of cereal directly via the product label.

Interactive Clothing. We also experimented with presenting data on clothing. This is useful for presenting, for example, data about the physiology of a patient, such as blood pressure and heart rate directly onto the body as the patient moves through a hospital setting. The tracking and projection equipment can be mounted on a bed, or, as long the patient stays within a range of up to 2m, be in a fixed location as well.

Textile Windows. Our final application is a low-cost textile version of PaperWindows [7]. Here, the drape is used to interact with otherwise non-interactive pieces of real paper. For example, after draping the cloth over a paper map of a city the map becomes fully searchable. Users can zoom the map by stretching and squeezing the cloth, and use route finding applications by touching locations on the map.

CONCLUSIONS

In this paper, we presented a prototype cloth computing interface. It allows for contextual interaction with a user's environment by draping a physically flexible cloth display over otherwise non-interactive objects. Future systems will require further developments into flexible, LED based display technologies such as Lumalive. We have presented some applications aimed at providing contextual interfaces to everyday objects that are otherwise not augmented with an electronic display, including surgical drapes, product labels and paper maps.

REFERENCES

1. L. Berglin. Spookies: combining smart materials and information technology in an interactive toy. In IDC '05, pp. 17–23, New York, NY, USA, 2005. ACM.
2. J. Berzowska and M. Coelho. Memory-rich clothing. In CHI '06, pp. 275–278, New York, NY, USA, 2006.
3. K. Cheng and K. Pulo. Direct interaction with large-scale display systems using infrared laser tracking devices. In AP Vis '03, pp. 67–74, Darlinghurst, Australia, 2003. Australian Computer Society.
4. F. Guimbretière. Paper augmented digital documents. In UIST '03, pp. 51–60, New York, NY, USA, 2003.
5. J.Y. Han. Low-cost multi-touch sensing through frus-trated total internal reflection. In UIST '05, pp. 115– 118. ACM: New York, NY, USA, 2005.
6. D. Holman and R. Vertegaal. Organic user interfaces: designing computers in any way, shape, or form. Comm. Of the ACM, 51(6):48–55, 2008.
7. D. Holman, R. Vertegaal, M. Altosaar, N. Troje, and D. Johns. Paper windows: interaction techniques for digital paper. In CHI '05, pp. 591–599, ACM: New York, NY, USA, 2005.
8. D. Merrill, J. Kalanithi, and P. Maes. Siftables: towards sensor network user interfaces. In TEI '07, pages 75– 78, ACM: New York, NY, USA, 2007.
9. F. Nack, et al. Pillows as adaptive interfaces in ambient environments. In HCM '07, pp. 3–12. ACM: New York, NY, USA, 2007.
10. W. Newman and P. Wellner. A desk supporting computer-based interaction with paper documents. In CHI '92, pp. 587–592, ACM: New York, NY, USA, 1992.
11. R.G. O'Hagan, A. Zelinsky, and S. Rougeaux. Visual gesture interfaces for virtual environments. Interacting with Computers, 14(3): pp. 231–250, 2002.
12. Philips. Lumalive. http://www.lumalive.com/.
13. B. Piper, C. Ratti, and H. Ishii. Illuminating clay: a 3-d tangible interface for landscape analysis. In CHI '02, pp. 355–362, ACM: New York, NY, USA, 2002.
14. J. Rekimoto. SmartSkin: an infrastructure for freehand manipulation on interactive surfaces. In CHI '02, pp. 113–120. ACM: New York, NY, USA, 2002.
15. C. Schwesig, I. Poupyrev, and E. Mori. Gummi: a bendable computer. In CHI '04, pp. 263–270, ACM: New York, NY, USA, 2004.
16. D. Sweatman. Organic Devices: A Review. In Micro-Electronic Engineering Research Conference, 2001.
17. H. Usuda. Ultra magic paper interface. In Proceedings of SIGGRAPH'97. ACM: New York, NY, USA, 1997.
18. A. Wakita and M. Shibutani. Mosaic textile: wearable ambient display with non-emissive color-changing modules. In ACE '06. ACM: New York, NY, USA, 2006.
19. Xuuk. eyebox2. http://www.xuuk.com/.

The NeverEndingStorytellingMachine: A Platform For Creative Collaboration Using a Sketchbook and Everyday Objects

Edwina Portocarrero, David Robert, Sean Follmer
MIT Media Laboratory
{edwina, sfollmer, lifeform} @media.mit.edu

Michelle Chung
Harvard Graduate School of Education
michelle_chung@mail.harvard.edu

ABSTRACT
TheNeverendingStorytellingMachine is a paper-based platform for story creation that allows asynchronous viewing and physical editing of shared media at a distance. The system incorporates analog and digital techniques as well as bi-directional capturing and sending of media, offering co-creation among peers whose expertise may not necessarily be in the same medium. A paper sketchbook serves as the primary interface, providing familiarity, portability and personalization. The interface maps pre-created digital media and newly captured analog content onto specific pages in the book through projection, permitting co-edition of the same page simultaneously or independently. Captured images are archived and displayed as a new layer over previous content on any other creation-station, collocated or at a distance. By gathering a history of the compositions, *TheNeverendingStorytellingMachine,* reveals narrative styles and strategies.

Author Keywords Paper Interface, Drawing, Storytelling, Collaboration, Tangible Interface.

ACM Classification Keywords H.5.1 Multimedia Information System.

General Terms Design, Human Factors.

INTRODUCTION

Stories become living agents, allowing us to learn about ourselves through others and about others through ourselves. Storytelling involves the use of our whole body, the exercising of our memory, the recollection of experience. A collective act, it requires a teller and a listener, roles exchanging as the story progresses; we make use of any expressive device to convey the message intended. As such, one particular media does not allow all users to express their vision equally. Some may prefer to capture and share their experiences or imagination through drawing, while others may be more inclined towards the written word, audio, music or collage.

TheNeverendingStorytellingMachine, is a graphical, audio-visual and language interface that invites storytellers of any age to experience, create, share and edit each other's work either in a linear narrative or free-style form, making use of any and every medium they feel comfortable with. By allowing any object to become material for creation, through the use of camera capture, we propose that everything is embedded with expressive potential and hope to incite exploration of our immediate surroundings.

UbiComp'10, September 26–29, 2010, Copenhagen, Denmark.
ACM 978-1-60558-843-8/10/09.

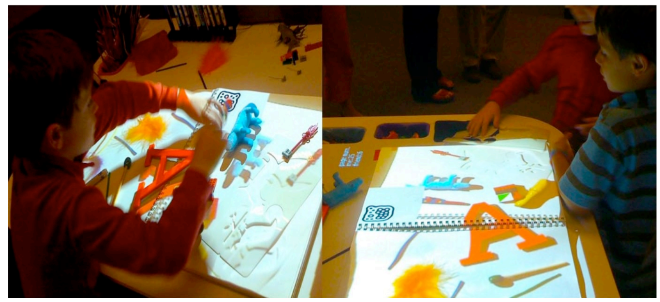


Figure 1. The boy on the left puts down a red, letter "A" block in his sketchbook and snaps a picture, sending the image to be projected onto his friend's sketchbook. The two boys view the resulting image together on the right.

The book format interface, through the use of a regular paper sketchbook, provides a flexible system by allowing a narrative to be developed and revisited. Having a page number and marker as reference gives users the option to work either simultaneously and in the same page or asynchronously.

Intended to be a platform for creative collaborative endeavor, *TheNeverendingStorytellingMachine* is nonetheless further maximized when used as a presentation system – where the input interface and final product are the same. Our goal is that users are compelled to see all the content that has been created prior to them, and build off of this, creating stories with others through different media.

RELATED WORK

Our work is motivated and inspired by a number of projects from the realm of CSCW. Kidpad allows for children to collaboratively draw stories together using on onscreen zoomable interface [6]. Nonetheless we are more interested in how we can allow users to bring in physical media across a distance. Clearboard allows users to draw on a shared surface at a distance while maintaining the ability to reveal each other's gaze and other gestural interactions [7].

However because of its vertical slant and real time capture system, it does not allow for the use of other media such as objects or paint, only drawing. VideoDraw and Double Digital Desk both allow for two remote users to share objects and sketches on two remote physical tables with the use of camera and projectors [11, 12]. Similarly, Sharetable applies this paradigm for remote collaboration between parents and children [13]. However these examples do not have "memory," they only display what is currently on the other side – you cannot easily copy objects or drawings to keep them on screen. Our system also introduces the book metaphor as a way to have more spatially multiplexed content. Also, our focus on asynchronous capture can provide for more artistic content to be produced.

We also draw on work from the field of tangible and augmented interfaces for children. Jabberstamp allows for children to add audio to specific locations in drawings, however it is not shared across a distance and only allows for drawings as input – not objects [8]. Similarly, Stifelman's work allows users to add audio to any physical document [10]. Our system permits audio input, but only on a page-by-page basis. IO brush explores how we can use objects in the real world as a pallet from which to paint with [9]. However we focus more on compositions and an augmented form of college, than augmenting the paint itself.

THE NEVERENDINGSTORYTELLINGMACHINE SYSTEM

TheNeverendingStorytellingMachine is a system to create and edit a visual and audio narrative in book form at a distance, making use of any type of media, digital or analog, over a paper sketchbook interface. Users can embed voices and soundtracks to their stories, draw over spoken narrative or collage digital and analog content. *TheNeverendingStorytellingMachine* is also a display and communication platform.

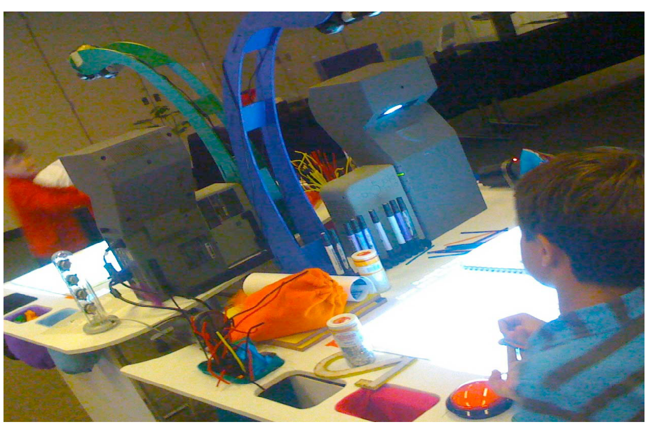


Figure 2. Two children experiment on collocated "creation-stations" by sending images back and forth via overhead webcams and projectors.

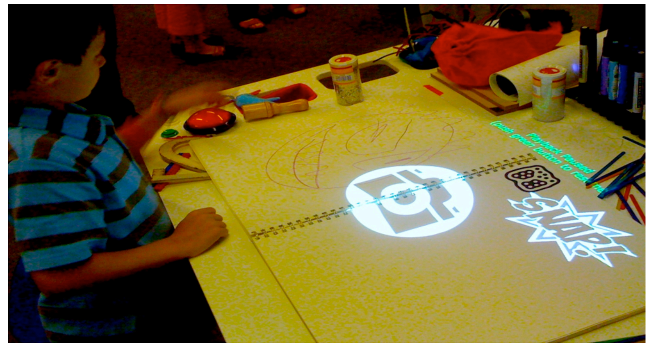


Figure 3. Child presses the red "capture" button to save and send a snapshot of his drawing to other creation-stations, corresponding to the correct page with the same Fiducial marker.

The system consists of two networked "creation-stations" that could be connected together through the Internet at any location. These creation-stations have a camera and projector system to capture and display content at scale. A large red button allows for capturing images of physical objects, or drawings, into the current page of a sketchbook. These pages are synchronized between the stations, i.e. what is captured in one book on page one is displayed in page one of the synchronized book. This allows users to build compositions together at a distance. Flipping pages will change the projected content to match the current page. The interface also displays the current page of the other user.

Since the system uses a physical paper sketchbook as the interface, the graphical and written content could be created anywhere, and captured later, making it a familiar, accessible tool to anyone anywhere. Once the sketchbook is brought into a "creation-station", it is transformed into shareable, enhanced media.

TECHNOLOGY

TheNeverendingStorytellingMachine works through the integration of a projector-camera system, a paper sketchbook with Fiducial markers, custom made software to capture and layer content and four buttons attached to an Arduino board: capture, send, record and pause. The custom made software is written in Touch, a visual programming environment [1, 2]. The Arduino was configured to listen for physical button presses and communicate their activity via serial protocol to Touch. Live video input from the webcam is run through a homography process, extracting corner-pin coordinates and re-projecting them to fit the sketchbook. A secondary camera uses the ReacTIVision [15] system to detect Fiducial Markers placed on each book page spread. The final resulting video projection sent from Touch is a composite image formed by the pre-made, background content (unique per book page spread) as well as the latest camera snapshots from both (or all participating) creation-stations.

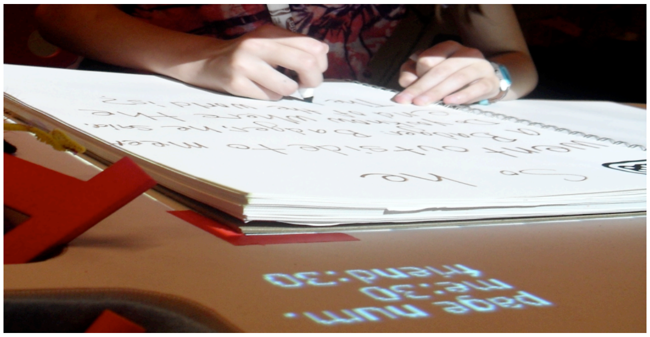


Figure 4. The creation station displays the page number of the current user and any other networked participants.

The creation-stations are networked to each other and access each other's file systems to read images and sounds. Additionally, metadata is sent over the network to inform each participating creation-station of each other's current page. Finally, video can be optionally broadcast over the network to enable a viewing station to watch the content evolving organically. All of the applications are built in a scalable, modular structure to support the expansion of the project based on site-specific needs.

DISCUSSION

Immersive Storytelling

Engaged by both physical and digital dimensions, users are no longer constrained by traditional story writing means, but instead, are encouraged to explore stories as multi-sensorial experiences.

A seamless hybridization of analog and digital modalities including visual, audio, and tactile elements, make *TheNeverendingStorytellingMachine* a powerful tool for communication. Allowing the user to bring their own personal objects, voices and ideas to the storytelling stage, *TheNeverendingStorytellingMachine* blends the edges of the conventional book with the outside world, offering limitless material for potential inspiration to produce meaningful narratives. Multiple entry points and multimedia resources invite users to immerse themselves in the creative process regardless of age or creative expertise.

Cooperative Authorship

TheNeverendingStorytellingMachine investigates new modes for content creation that folds author, viewer, and editor into one role shared by multiple users. Stories are treated as evolutionary collages with open participation. With the possibility of saving editions of the story in real-time, *TheNeverendingStorytellingMachine* could offer a type of hybrid library, where stories are archived in both digital and analog form and where authorship is seen as a social collective [5]. Documenting the evolution of content over time could also act as evidence to reveal narrative styles and strategies or to research co-creative processes.

Distance Collaboration

As a portable, self-contained system, creation-stations can be installed locally or installed over distances. This means that users could be in two different places, but synchronized to work on the same content. *TheNeverendingStorytellingMachine* could thus be an important informal and formal learning instrument as users in one location create stories together with users of different countries, cultures, or backgrounds. *TheNeverendingStorytellingMachine* in classrooms could help teachers expose students to stories from around the world, while teaching formal literacy skills and building empathy, drawing upon constructivism and collaborative pedagogies [3, 4]. Installed in museums and public spaces, the system could exhibit various stories from the public, gather a single, collective memory across time and space, or be used in performance art. In the home, family members could play and tell stories with loved ones far away or work autonomously on personal compositions.

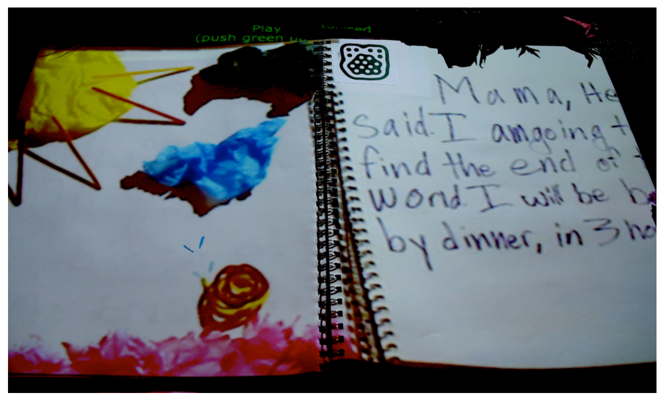


Figure 5. Mixed media content for one story co-created over time by multiple users in different locations.

FUTURE WORK

Based on observation, users engaged in their work using both hands, especially when holding down and object or stamping body parts, making it difficult to press the “send button”, usually resorting to their elbow. We plan to add a pedal button, so that the user can make full use of both hands and instead use the foot if chosen so.

In order to take full advantage of the accessibility of using a paper sketchbook interface, we plan to implement d-touch [14] markers. This have the advantage of being designable, convey meaning and being easily produced by the user without the need of a computer, printing and adding. By following a set of simple rules, the user can use any medium that provides high contrast to create them. Thus the user could draw personalized markers that seamlessly integrate to their creation.

REFERENCES

1. Arduino. www.arduino.cc
2. Touch. www.derivative.ca
3. Dewey, John. Experience and Education. New York, Touchstone, 1983.
4. Papert, S. *Mindstorms*. New York: Basic Books. 1980
5. Jenkins, Henry. (2006). Confronting the Challenges of Participatory Culture: Media Education for the 21st Century. Cambridge, MA: John D. And Catherine MacArthor Foundation. Retrieved from www.macfound.com
6. Hourcade, J. P., Bederson, B. B., Druin, A., and Taxé G. 2002. KidPad: collaborative storytelling for children. *Proc CHI 2002.*
7. Ishii, H. and Kobayashi, M. 1992. ClearBoard: a seamless medium for shared drawing and conversation with eye contact. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (1992). P. Bauersfeld, J. Bennett, and G. Lynch, Eds. CHI '92. ACM, New York, NY, 525-532.
8. Raffle, H., Vaucelle, C., Wang, R., and Ishii, H. 2007. Jabberstamp: embedding sound and voice in traditional drawings. ACM *SIGGRAPH 2007*.
9. Ryokai, K., Marti, S., and Ishii, H. 2004. I/O brush: drawing with everyday objects as ink. *Proc. SIGCHI Conference on Human Factors in Computing Systems* 2004.
10. Stifelman, L. J. Augmenting real-world objects: a paper-based audio notebook. *Proc.* CHI 1996.
11. Tang, J. C. and Minneman, S. L. 1990. VideoDraw: a video interface for collaborative drawing. *Proc. SIGCHI Conference on Human Factors in Computing Systems: Empowering People* 1990.
12. Wellner, P. and S. Freeman, The DoubleDigitalDesk:Shared Editing of Paper Documents. Technical Report EPC-93-108, EuroPARC 1993.
13. Yarosh, S., Cuzzort, S., Mueller, H., and Abowd, G.D. Developing a Media Space for Remote Synchronous Parent-Child Interaction. *Proc. IDC*, ACM 2009.
14. d-touch. www.d-touch.org
15. ReacTIVision. www.reactivision.sourceforge.net

Interactive Paper: Past, Present and Future

Beat Signer
Vrije Universiteit Brussel
1050 Brussels, Belgium
bsigner@vub.ac.be

Moira C. Norrie
ETH Zurich
8092 Zurich, Switzerland
norrie@inf.ethz.ch

ABSTRACT
Over the last few years, there has been a significant increase in the number of researchers dealing with the integration of paper and digital information or services. While recent technological developments enable new forms of paper-digital integration and interaction, some of the original research on interactive paper dates back almost twenty years. We give a brief overview of the most relevant past and current interactive paper developments. Then, based on our experience in developing a wide variety of interactive paper solutions over the last decade, as well as the results of other research groups, we outline future directions and challenges for the realisation of innovative interactive paper solutions. Further, we propose the definition of common data formats and interactive paper design patterns to ensure future cross-application and framework interoperability.

Author Keywords
Interactive Paper, Augmented Paper, Paper-Digital Interfaces

ACM Classification Keywords
H.5.2 Information interfaces and presentation (e.g., HCI): Miscellaneous.

General Terms
Design, Documentation

INTRODUCTION

In the early 1990s, the visionary Mark Weiser described a scenario of how intelligent paper might be integrated into future working environments in his seminal paper entitled 'The Computer for the 21st Century' [31]. Weiser coined the term *ubiquitous computing* and claimed that the most profound technologies are those that disappear by weaving themselves into the fabric of everyday life, as manifested in today's paper computing solutions. There are basically two main approaches to how paper can be integrated with digital information. While the *electronic paper* approach aims to make existing devices as paper-like as possible, the *augmented* or *interactive paper* approach focuses on augmenting regular paper by linking it to supplemental digital information and services.

In their book 'The Myth of the Paperless Office' [20], Sellen and Harper outline a number of paper affordances, including free-form annotations and reading across multiple documents, that, even with most recent electronic paper solutions, are difficult to emulate in digital media. Further evidence for the retention of paper as a key information medium is given by the fact that many forms of paper-based collaboration and interaction are hardly supported in digital environments [13]. Based on several field studies, Heath and Luff came to the conclusion that paper and screen-based interaction provide rather distinctive support for cooperation and that the use of paper not only persists due its intrinsic properties but also because of its mobile interactional flexibility [9].

Paper documents support various forms of content markup or annotation and, as outlined by Marshall [16], it is not easy to provide the same richness and flexibility to knowledge workers dealing with digital systems only. Studies with the most recent generation of e-book readers based on electronic paper, such as Amazon's Kindle[1], have shown that users are asking for better bookmarking and free-form note-taking support on these types of digital devices [3]. Even with the ongoing research on enhanced placeholders in digital documents [2], it seems to be hard to achieve the same flexibility and simplicity as offered by paper documents for particular tasks.

To prevent the loss of paper affordances that results from replacing paper with digital artefacts, a second research area focuses on the augmentation of regular paper with digital information and services. The first system closing the gap between paper and digital information spaces was Wellner's DigitalDesk [32]. By working on a special desk equipped with a camera-based finger and document tracking system and projector-based tabletop projection, interactions with paper documents can be tracked and linked to the appropriate digital services. Note that, with this approach, one of the most important affordances of paper—mobility—is lost. Over the years, interactive paper solutions for various domains, including work with engineering drawings and video storyboards, have been realised based on the DigitalDesk and similar paper document tracking systems [15].

UbiComp '10, Sep 26-Sep 29, 2010, Copenhagen, Denmark.
ACM 978-1-60558-843-8/10/09.

[1] http://kindle.amazon.com

In the Listen Reader project [1], a paper book was augmented with a multi-layered interactive soundtrack consisting of music and sound effects for a given story based on radio frequency identification (RFID) technology. Due to the fact that this form of digital augmentation enhances the reading process, we talk about an *enhanced reading* solution, whereas *enhanced writing* addresses the capture and processing of handwritten information. An example of an enhanced writing application is the Audio Notebook [26] which synchronises and links handwritten notes to pieces of recorded structured speech. Individual voice recordings can later be retrieved by simply pointing to specific parts of the handwritten notes. The idea of capturing handwritten notes and synchronising them with voice recordings was recently commercialised in the form of Livescribe's Pulse Smartpen[2] based on Anoto's digital pen and paper technology[3].

Anoto's digital pen and paper technology has led to increased interest in research on paper-digital interfaces. This was mainly due to the fact that, in contrast to earlier augmented desk or similar tracking solutions, it became easier to deal with the necessary hardware. Furthermore, the Anoto technology offers high resolution pen tracking and works with regular paper that simply has to be augmented with a special unintrusive positional dot pattern. In comparison to the DigitalDesk and related technologies, Anoto's solution enables mobile interaction with paper documents. Many of the recent interactive paper applications presented in the next section are based on the digital pen and paper technology. However, there exist other object and document identification solutions, such as linear barcodes, 2D barcodes, RFID tags and NFC tags, that can also be used to integrate paper documents with digital information spaces. One advantage of visual identifiers is the fact that most mobile phones now have a camera that can be used to read these identifiers and output any related digital information.

CURRENT INTERACTIVE PAPER SOLUTIONS

As mentioned above, many recent interactive paper solutions are based on Anoto's digital pen and paper technology. A camera that is integrated in the digital pen reads the unique printed dot pattern on a paper document and thus can detect the pen's position within a given document. The digital pen and paper solution was introduced to capture handwritten information in order to optimise certain business processes. For example, information written on a paper form can be captured, digitised via intelligent character recognition (ICR) and automatically stored in a database. The first generation of digital pens were designed for *batch processing* and worked in *offline mode* without any real-time interaction. The captured information was only transferred to a computer when the pen was docked to a computer.

Several interactive paper solutions have been realised based on digital pens working in offline mode. ButterflyNet [33] is a mobile notebook application for the capture and retrieval of information based on digital pen and paper technology. Field biologists can capture their handwritten notes and link them to other digital or physical media that they have collected in the field. An interactive paper-based notebook solution for biologists in labs has been investigated in Prism [27]. The Paper Augmented Digital Documents (PADD) [7] document workflow infrastructure supports the basic integration of paper documents with their digital counterparts by allowing a document that has been printed with the supplementary Anoto dot pattern to be annotated with a digital pen and the pen strokes are automatically shown as an overlay in the original digital document. However, the integration is simply based on a positional digital ink mapping rather than a semantic integration which means that, if a digital document is edited after printing, the mappings will no longer be correct for the new digital document version. To address this issue, the idea was taken further in PaperProof [29] where the mapping is no longer simply positional but based on the underlying digital document model. Furthermore, intelligent character recognition is used in combination with gesture recognition [22] to process pen strokes and transform them into the corresponding operations on the digital document.

While the Anoto pens originally supported no real-time interaction, more recent pens such as the magicomm G303 can be used in *streaming* or *online mode*. Interactive paper applications working in streaming mode include EdFest [17], a guide for the Edinburgh Fringe Festival that combines pen and paper with voice interaction and digital festival services, and PaperPoint [23], a presentation aid for controlling PowerPoint presentations via interactive paper slide handouts. As part of the NiCE Discussion Room project [8], digital pen and paper tools have been integrated into a meeting support system to collaboratively manage information. Note that digital pens can also be used on LCD screens which enables a seamless transition between paper and screen-based interactions [11].

The interactive paper applications presented above are based on different interactive paper frameworks and toolkits including PADD [7], the PaperToolkit [34], Letras [10] and iPaper [18, 21]. Our iPaper solution was developed to support different applications in the European PaperWorks[4] and Paper++[5] projects. The iPaper framework enables the rapid prototyping and development of applications based on an information-centric approach with a clear separation of concerns between the application logic and the interaction design. We also developed a number of powerful authoring tools (iPublish) [28] which were used to automatically generate the EdFest [17] guide from database content in terms of the PDF document to be printed and the paper-digital link definitions. They are also used to semi-automatically generate interactive PaperPoint [23] handouts.

A problem of digital pen and paper is the limited support for feedback when an application is not used in combination with a screen. To overcome this, non-visual output channels such as sound could be used. Another possibility could be the use of mobile and spatially aware projection of information as in MouseLight [24]. Recent projects based on other

[2] http://www.livescribe.com

[3] http://www.anoto.com/digital-pen-paper.aspx

[4] http://www.paper-works.org

[5] http://www.paperplusplus.net

technologies include PACER [14] which is a gesture-based interactive paper system that supports the manipulation of digital documents via the touch screen of a camera-equipped mobile phone. A similar solution for camera-based interaction with paper documents was realised in HotPaper [5]. Both systems apply a content-based recognition approach to identify document patches via their textual features without any need of visual markers.

FUTURE DIRECTIONS AND CHALLENGES

As we have highlighted in the previous section, there are a variety of interactive paper applications covering different domains. In addition to different hardware solutions, there exist a number of software frameworks for the digital pen and paper technology. In the remaining part of this paper, we would like to outline some future technical as well as non-technical challenges to stimulate a discussion between interactive paper application and framework developers.

Device independence. The interaction with the application logic of an interactive paper solution should be decoupled from any device-specific details. This enables an easy migration of applications in the case that new devices become available by just implementing the necessary device drivers. Furthermore, specific device drivers could be shared across interactive paper platforms.

Digital ink abstraction. While there exist standards for digital ink representation such as InkML, none of the existing interactive paper frameworks makes extensive use of these standards. Open and *standardised data formats* might help to not only exchange information across frameworks but also enable the integration of pen and paper data with other types of media to realise generic mixed-reality environments [4].

Application deployment. Currently, most interactive paper applications are isolated solutions without any potential cross-application interaction. In general, a user has to ensure that they have installed the right application before they start to interact with a document. In the future, it might be worth investigating a service-oriented architecture where interactive paper applications can be automatically downloaded and installed on demand based on specific pen and paper interactions. Individual interactive document identifiers might be bound to the corresponding services via a global *Paper Lookup Service (PLS)* [6] that represents some kind of yellow pages for interactive paper solutions, in a similar way to how a domain name service (DNS) is used on the Internet. The *modularisation* of components within different interactive paper platforms might further facilitate the cross-application composition of services and enhance the reusability of interactive paper functionality.

Visual encoding. Since the design of interactive paper interfaces is a relatively new domain, there are no established guidelines on how to design an interactive paper interface. When implementing specific applications, one realises that some visual encodings work better than others. To share this knowledge, it might be beneficial to come up with some common design guidelines. The reuse of *visual design patterns* across different applications could further improve the performance of individual interactive paper users.

Interaction design. Similar to the lack of visual guidelines, there are no rules on how to design the interaction with an application and it might be worth investigating digital pen and paper interaction strategies [25]. There are many differences compared to traditional digital user interfaces. For example, there are the previously mentioned limitations for visual feedback and there is a lack of a transactional operation concept as manifested in GUIs in the form of modal dialogues. The limited possibilities for feedback might also be overcome by printed electronics [12] and the *fusion of electronic paper with interactive paper*. Note that the underlying hardware platform can restrict the possible interactions. For example, with Anoto's solution one has to decide at printing time whether a specific part of a document is going to be used in online or offline mode.

Authoring and publishing. Most existing interactive paper applications are authored via a manual authoring tool or even through hard-coded interactions. More advanced solutions apply a content-driven cross-media media publishing approach [17] or the automatic transformation of existing documents into interactive paper versions [19] based on mixed document models [30]. Also the scalability and distribution of interactive paper documents in combination with the versioning of documents remains an open problem. The phenomenon that we witnessed with Web 2.0 applications seems to be applicable in interactive paper environments with users composing their own applications based on a future *Interactive Paper 2.0* infrastructure.

CONCLUSIONS

The development of interactive paper solutions has become a very active research area. While different interactive paper frameworks support the application development, the question is whether these frameworks are missing a common abstraction layer. It might be the right time to reflect and share some wisdom. The definition of common data formats and design guidelines could be a first step towards real cross-application and cross-framework interoperability.

ACKNOWLEDGEMENTS

We would like to thank all of our colleagues involved in our interactive paper projects over the last decade.

REFERENCES

1. M. Back, J. Cohen, R. Gold, S. Harrison, and S. Minneman. Listen Reader: An Electronically Augmented Paper-Based Book. In *Proc. CHI 2001*, pages 23–29, 2001.
2. G. Buchanan and J. Pearson. Improving Placeholders in Digital Documents. In *Proc. ECDL 2008*, pages 1–12, 2008.
3. D. T. Clark, S. P. Goodwin, T. Samuelson, and C. Coker. A Qualitative Assessment of the Kindle E-book Reader: Results from Initial Focus Groups. *Performance Measurement and Metrics*, 9(2), 2008.

4. E. Costanza, A. Kunz, and M. Fjeld. Mixed Reality: A Survey. *LNCS*, 5440:47–68, 2009.

5. B. Erol, E. Antúnez, and J. J. Hull. HOTPAPER: Multimedia Interaction with Paper Using Mobile Phones. In *Proc. ACM Multimedia 2008*, pages 399–408, 2008.

6. M. Frisoni. A Distributed Lookup Service for Interactive Paper Applications. Master's thesis, ETH Zurich, 2008.

7. F. Guimbretière. Paper Augmented Digital Documents. In *Proc. UIST 2003*, pages 51–60, 2003.

8. M. Haller, J. Leitner, T. Seifried, J. R. Wallace, S. D. Scott, C. Richter, P. Brandl, A. Gokcezade, and S. Hunter. The NiCE Discussion Room: Integrating Paper and Digital Media to Support Co-Located Group Meetings. In *Proc. CHI 2010*, pages 609–618, 2010.

9. C. Heath and P. Luff. *Technology in Action*. Cambridge University Press, 2000.

10. F. Heinrichs, J. Steimle, D. Schreiber, and M. Mühlhäuser. Letras: An Architecture and Framework for Ubiquitous Pen-and-Paper Interaction. In *Proc. EICS 2010*, pages 193–198, 2010.

11. R. Hofer and A. Kunz. Digisketch: Taming Anoto Technology on LCDs. In *Proc. EICS 2010*, pages 103–108, 2010.

12. V. Kantola, J. Kulovesi, L. Lahti, R. Lin, M. Zavodchikova, and E. Coatanéa. Printed Electronics: Now and Future. *Bit Bang - Rays to the Future*, pages 63–102, 2009.

13. D. M. Levy. *Scrolling Forward: Making Sense of Documents in the Digital Age*. Arcade Publishing, 2001.

14. C. Liao, Q. Liu, B. Liew, and L. Wilcox. PACER: Fine-grained Interactive Paper via Camera-touch Hybrid Gestures on a Cell Phone. In *Proc. CHI 2010*, pages 2441–2450, 2010.

15. W. E. Mackay and A.-L. Fayard. Designing Interactive Paper: Lessons from three Augmented Reality Projects. In *Proc. IWAR '98*, pages 81–90, 1998.

16. C. C. Marshall. The Future of Annotation in a Digital (Paper) World. In *Proc. GSLIS Clinic: Successes and Failures of Digital Libraries*, 1998.

17. M. C. Norrie, B. Signer, M. Grossniklaus, R. Belotti, C. Decurtins, and N. Weibel. Context-Aware Platform for Mobile Data Management. *ACM/Baltzer Wireless Networks (WINET)*, 13(6):855–870, 2007.

18. M. C. Norrie, B. Signer, and N. Weibel. General Framework for the Rapid Development of Interactive Paper Applications. In *Proc. CoPADD 2006*, pages 9–12, 2006.

19. M. C. Norrie, B. Signer, and N. Weibel. Print-n-Link: Weaving the Paper Web. In *Proc. DocEng 2006*, pages 34–43, 2006.

20. A. J. Sellen and R. Harper. *The Myth of the Paperless Office*. MIT Press, 2001.

21. B. Signer. *Fundamental Concepts for Interactive Paper and Cross-Media Information Spaces*. PhD thesis, ETH Zurich, May 2006. Dissertation ETH No. 16218.

22. B. Signer, U. Kurmann, and M. C. Norrie. iGesture: A General Gesture Recognition Framework. In *Proc. ICDAR 2007*, pages 954–958, 2007.

23. B. Signer and M. C. Norrie. PaperPoint: A Paper-Based Presentation and Interactive Paper Prototyping Tool. In *Proc. TEI 2007*, pages 57–64, 2007.

24. H. Song, F. Guimbretière, T. Grossman, and G. Fitzmaurice. MouseLight: Bimanual Interactions on Digital Paper Using a Pen and a Spatially-Aware Mobile Projector. In *Proc. CHI 2010*, pages 2451–2460, 2010.

25. J. Steimle. Designing Pen-and-Paper User Interfaces for Interaction with Documents. In *Proc. TEI 2009*, pages 197–204, 2009.

26. L. J. Stifelman, B. Arons, and C. Schmandt. The Audio Notebook: Paper and Pen Interaction with Structured Speech. In *Proc. CHI 2001*, pages 182–189, 2001.

27. A. Tabard, W. Mackay, and E. Eastmond. From Individual to Collaborative: The Evolution of Prism, a Hybrid Laboratory Notebook. In *Proc. ECSCW 2008*, pages 569–578, 2008.

28. N. Weibel. *A Publishing Infrastructure for Interactive Paper Documents: Supporting Interactions across the Paper-Digital Divide*. PhD thesis, ETH Zurich, 2009. Dissertation ETH No. 18514.

29. N. Weibel, A. Ispas, B. Signer, and M. C. Norrie. PaperProof: A Paper-Digital Proof-Editing System. In *Proc. CHI 2008*, pages 2349–2354, 2008.

30. N. Weibel, M. C. Norrie, and B. Signer. A Model for Mapping between Printed and Digital Document Instances. In *Proc. DocEng 2007*, pages 19–28, 2007.

31. M. Weiser. The Computer for the 21st Century. *Scientific American*, 265(3), 1991.

32. P. Wellner. Interacting with Paper on the DigitalDesk. *Communications of the ACM*, 36(7):87–96, 1993.

33. R. B. Yeh, C. Liao, S. R. Klemmer, F. Guimbretière, B. Lee, B. Kakaradov, J. Stamberger, and A. Paepcke. ButterflyNet: A Mobile Capture and Access System for Field Biology Research. In *Proc. CHI 2006*, pages 571–580, 2006.

34. R. B. Yeh, A. Paepcke, and S. R. Klemmer. Iterative Design and Evaluation of an Event Architecture for Pen-and-Paper Interfaces. In *Proc UIST 2008*, pages 111–120, 2008.