Paper-Based Augmented Reality

Conference Paper · November 2007

DOI: 10.1109/ICAT.2007.49 · Source: DBLP

7 authors, including:

Jonathan J. Hull
Ricoh Innovations, Inc.
175 PUBLICATIONS **6,395** CITATIONS

SEE PROFILE

Paper-Based Augmented Reality

Jonathan J. Hull, Berna Erol, Jamey Graham, Qifa Ke, Hidenobu Kishi, Jorge Moraleda, Daniel G. Van Olst
Ricoh Innovations, Inc., California Research Center, Menlo Park, CA, USA
hull@rii.ricoh.com

Abstract

A new method for augmenting paper documents with electronic information is described that does not modify the format of the paper document in any way. Applicable to both commercially printed documents as well as documents that are output from PC's, the technique we call *Paper-Based Augmented Reality* substantially improves the utility of paper. We describe the recognition technology that makes this possible as well as several applications. An implementation on a camera phone is discussed that lets users retrieve data and access links from paper documents to electronic data. Recognition is performed at 4 frames per second on a Treo 700w and support is provided for several user applications, including "clickable paper" – printed web pages whose appearance is unchanged but that can be navigated with a camera phone.

Key words: camera phone, paper documents, data links

1. Introduction

Linking the physical and digital worlds is a long standing goal of augmented reality. Image recognition is often used to derive link information from explicit or implicit markers in a scene and to trigger processes, such as the retrieval of information from a web site [1-4]. Previous augmented reality techniques for paper documents alter their appearance with bar codes or textured paper. Often, a special purpose device, such as a pen with a camera in it, must be employed to recognize the embedded code. These characteristics all inhibit the use of augmented reality with paper documents.

Our prior work indicated that images of small patches of text contain enough information to make them as unique as a fingerprint [6]. We showed that it was possible to distinguish a small rectangular region (one inch square) from among thousands of other text image patches. At that time, we leveraged this characteristic to identify the electronic original for a given paper document. However, the same result indicates that patches of text can be used as markers in an augmented reality system in which arbitrary x-y positions in printed text passages can be associated with electronic data.

This paper proposes a new method of interacting with documents termed *Paper-Based Augmented Reality* that links patches of text to electronic data and uses a camera phone as the recognition device. This brings a new level of interactivity to paper documents and allows them to be updated without reprinting them. New links to electronic data can be easily created and the content of old links can be changed without modifying the original document.

2. Algorithm Outline

The operation of a paper-based augmented reality system is illustrated in Fig. 1. PBAR-enabled documents are created by scanning a document and indexing it for text patch recognition as shown in Fig. 1 (a). Data is associated with regions on the document by choosing "hot spots" that are rectangular patches of text, and adding data to the hot spots (Fig. 1 (b)). The index information and symbolic hot spot data are stored in the PBAR database. A simple example of data in a hotspot is a URL that points to a web page. However, it could just as easily be a video file, an audio clip or even an electronic version of the original document itself.

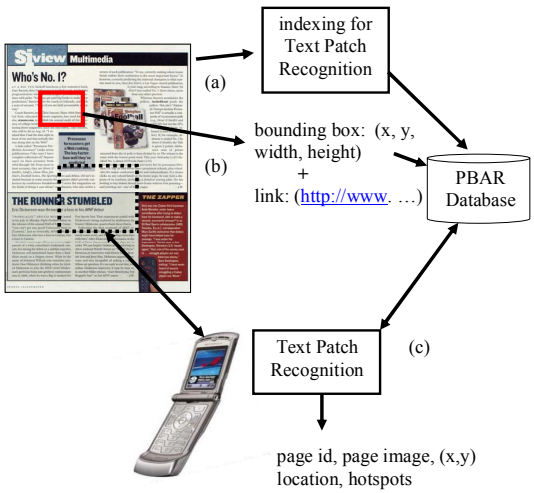


Fig. 1. Creating and using paper-based augmented reality documents: hotspots and data associated with them are designated (a) and the page image is indexed (b). Subsequently, the text patch recognition algorithm identifies the page, its image, the (x,y) location at which the camera is pointing, and nearby hotspots.

0-7695-3056-7/07 $25.00 © 2007 IEEE
DOI 10.1109/ICAT.2007.49

At a subsequent time, Fig. 1 (c), a user captures an image of a portion of the page with a camera phone; the system applies the same text patch recognition algorithm used during the indexing phase and determines whether the database contains the page image. An identification for the page is returned as well as the (x,y) location and extent of the input image and the location of hotspots on that page and other nearby pages in the document. The hotspot data can be returned to the phone and the appropriate rendering application applied to it. If the data is a URL, a web browser could be invoked.

3. Text Patch Recognition

The objective of the text patch recognition algorithm is to correctly determine the identity of a page and the x-y position in the page of a small patch of text. The technical challenge is illustrated by the image in Fig. 2 that shows the typical quality of images produced by commonly available camera phones. Characters are so blurry that "OCR" is basically impossible. However, it is still possible in almost every case to identify the bounding boxes around words since the spaces between words and lines can still be distinguished.

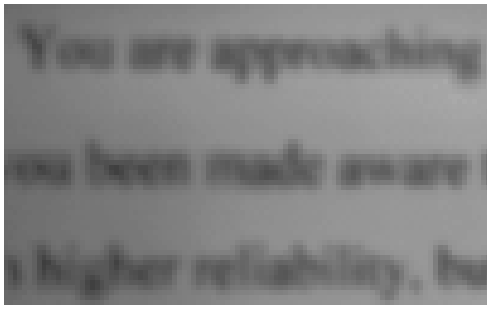


Fig. 2. Typical camera phone image.

A previously developed technique for page image matching has demonstrated high accuracy [5]. It used the number of characters in N horizontally adjacent words (called n-grams) as a feature or ***descriptor*** and identified the input image as the document that contained the highest number of descriptors. Experience with this method demonstrated its reliability. Typically, values of N equal to 4, 5 or 6 provided increasing accuracy when applied to synthetic data but decreasing accuracy in the presence of noise since the percentage of incorrect n-grams increases as the percentage of words with inaccurate word lengths increases.

Recently, we observed that the accuracy of the previous method for patch recognition can be substantially improved by using the vertical arrangement of words in addition to their horizontal placement. The algorithm shown in Fig. 3 takes advantage of this characteristic and independently classifies a patch of text using horizontal and vertical features. The separate rankings are combined using information about how the detected features co-occur in the original database. This approach increases the information extracted from the image and as a result uniquely identifies a patch of text in a large collection of document images [7].

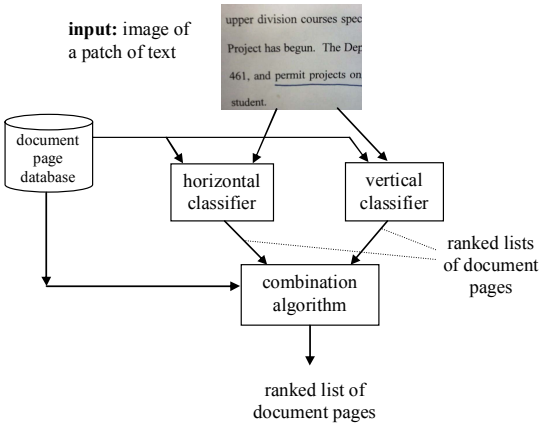


Fig. 3. Combination of horizontal and vertical classifiers for text patch recognition.

Fig. 4 shows an example of how vertical layout is integrated with horizontal n-grams. Vertical trigrams are exhaustively generated (c) from all the words above and below a given word, where *above* and *below* are based on overlap of bounding boxes. Lists of documents that contain both the horizontal (d) and vertical (e) trigrams are looked up in databases that are organized to support such queries. Two lists of votes (f) and (g) are provided.

This example illustrates the typical behavior that is observed in practice: documents that receive votes from horizontal trigrams in general are different from documents that receive votes from vertical trigrams. That is, the two sources of information appear to be independent. This is very important since it indicates that in practice we will be able to build a two-classifier system with significantly higher accuracy than either of the single classifiers.

The algorithm that combines the lists of votes from horizontal and vertical n-gram matching uses information about the physical location of n-grams on the original printed pages. For every document in common among the top M choices output by each classifier, the location of every horizontal n-gram that voted for that document is compared to the location of every vertical n-gram that voted for the document. A document receives a number of votes equal to the number of horizontal n-grams that overlap any vertical n-gram, where "overlap" occurs when the bounding boxes of two n-grams overlap.

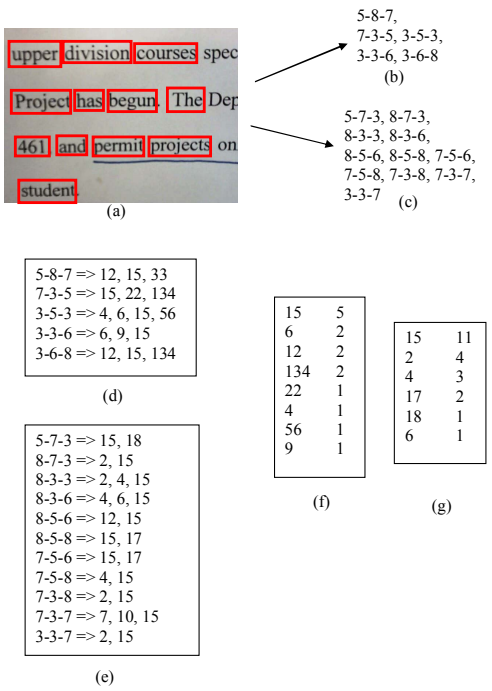


Fig. 4. Example of algorithm that uses horizontal ***and*** vertical word length n-grams. (a) input image, (b) horizontal and (c) vertical word length trigrams extracted from it. Lists of documents that contain each (d) horizontal and (e) vertical trigram, and number of votes for each document by (f) horizontal and (g) vertical trigrams.

The experimental performance of the combination of horizontal and vertical word length n-grams was tested with data extracted from 738 Words files, containing 5699 pages, downloaded from the Internet. Those files were converted to an XML representation that includes every character printed on every page.

Hash tables were constructed using the horizontal and vertical word-length trigrams. Each unique trigram points to a list of the document pages that contain it. The training data provided 1,347,197 horizontal and 1,434,792 vertical trigrams. Of these, 4883 of the horizontal trigrams and 10,250 of the vertical trigrams were unique. Note: this provides an indication of the information content in vertical trigrams since the same amount of data produced more than double unique vertical than horizontal trigrams.

Test data were extracted from the XML files that simulate the scanning of 2" high by 3" wide patches that begin 3" to the right and 4" down from the top left corner of the first six pages in each document. Trigrams were constructed from each patch that contained at least 20 words. Fig. 5 shows an example of the data generated by this procedure.

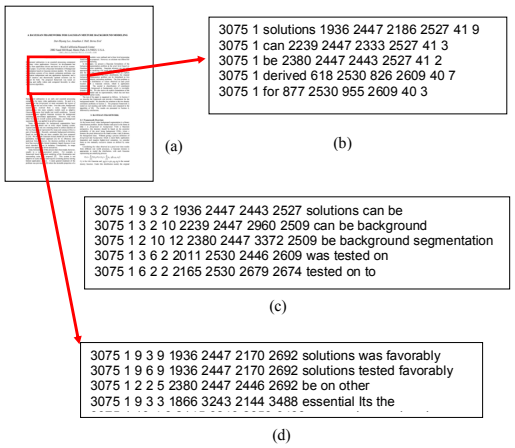


Fig 5. Example of test data generation. (a) image patch, (b) the horizontal word lengths, (c) horizontal trigrams and (d) vertical trigrams extracted from it.

Each word, such as "solutions" at the top of Fig. 5 (b) is characterized by the document number "3075" and page number "1" in which it occurred. The bounding box coordinates are given as well as number of characters in the word, 9 in this case.

Each trigram, such as "3075 1 9 3 2 1936 2447 2443 2527 solutions can be" at the top of Fig. 5 (c), includes an identification of the document "3075" and the page number within the document "1" where it occurs. This is followed by the trigram of word lengths: "9 3 2" for the words "solutions can be". The coordinates of the bounding box for the trigram are minx=1936, miny=2447, maxx=2443 and maxy=2527. A noise simulator was developed that randomly chose a given percentage of words and corrupted their length by randomly adding or subtracting 1 from its length.

An experiment was run in which the top 10 choices from the horizontal and vertical classifiers were combined with the overlap measure described earlier. The results are shown in Table 1. If there was no patch in common among the results provided by the horizontal and vertical classifiers, the system rejected that patch.

With no noise, the combined classifier was 100% correct. When 10% of the word lengths were corrupted, the system could make a decision for 1944 of the 1988 patches. The other 44 or 2% of the patches were rejected. In this case, horizontal trigrams correctly classified 93 of the 1944 patches, vertical trigrams 92% and the combination of the two was 99% correct. This trend continued with a 20% noise level where the system could recognize 1754 of the patches with a 99% correct rate for the combination and only 82% correct for the vertical trigrams. At 30% noise, vertical trigrams could only provide a 66% correct rate but the combination provided a 95% correct rate.

Noise ratio	Num. patch's	Num. decided	% reject	% corr. horiz	% corr. vert	% corr. comb.
0.00	1988	1974	1%	97%	97%	100%
0.10	1988	1944	2%	93%	92%	99%
0.20	1988	1754	12%	78%	82%	99%
0.30	1988	1214	39%	57%	66%	95%
0.40	1988	468	76%	29%	40%	81%
0.50	1988	150	91%	7%	10%	26%

Table 1. Combination of top 10 results from the horizontal and vertical classifiers.

The adaptation of the bounding box classifier to image data like that shown in Fig. 2 led to the development of descriptors that represent word length in units called "nubs" that are determined by dividing the width of a word in pixels by its height. Descriptors are calculated that combine horizontal and vertical information by using the angles between horizontally and vertically adjacent groups of bounding boxes. The hash table is organized by patches rather than complete pages. The output of the patch recognizer is a list of patches (including the document in which they occur and their x-y location) sorted by the number of descriptors they have in common with the input image.

The current implementation runs on a Treo 700w with a 312 MHz PXA272 processor at about 4 frames per second with a database of about 250 documents. We performed experiments that tested the accuracy of the real-time system (running on a PC) with databases of nearly 5,000 images of size 176x144 that were generated with a system [8] that simulates the video output by a Treo 700w as it's moved over a document. Results showed that a 55% correct rate can be achieved with images that contain about 8 lines of text. It's important to note that while a 55% correct rate may seem low, in practice it's more than adequate because a PBAR system runs in real time on a video stream as the user actively moves the camera over the document, essentially cooperating with the recognizer to improve its performance.

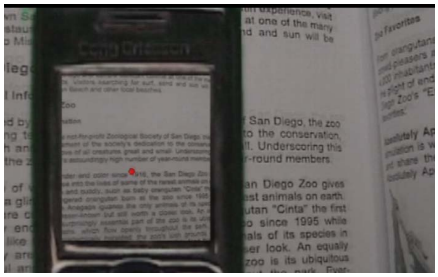
4. Applications

There are many possible applications for PBAR. Important considerations include whether the PBAR database is on the phone or on a server and whether the database is created as a side effect of printing a document on a PC. This section presents two examples, a travel guidebook and web page printouts.

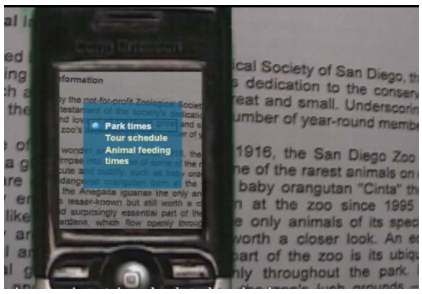
4.1 Travel Guidebook

Travel guidebooks are almost out-of-date the minute they are printed. The accuracy of the time-sensitive information they contain is always in question. A reader has no way of knowing whether the opening times of the attractions mentioned in such a publication are correct. The only reasonable solution is to call the facility in question. Instead, PBAR allows someone to point a camera phone at the text that describes the facility and retrieve the currently available information about it.

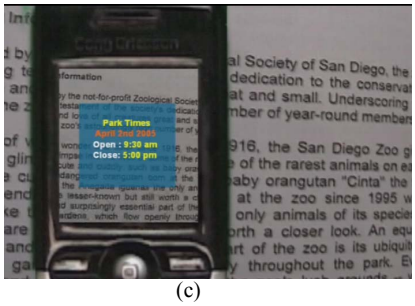
Fig. 6 shows an example of an augmented reality travel guidebook. The real-time recognition system described here makes it possible to overlay a marker (in this example a red dot) that indicates extra information is available related to the underlying passage of text. When the button on the phone is pressed, the client application retrieves the menu of choices (Fig. 6 (b)) related to the text passage. Based on the user's selection, the appropriate information is displayed, in this case the opening times for the San Diego Zoo.



(a)



(b)



(c)

Fig. 6. An augmented travel guidebook. (a) an indication that information is present, (b) a menu of choices, and (c) the selected information.

4.2 Self-Printed Documents

Documents that are printed on a desktop PC are typically created by individuals for their personal use. PBAR allows users to customize the interactivity of those documents based on their own needs and permits

the database to be under the user's personal control: it could be shared on a networked server, saved on the individual's PC, or pushed to the user's camera phone.

We created the architecture for printing and indexing web pages shown in Fig. 7 that automatically captures an image of every document in the print driver, indexes them in the PBAR database, and associates the URL's in the document with their physical location on the web page. This includes a plug-in in Internet Explorer that exports URL's to text patch feature extraction software. This system is fully implemented on Windows XP and Vista and can be installed on any PC.

The version of PBAR we call *Clickable Paper* is shown in Fig. 8. The paper printout (a) of a web page is imaged and (b) the region surrounding a URL is captured. That image as well as the web page corresponding to the URL (c) is shown in the user interface on the camera phone (d). A short history of the last three URL's accessed with this system is also displayed in the user interface.

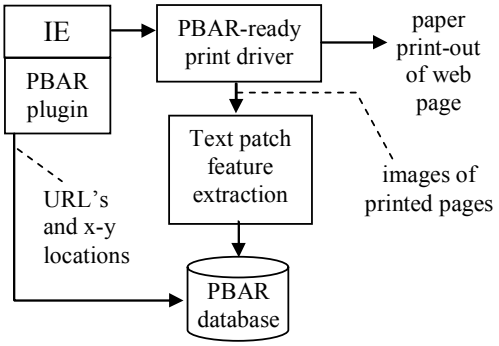


Fig. 7. Architecture for automatic creation of augmented web page print-outs.

5. Conclusions

A new paradigm for augmented reality was described in which electronic data is added to paper documents without changing the appearance of the paper document in any way. This approach leverages the essential discovery that a unique signature can be derived from an image of a small patch of text and that signature can be linked to electronic data. We described a new algorithm for text patch recognition and presented experimental results that demonstrated it can distinguish an image of a patch of text from a collection of thousands of examples. An implementation on a camera phone that runs at 4 frames per second was discussed. Two of the applications that we've created were presented: an augmented travel guidebook and Clickable Paper. Both of them show the potential value of paper-based augmented reality for common applications.

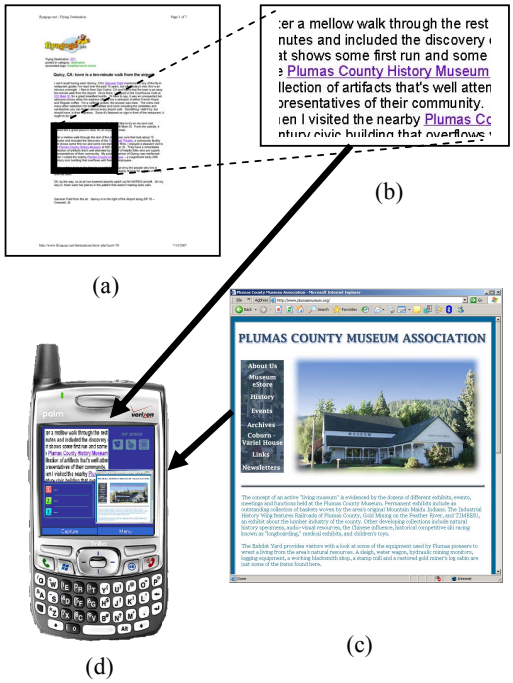


Fig. 8. Clickable Paper system. (a) a printed web page, (b) image of patch containing a URL, (c) web page, and (d) UI on Treo.

References

1. E. Bier, M. Stone and K. Pier, "Enhanced Illustration Using Magic Lens Filters," IEEE Computer Graphics and Applications, v. 17, no. 6, 62-70, Nov. 1997.
2. M. N. Billinghurst and A. Henrysson, "Research Directions in Handheld AR," Int. Journal of Virtual Reality, v. 5 no. 2, 51-58, 2006.
3. V. Ferrari, T. Tuytelaars, and L. Van Gool, "Markerless Augmented Reality with a Real-Time Affine Region Tracker," IEEE and ACM Intl. Sym. on Augmented Reality, v. I, 87-96, Oct. 2001.
4. A. Henrysson, M. Billinghurst and M. Ollila, "Virtual Object Manipulation Using a Mobile Phone," Int. Conf. on Augmented Telexistence, Christchurch, New Zealand, 164-171, Dec. 5-8, 2005.
5. J. J. Hull, "Document Image Matching and Retrieval with Multiple Distortion-Invariant Descriptors," in Document Analysis Systems, World Scientific, pp. 379-396, 1995.
6. J. J. Hull and J. F. Cullen, "Document Image Similarity and Equivalence Detection," IEEE Int. Conf. on Doc. Analysis and Recog., Ulm, Germany, 308-311, Aug. 18-20, 1997.
7. J.J. Hull, B. Erol, P. E. Hart, D. S. Lee and K. Piersol, "Method and System for Image Matching in a Mixed Media Environment," U.S. Patent Application 20060262352, Nov. 23, 2006.
8. A. Lookingbill, E. R. Anutnez, B. Erol, J. J. Hull, Q. Ke, and J. Moraleda, "Ground-truthed Video Generation from Symbolic Information," IEEE Int. Conf. on Multimedia and Expo, Beijing, 1411-1414, July 3-5, 2007.