

HOTPAPER: Multimedia Interaction with Paper using Mobile Phones

Berna Erol

RICOH Innovations
California Research Center
2882 Sand Hill Road,
Menlo Park, CA, USA
berna_erol@rii.rioh.com

Emilio Antúnez

Stanford University
Department of Electrical Eng.
Packard 243, 350 Serra Mall,
Stanford, CA, USA
eantunez@stanford.edu

Jonathan J. Hull

RICOH Innovations
California Research Center
2882 Sand Hill Road,
Menlo Park, CA, USA
hull@rii.rioh.com

ABSTRACT

The popularity of camera phones enables many exciting multimedia applications. In this paper, we present a novel technology and several applications that allow users to interact with paper documents, books, and magazines. This interaction is in the form of reading and writing electronic information, such as images, web urls, video, and audio, to the paper medium by pointing a camera phone at a patch of text on a document. Our application does not require any special markings, barcodes, or watermarks on the paper document. Instead, we propose a document recognition algorithm that automatically determines the location of a patch of text in a large collection of document images given a small document image. This is very challenging because the majority of phone cameras lack autofocus and macro capabilities and they produce low quality images and video. We developed a novel algorithm, Brick Wall Coding (BWC), that performs image-based document recognition using the mobile phone video frames. Given a document patch image, BWC utilizes the layout, i.e. relative locations, of word boxes in order to determine the original file, page, and the location on the page. BWC runs real-time (4 frames per second) on a Treo 700w smartphone with a 312 MHz processor and 64MB RAM. Using our method we can recognize blurry document patch frames that contain as little as 4-5 lines of text and a video resolution as low as 176x144. We performed experiments by indexing 4397 document pages and querying this database with 533 document patches. Besides describing the basic algorithm, this paper also describes several applications that are enabled by mobile phone-paper interaction, such as inserting electronic annotations to paper, using paper as a tangible interface to collect and communicate multimedia data, and collaborative homework.

Categories and Subject Descriptors

I.7 [Document and Text Processing]: General.

H.5.2 [Inf. Interfaces and Presentation]: User Interfaces

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

Conference'08, October 27–November 1, 2008, Vancouver, BC, Canada.
Copyright 2008 ACM 1-58113-000-0/00/0004...\$5.00.

Keywords

linking paper to electronic data, markerless linking, mobile imaging, mobile interaction

1. INTRODUCTION

Camera phones are powerful image capture and processing devices that almost everybody carries with them. Therefore, it is not surprising that recently many research efforts have used mobile image and video analysis for applications such as gesture recognition [1][2], location identification [3], and bridging the gap between electronic and physical worlds [4]-[14].

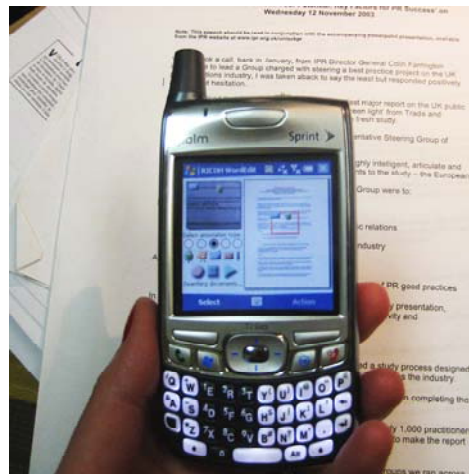


Figure 1. HotPaper document recognition and multimedia annotation application running on a Treo 700w mobile phone.

In this paper we propose a novel solution for multimedia interaction with paper documents using mobile phones that we call HotPaper. In the literature several methods have been suggested for using imaging devices to link paper documents to electronic data. Examples include watermarking or glyph-based techniques for linking electronic data to images [4]-[6] and barcode-based methods for video retrieval [7]-[9]. A significant issue for these applications is the need to modify the document's format by introducing a machine-readable code that improves the utility of the document but disrupts its appearance. Furthermore, the correctness of any marking must be verified

before the document is printed because it is not possible to modify the printed document afterwards.

HotPaper analyzes the contents of a captured document patch image or a video frame to identify the corresponding electronic document, page number, and location on the page. The documents look the same as they always have; there's nothing special about their layout or format and no special markings are applied. Content analysis and paper linking is done by employing a novel algorithm that we call Brick Wall Coding (BWC). BWC builds on our previous n-gram based algorithm described in [10]. BWC uses the local configuration of word bounding boxes and geometric constraints to identify the document patch uniquely among huge collections of documents. In this paper we demonstrate the effectiveness of the algorithm using a database of 4397 document pages. The current algorithm runs on a Treo mobile phone with a 312 MHz processor and can perform 4 frames per second recognition. Documents with as low as 4-5 lines of text can be recognized. Once a document region is digitally identified, the user then can insert and playback electronic annotations, such as audio and video clips, images, urls, handwritten and typed annotations. A prototype of our system is shown in Figure 1.

In this paper we also describe some unique applications that allow users to electronically annotate paper documents. This includes real-estate guide and meeting agenda applications that use tangible paper artifacts as a tool for collecting multimedia data, including images, audio, and typed annotations.

Next, we give an overview of prior art. Section 3 describes the Brick Wall Coding algorithm and how document patch retrieval works. The experimental setup and results are presented in Section 4. Section 5 describes several electronically writable paper applications. Conclusions and Outlook is presented in Section 6.

2. PRIOR ART

Connecting the paper and electronic worlds has been a longstanding goal and many research efforts have tried to achieve this. Research in this area falls in two categories: (1) methods that use special markings on paper and (2) methods that perform document linking via content analysis.

Unique dot patterns and glyphs have been used in combination with camera-pens to detect the location of user markings on paper and paper-digital information linking [4]-[6]. In most cases, dot patterns on paper are barely visible, so they do not distract the user from the content. However, special paper needs to be used for printing and mobile phone cameras are not capable of reading such high density markings. Barcode based systems have also been also suggested to link paper documents with digital media [7]-[9]. Adaptation of these techniques to mobile phones is more feasible since some mobile phones are already equipped with barcode or QR code readers. Disadvantages of using barcodes for such linking are that (1) users cannot link digital information to already printed materials, and (2) many barcodes are not attractive and can detract from the content.

Several researchers also focused on content-based recognition using mobile phones for linking digital data to physical world. Most mobile image recognition techniques support capturing a digital image, sending the image to a server, and performing

image recognition using SIFT features [11] or its variations [12][13].

Recently, researchers have investigated content-based recognition of documents using mobile phone images. In [14], Kise et. al uses projection invariant features to retrieve document images. Because the features they employ are not discriminative enough, they use entire page images for retrieval, instead of small document patches. Doermann et. al. [15] extends their work to use partial images, but the partial document images they use contain many text lines and the algorithm runs on images with 1024x1280 resolution. In contrast, our algorithm can recognize document text patches with as low as 4-5 lines of text. This way, users can more accurately specify the location of annotations. Also our algorithm runs on video frames of size 176x144 at 4 frames per second enabling real-time interaction with paper.

3. MOBILE DOCUMENT RETRIEVAL ALGORITHM

3.1 Technical Challenge

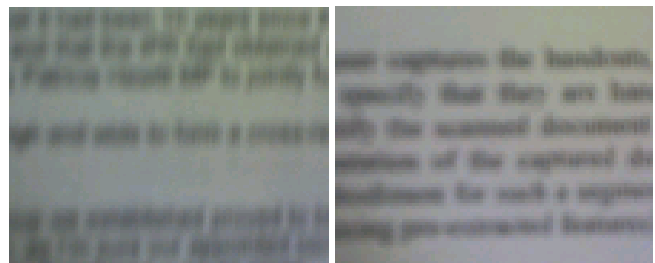


Figure 2. Video frames captured by a typical mobile phone without a macro lens.

The technical challenge of HotPaper is linking low quality document patch images/video frames captured by mobile phones to digital data. As can be seen in Figure 2, the video frames of document patches captured by a cell phone are blurry, out of focus, and nearly impossible to recognize using optical character recognition. However, we can still reliably extract word bounding boxes from these images. Our proposed image representation, Brick Wall Coding (BWC), makes use of the layout and the aspect ratios of word boundaries when retrieving document images.

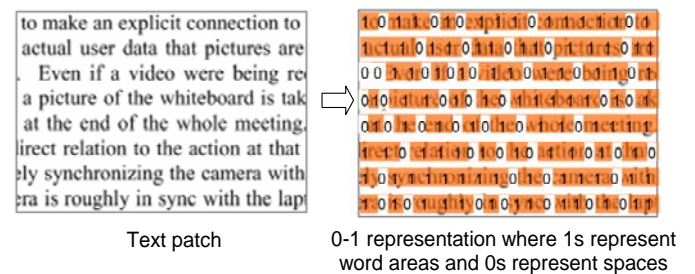


Figure 3. Word layout of a text patch encodes a unique signature with $2^{(18 \times 8)}$ unique values.

The horizontal and vertical alignment of word boxes in a document encodes a signature that is very much like a fingerprint. However, is this signature unique enough to identify a document from a small document patch? Consider the document image shown in Figure 3. If the word bounding boxes are computed, and word regions are represented with ones and spaces are represented with zeros, $2^{(18 \times 8)}$ (approximately 18 columns and 8 lines) unique signatures can be obtained. Therefore, in theory, the x-y location of a small patch of text can be uniquely identified in a huge collection of documents using only the alignments of word boxes.

3.2 Brick Wall Coding Retrieval Overview

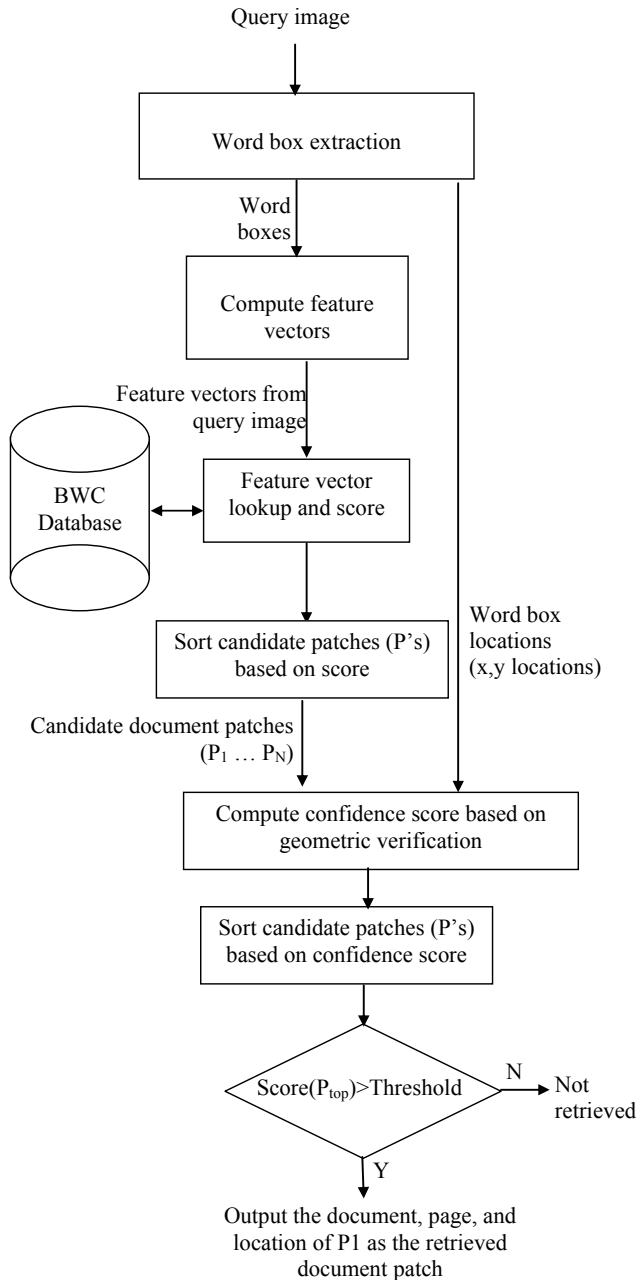


Figure 4. Overview of the BWC retrieval algorithm.

An overview of the document patch retrieval algorithm is presented in Figure 4. The BWC database indexes BWC feature vectors from printed or scanned documents. When an image that contains a patch of text is received as a query, word bounding boxes are extracted and represented with Brick Wall Coding (BWC) feature vectors. BWC feature vectors are submitted as queries to the database and the original document patches that contain BWC features in the query image are retrieved with hash-lookup. The retrieved document patches are sorted based on the number of BWC features they contain. Then, the top N candidate document patches are identified.

In the geometric verification step, the top N patches, $P_1 \dots P_N$, are scored based on the similarity of the relative locations of their descriptors to those of the query image. After the scores are computed, the document patch with the top score (P_{top}) is output if its score is larger than a threshold. If not, the query image is rejected. In the next sections we explain each of these steps in detail.

3.3 Word Box Extraction

Currently most mobile phones do not have macro or autofocus capabilities. Therefore, identifying the word boundaries in a video taken by a mobile phone is non-trivial. Since we expect the majority of the video frames to be taken close to the document, the images will be blurry as shown in Figure 5.a. Therefore typical edge-based text detection schemes will not be reliable. Global thresholding is also not useful, since the images generally exhibit lighting variations and vignetting which make the borders of the image darker than the center. Simple blockwise adaptive thresholds are not reliable because hard shadows may be cast on the page. There are techniques designed to overcome these problems, but they are usually too compute-intensive for mobile computation.

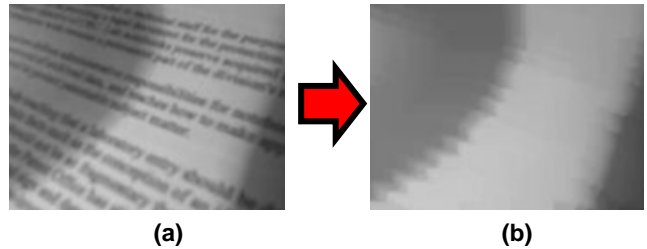


Figure 5. (a) Typical input image and (b) estimated background.

Our solution is to first estimate the background, i.e. paper color and lighting, at every pixel and then subtract this from the original image to obtain a difference map. This method works well in practice especially when the background is mostly uniform. Background estimation is accomplished using a continuous-scale morphological closing operator. A square structuring element is used for the closing operator, since the 2D operation becomes 1D-separable for rectangular structuring elements. 1D morphological operations are performed using the algorithm in [16]. The main advantage of the algorithm is that instead of computing max/min in a window, extrema over windows extending forward and backward away from regularly-spaced pivot points are computed. Computing extrema over these windows is very efficient; to find the maximum over a

window from position 0 to i , we compare element i to the maximum over the window from position 0 to $i-1$. We can then compute the extremum over a window centered on each pixel by comparing the appropriate extrema on either side of the nearest pivot. This technique, which uses only two comparisons per pixel per 1D sweep, is very efficient and effective. Figure 5.b shows the background estimation results.

With the estimated background image from the previous step, foreground binarization is done by subtracting the background from the original image. We used a global threshold for binarization which produces reliable results over a range of lighting conditions. Once the image has been binarized, connected components are identified through a scanline algorithm, which sweeps once through the image, labeling each foreground pixel with the unique ID number of the connected component to which it belongs. Example results from background extraction, binarization, and connected component analysis are shown in Figure 6. Once we obtain the connected components we perform iterative rotation and line projection to find the rotation angle and correct for the rotation.

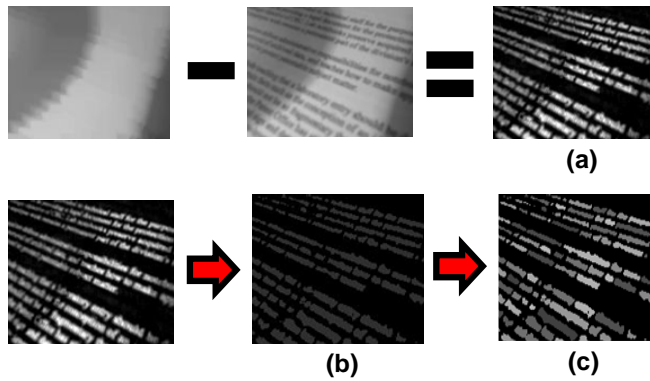


Figure 6. Sample results of the (a) background subtraction, (b) binarization, and (c) connected component analysis.

3.4 Brick Wall Coding Features

The feature we propose for representing bounding boxes is a local feature that is scale invariant and it is robust to slight perspective distortion. Robustness to slight perspective distortions is sufficient for most practical applications that we envision since we expect users to cooperate with the application and hold the document relatively parallel to the camera.

3.4.1 Length in Nubs

After word boundaries are identified in the query image, we compute the aspect ratio of each word bounding box. We refer aspect ratio as its length in nubs, U . $U=w/h$, where w is the width and h is the height of a word box in pixels. U is scale invariant.

Readers should note that, even though we assign the numerical value as *length in nubs* to each word, assignment of other numerical values to each word box is possible. The value for each word box can be the number of characters it contains, number of holes, ascenders, descenders, vertical lines, horizontal lines, connected components in the word box, density, mass center, or any other numerical value that represents the word box.

3.4.2 Word Clusters

After the length in nubs is computed for each word box, local features are calculated based on word clusters that capture the layout of word boxes. It is possible to select many different combinations of words as clusters. Our intuition was that word clusters need to be small and spatially localized to make the recognition more robust to the potential word box computation errors. This way, if one part of an image contains effects such as shadow or motion blur, small and spatially localized word clusters on parts of the text image that do not have such effects would be easier to retrieve correctly. However if we select a word cluster that is very small, for example a cluster that contains just the word itself, or itself as well as its horizontal neighbor (horizontal 2-gram), it may not be discriminative enough to uniquely identify a document patch.

After performing error case simulations with horizontal and vertical word clusters, BWC word clusters were defined as word boxes that have both above-horizontally overlapping and below-horizontally overlapping words, and at least two above- or below-horizontally overlapping words. Examples of word clusters are given in Figure 7.

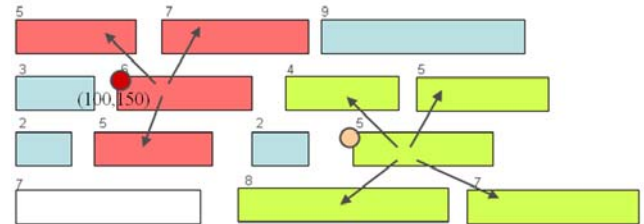


Figure 7. Computation of feature vectors from word clusters.

3.4.3 Feature Vectors

A feature vector, \vec{F} , is computed for each word cluster as the length in nubs of the word box in the cluster center, U_c , the length in nubs of the above words, \vec{U}_a , and the length in nubs of the below words, \vec{U}_b , i.e. $\vec{F}_i = \{U_c, \vec{U}_a, \vec{U}_b\}$. Figure 7 shows an example of the feature vector computation. The feature vector corresponding to the location of the red circle is $\vec{F}_i = \{6, \{5,7\}, \{5\}\}$. The location of each feature vector, which is the coordinates (x_i, y_i) of the upper left corner of U_c , is also stored.

Because video frames can be blurry, the computation of U is prone to quantization errors. To accommodate this, more than one feature vector is computed for each vector location (x_i, y_i) . We compute the fraction of pixels remaining from w/h , in order to determine our confidence in the value of U . If the fraction, $f=w \bmod h$, is less than a threshold T , then the word box is also assigned an alternative value that is $U-1$. If it is larger than a threshold K , then the word box is assigned an alternative value $U+1$. It is important to note that considering a word cluster may contain several word boxes, the number of feature vectors will grow rapidly when all the alternative U values are used for computation of additional feature vectors. In our application, we

limit the number of feature vectors that can be assigned to a location to 64.

3.5 Indexing and Database

In order to retrieve document patches using BWC, the original documents, books, and magazines need to be indexed. For documents, a specially designed printer driver saves high resolution bitmap files for each printed page. If an electronic file for printing is not available, scanned documents can also be indexed. In this case, the indexer uses commercial OCR software [17] that outputs word boxes for document images. We do not employ our word extraction algorithm because it is designed mainly for blurry text images, not the type of images captured by printing or scanning. After word boxes are identified using [17], word clusters are formed and feature vectors and vector locations are computed for each word cluster as described in Sections 3.4.2 and 3.4.3.

Because we are not only interested in identifying the page of a document, but also location on the page, the document image is divided into $M \times N$ overlapping segments S_k . The size of overlapping segments, $M \times N$, is determined based on the approximate viewing area on the document that a mobile device captures. Obviously this would depend on many parameters, such as the distance of the camera from the paper. We make an approximation and in our implementation we select $M=3''$ and $N=2''$. The values of M and N in number of pixels would depend on the dpi (dots per inch) resolution of the captured document page.

Once document image is divided into overlapping segments, the (x_i, y_i) location of each feature vector \vec{F}_i is used to determine which segments a feature vector belongs to on a document page. Because segments overlap, a feature vector may belong to several segments on one page.

The document database is comprised of a hash table. The hash table has the feature vector as a key, and the document id, document page, patch id S_k , and the location of the feature as key-values. Next we describe how retrieval works.

3.6 Retrieval

The retrieval process first finds the indexed document patch candidates that could potentially contain a given query image. It then applies a geometric verification algorithm to determine which candidate best matches the patch.

3.6.1 Hash look-up and scoring

Each feature vector obtained from the query image is used for hash look-up to obtain a list of document patches that contain this feature vector. The retrieved document patches include information such as document id, page id, and patch id. For each feature vector, the score of document patches that contain this vector is incremented by one. Then, the document patches are sorted based on their hash look-up scoring. The highest scoring patches are the ones that contain the highest number of matching feature vectors. The first N indexed patches are determined to be the document patch candidates, $P_1 \dots P_N$.

3.6.2 Geometric verification

Different document patches may contain similar feature vectors. This could potentially degrade retrieval accuracy, particularly when dealing with large document databases. To address this problem, we compute a confidence score based on how well the relative locations of feature vectors in the query image match those in the candidate patches.

Geometric verification needs to be fast and robust against small deformations of the paper, such as the bending of a document. We compute angles, θ , between locations of pairs of feature vectors in the query image and compare them to the angles between the locations of same feature vectors on the document patch candidate. If the two angles are similar (i.e. the L1 norm is smaller than a threshold), then the confidence score for the document patch candidate is incremented by one.

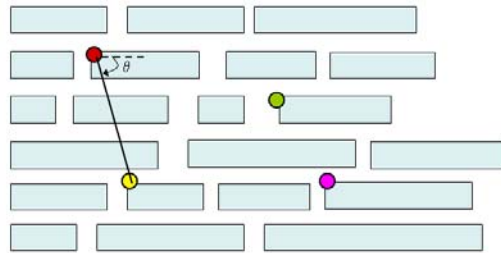


Figure 8. Computing a similarity score based on the relative locations of descriptors.

Once the confidence scores are computed between the query image and each document patch candidate, the candidate with the highest score is selected and compared to a threshold to determine if the match is good enough. If it is, then a matching document patch is found.

4. EXPERIMENTAL RESULTS

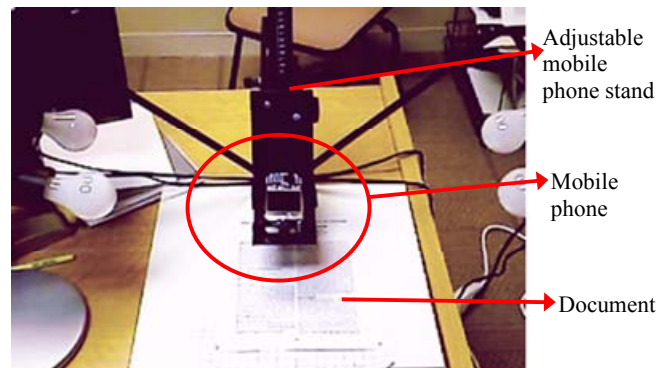


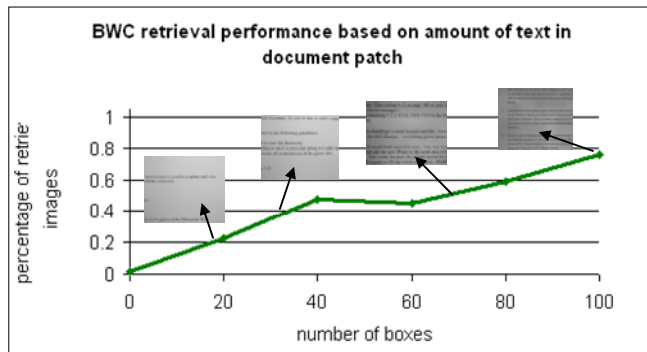
Figure 9. Device for capturing and ground truth.

Our applications require not only retrieving the original document page, but also location on a page. It is challenging to collect and ground truth query images with this information. We built a device, as shown in Figure 9, that can hold a Treo 700w mobile phone and allows us to capture and ground truth document patches systematically.

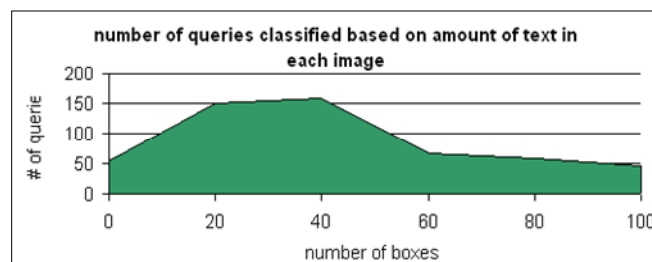
4.1 Retrieval Performance

We tested the document patch retrieval accuracy by indexing 4397 document pages using BWC features. Approximately 20Kbytes of storage per indexed document page is required. We also collected and ground truthed 533 mobile phone images, using the device shown in Figure 9. The captured image size was that of a small video frame, i.e. 176×144.

In Figure 10.a we present the change in retrieval rate based on the contents of the query document patches. Figure 10.a shows the distribution of query images based on the amount of text boxes they contain. As can be seen from the figure, the accuracy of BWC largely depends on the number of word boxes present in the document patch. If the document patch contains very few words then there is not enough information for document retrieval. If the document patch contains 8 or more lines, the retrieval rate is 60% or better. Examples of document patches that are recognized by the BWC algorithm are presented in Figure 11. Examples of document patches that cannot be recognized are presented in Figure 12. These images contain mostly figures or a small number of word boxes.



(a)



(b)

Figure 10. (a) BWC retrieval performance based on number of word boxes in query images (b) distribution of query images based on number of word boxes that they contain.

Readers should note that 60% recognition rate is sufficient for the interactive multimedia applications that we implemented. Because we perform document retrieval at 4 fps on video input, document recognition feels instantaneous to the user. Also in our applications we use a very conservative confidence score, T_c , where the rate of the false positives, i.e. falsely identified documents, is less than 0.01%. In the results we present in Figure 10, there were no false positives.

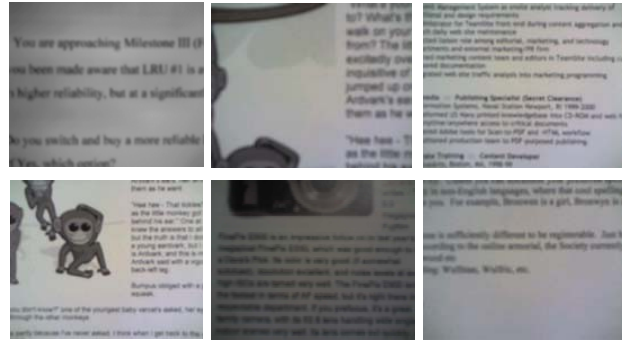


Figure 11. Examples of document patch frames that were correctly retrieved using BWC from a database of 4397 document pages.

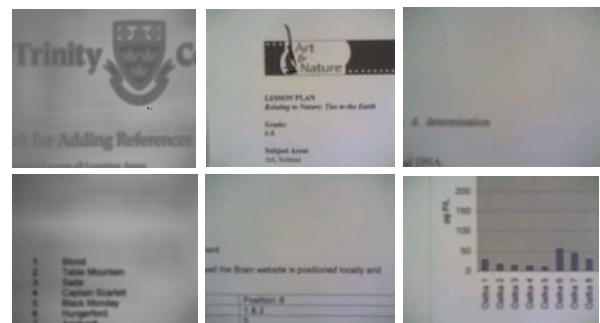


Figure 12. Examples of document patch frames that were not retrieved.

4.2 Computational Performance

The processing times required by each module are presented in Table 1. These results are obtained by performing 2200 queries with video frames that are 176×144 pixels. The database on the mobile device contained 129 document pages. As can be seen from the table, the majority of time was spent in word bounding box extraction. Readers should note that as the size of the database increases, hash look-up and geometric verification may take a larger percentage of the total computation time.

Task	Time (ms)
Bounding box extraction	172
Hash look-up and Geometric verification	83
Feature computation	11
Other processes	13
Total Processing time	279

Table 1. Processing times for different BWC retrieval modules on Treo 700w.

5. APPLICATIONS

5.1 Interface

Several HotPaper interfaces were implemented in C++ and in C# in the Windows Mobile environment using the .Net framework. The interface for electronically writable paper applications, such as the real estate guide and meeting agenda, is presented in Figure 13. In these applications, the database is copied to the phone when users sync with their computer. The application interface includes a video window where users can preview video when they are pointing the mobile phone at a document. When a document is recognized, the thumbnail of the recognized document is shown in the right window of the screen and a red rectangle is drawn around the recognized document patch. Once a document region is recognized, the user can press the 'select' button and add voice, photo, typed notes, and handwritten notes to the document patch. Selecting an annotation method and pressing the record button brings up the appropriate interface based on the selected media type. For a photo, a camera preview window is displayed, for voice, an audio recording interface is displayed, and for handwriting an empty notepad is displayed, etc. When the user stops recording, the media is associated with the document patch is saved in the media database on the phone.



Figure 13. Interface running on Treo 700w.

For playback, the user points the camera phone at a document patch. Once recognition is successful and a media file linked to the document patch is found, the phone vibrates and a media icon is displayed on the recognized document. The user can then press the play button to playback or view the media.

5.2 Real Estate Guide

House hunting is an activity where potential home buyers collect pictures, video, and notes related to the home they visit. In addition, most real estate agents prepare a paper printout for their client that shows pictures and information about the houses to be visited. The collection of media using a paper document is an ideal HotPaper application.

In the real estate guide application, users collect their annotations and photos about a house using the paper printouts. Before they start using HotPaper, users synchronize their mobile phone with

their PC and the database containing the printout and page thumbnails is copied to the mobile device. When visiting houses, users can add a picture or a comment about a house by pointing their camera to the section of the printout that contains information about the house that they are visiting. This is illustrated in Figure 14. Once the document patch is recognized, they press the "select" button, choose the annotation type, and record the annotation. As shown in Figure 14.a, b, and c, currently we support 4 annotation types: picture, handwritten annotations, typed annotations, and audio. Our interface has a placeholder for video annotations, as can be seen in Figure 13, but it is currently not supported.

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

During playback, users point their camera phone at the paper document. The document id, page id, and the (x_i, y_i) coordinates are retrieved using BWC. Then the hotspot file is loaded and the document ids, pages, and (x_i, y_i) locations of hotspots are compared against the retrieved location. If there are any hotspots on the retrieved document and the retrieved page, media icons are drawn on the thumbnail of the document page. If the user is pointing the mobile phone close to a hot spot, the phone provides tactile feedback and vibrates. Proximity is determined by computing the L2 norm of the difference between the retrieved location (x_r, y_r) and each hotspot on the page (x_i, y_i) .

Our system also allows synchronizing with the server, where the collected metadata is inserted into the original document that was used to obtain the printout. Currently we support annotation of MsWord files as shown in Figure 15. When the device is synchronized with the server, the hotspot XML file is received by the server and the annotations are automatically inserted as "comments" in the original document. Therefore the integrity of the document is preserved, i.e. if the user wishes to do so they can turn off the comments. An alternative to MsWord file modification would be creating an HTML representation so that users can access and view all annotations in one place. Using either the MsWord or HTML representation, users can access the annotated document at their computer, without using the paper and mobile phone interface, and share the multimedia annotations easily via e-mail.

5.3 Using Meeting Agenda or Conference Guide to Collect Multimedia

In meetings and conferences attendees create, use, and capture many forms of media, such as slides, videos, whiteboard, annotations, notes, documents etc. Most of the time it is very difficult to link all this information. In this HotPaper application, meeting agendas, conference guides, presentation slides, or any other related printed documents are used to collect and link this media. For this, we employ the interface described in Sections 5.1 and 5.2. Figure 16 illustrates the types of media that can be collected and linked using a meeting agenda, presentation slides, or a conference guide.

(a) audio annotation



a.1. User recognizes document patch by pointing the mobile phone at the paper document



a.2. User selects document patch, selects the audio icon, and records an audio clip



a.3. After the user completes audio recording, an audio icon appears on the document patch

(b) image annotation



b.1. User recognizes another document patch by pointing at the mobile phone at another part of the paper document



b.2. User selects document patch, selects digital camera icon, and takes a picture



b.3. After user takes the picture, a digital camera icon appears on the document patch

(c) handwritten notes and typed notes annotation



c.1. User recognizes another document patch by pointing the mobile phone at another part of the paper document. Once a document is recognized, previous annotations on the other document patches are also displayed.



c.2. User selects document patch, selects stylus, and starts recording



c.3. User is directed to a notepad where she/he can enter handwritten notes using a stylus



c.4. After user is done with a stylus entry, a stylus icon appears on the document patch



c.5. User selects text entry icon and entered typed notes for the same document patch



c.6. After user is done with the text entry, a keyboard icon appears on the document patch

Figure 14. Users can point their mobile phone to a paper document and annotate paper with (a) audio recordings, (b) digital images, and (c) handwritten and typed notes.

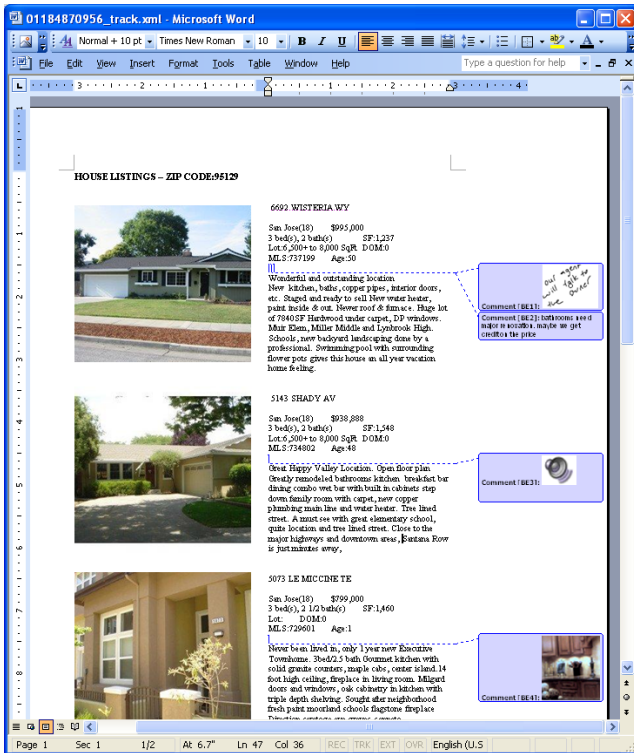


Figure 15. Collected multimedia clips are inserted as comments into the original document.

5.4 Multimedia Communication via Paper Interface

One common business task is collaboratively editing documents. Many document editing software packages, such as MsWord, support collaborative editing and user note insertion. On the other hand, many people still prefer reviewing documents on paper and

making corrections on paper even though communicating such comments is very difficult. As a solution, many people either duplicate the work by typing the comments when they have access to a computer, or by faxing the modifications. For the receiver, getting faxed changes back into the electronic document is a tedious and time consuming task.

HotPaper enables communication of a user's changes with ease using paper as the tangible medium. An example of such communication is illustrated in Figure 17. In this case, the user first points his mobile phone at a section of the paper and inserts images, voice annotation, scribbles, and typed notes. When he synchronizes his device with the server, the user whom he is collaborating with gets notified. Anytime from that point forward, when the second user points his mobile phone at the same document, he can retrieve the first user's annotations and comments.

5.5 Multimedia Textbooks and Collaborative Homework

Although our current interface is not designed for collaborating on homework, a natural extension of our work is to playback multimedia data and share comments using textbooks. Users can insert comments and give ratings on specific paragraphs and pages of a textbook that they are reading and see others' comments. They can link information to paragraphs in the book, such as urls of extra information, solutions to problems, video clips, and audio clips.

6. CONCLUSIONS AND OUTLOOK

The capabilities of mobile phones are ever increasing and their ubiquity makes them attractive devices for connecting our physical and electronic worlds. In this paper we presented a novel document patch recognition algorithm BWC that enables

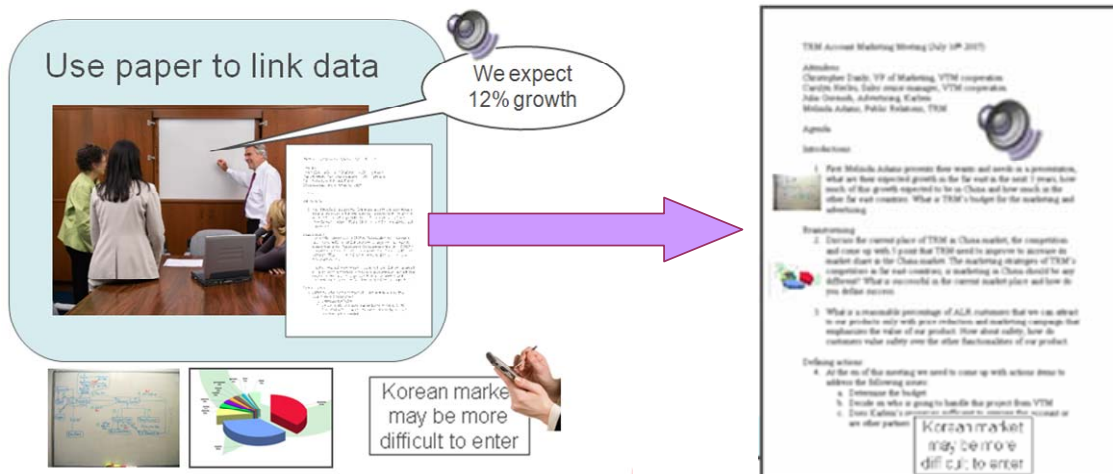


Figure 16. Users collect meeting artifacts, such as slides and whiteboard, using HotPaper application and printout of a meeting agenda.

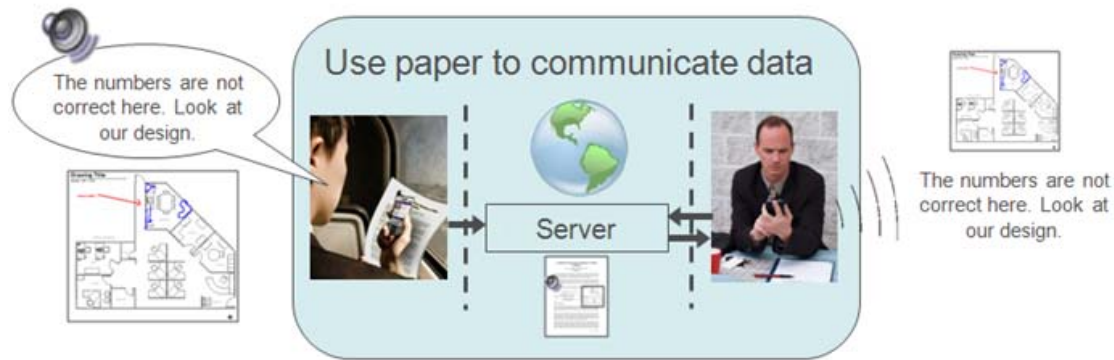


Figure 17. Paper documents is utilized as a tangible interface for communication of audio clips, pictures, notes by people who are working on the same document.

document recognition by analyzing a small number of blurry text lines in low resolution (176×144) video frames. BWC runs at 4 frames per second on a mobile phone and provides almost instantaneous document recognition. Our current algorithm can recognize document patches that contain at least 4-5 lines of text. How about document patches that contain fewer text lines or only images? These are some technology directions that our work can be improved on.

We also presented some novel applications that were implemented using BWC. These applications enable inserting electronic annotations to paper documents, books, and magazines. They also enable playback of these annotations using a mobile phone and accessing them through HTML or the original document, such as the MsWord file. Many other applications are possible using HotPaper technology, such as following urls on web page printouts using a camera phone, finding an electronic version of a document by simply imaging its paper printout, and accessing real-time information, such as stock quotes or classifieds, from newspapers and magazines.

Another clear future direction is to perform user studies to understand the value of paper-mobile phone interaction. Will users like being able to insert electronic annotations in a paper document and under what scenarios? Will they use such annotations themselves or share them with others? Will they use the mobile phone for retrieval of these annotations or will they prefer an HTML interface for presenting the information? There are many interesting HCI questions that can be explored with HotPaper technology.

7. REFERENCES

- [1] J. Wang, S. Zhai, J. Canny, "Camera Phone Based Motion Sensing: Interaction Techniques, Applications and Performance Study", ACM Symposium on User Interface Software and Technology, pp. 101-110, 2006.
- [2] A. Haro, K. Mori, V. Setlur, T. Capin, "Mobile Camera-based Adaptive Viewing", ACM Int. Conf. on Mobile Ubiquitous Multimedia, pp. 78-83, 2005.
- [3] M. Davis, M. Smith, F. Stentiford, A. Bambidele, J. Canny, N. Good, S. King, and R. Janakiraman. "Using Context and Similarity for Face and Location Identification", IS&T/SPIE Electronic Imaging Conf., 2006.
- [4] R. B. Yeh, C. Liao, S. R. Klemmer, F. Guimbretière, B. Lee, B. Kakaradov, J. Stamberger, A. Paepcke, "ButterflyNet: A Mobile Capture and Access System for Field Biology," ACM CHI, pp. 571-580, 2006.
- [5] Anoto Pen, <http://www.anoto.com/>
- [6] D. Hecht, "Printed Embedded Data Graphical User Interfaces," IEEE Computer, v. 34, no. 3, 47-55, 2001.
- [7] J. Graham, B. Erol, J. J. Hull and D. S. Lee, "The Video Paper Multimedia Playback System," ACM Multimedia Conference, 2003.
- [8] S. Klemmer, J. Graham, G. Wolff, J. Landay, "Books with Voices: Paper Transcripts as a Tangible Interface to Oral Histories", ACM CHI Conference, vol. 5, no. 1, pp. 89-96, 2003.
- [9] D. Schmalstieg, D. Wagner, "Experiences with Handheld Augmented Reality", IEEE/ACM ISMAR, pp. 3-15, 2007.
- [10] J. J. Hull, B. Erol, J. Graham, Q. Ke, H. Kishi, J. Moraleda and D. G. Van Olst, "Paper-based Augmented Reality," Int. Conf. on Artificial Reality and Telexistence, pp. 205-209, 2007.
- [11] D. G. Lowe, "Object Recognition from Local Scale-Invariant Features", Proc. of the Int. Conf. on Computer Vision, pp. 1150-1157, 1999.
- [12] Q. Liu, P. McEvoy, and C.J. Lai, "Mobile Camera Supported Document Redirection", ACM Multimedia Conf., pp. 791-792, 2006.
- [13] W-C. Chen, Y. Xiong, J. Gao, N. Gelfand, R. Grzeszczuk. "Efficient Extraction of Robust Image Features on Mobile Devices", IEEE/ACM ISMAR, pp. 287-288, 2007.
- [14] T. Nakai, K. Kise, M. Iwamura, "Use of Affine Invariants in Locally Likely Arrangement Hashing for Camera-Based Document Image Retrieval", Document Analysis Systems, pp. 541-552, 2006.
- [15] X. Liu and D. Doermann, "Mobile retriever - Finding document with a snapshot", Int. Workshop on Camera-Based Document Analysis and Recognition, 2007.
- [16] J.Y. Gil and R. Kimmel, "Efficient Dilation, Erosion, Opening, and Closing Algorithms", IEEE Transactions on PAMI, pp. 1606-1617, 2002.
- [17] Transym Optical Character Recognition software, <http://www.transym.com/>