

Mobile Retriever: access to digital documents from their physical source

Xu Liu · David Doermann

Received: 20 July 2007 / Accepted: 21 July 2008 / Published online: 26 August 2008
© Springer-Verlag 2008

Abstract In this paper, we describe an image based document retrieval system which runs on camera enabled mobile devices. “Mobile Retriever” aims to seamlessly link physical and digital documents by allowing users to snap a picture of the text of a document and retrieve its electronic version from a database. Experiments show that for a database of 100,093 pages, the correct document can be retrieved in less than 4 s at a success rate over 95%. Our system extracts token pairs from the text, to efficiently index and retrieve candidate pages using only a small portion of the image. We use token triplets that define the orientation of three corresponding tokens to effectively prune the false positives and identify the correct page to retrieve. We stress the importance of geometrical relationship between feature points and show its effectiveness in our camera based image retrieval system.

Keywords Document retrieval · View point invariant features · Camera phone

1 Introduction

1.1 Motivation and overview

As the trend towards paperless office leads to more documents being born digital (or digitalized), most of the physical documents that we see today have a corresponding electronic version. Many people however still feel more comfortable interacting the traditional paper documents because they are

easier to browse and can be more portable. Consider for example, a newspaper or a weekly periodical. Such documents are typically configured for browsing with a “front page” that allows the reader to quickly identify articles of potential interest [4]. While web or electronic access is growing in popularity, search is typically the preferred means of content selection, and this is typically done in front of a computer. We are exposed to printed material continuously throughout our day, and this is of great importance to publishers and advertisers. It is therefore very likely that physical and digital documents will coexist for the foreseeable future. Readers can benefit from both forms of documents if we can link them in a meaningful way.

Over the past several decades there have been numerous attempts to link physical documents with their digital representations. Pioneering research was conducted by Pierre Wellner et al. [13] in early 1990s. Cameras were mounted above the “Digital Desk” (Fig. 1a) to read the documents on it and understand human action. Arai et al. [1] attached a camera to a “VideoPen” (Fig. 1b) and explicitly used optical character recognition (OCR) to extract a small piece of text and retrieve the text from a “pattern buffer”. Since the “VideoPen” stays very close to the paper, however, only a tiny portion of a few words can be captured and the “pattern buffer” could store a limited number of words for matching. The primary application of these technologies was to edit the e-text.

An alternative application is retrieving the electronic document so that it can be presented to the user in an alternative form—as a summary, translated, on a custom WWW page, or perhaps read and delivered through a mobile device. Exbiblio (<http://www.exbiblio.com>) is an ambitious project which involves building a key-chain scanner and retrieval from very large scale document databases. All of these applications provide publishers and advertisers tremendous potential for

X. Liu (✉) · D. Doermann
Institute for Advanced Computer Studies, University of Maryland,
College Park, MD, USA
e-mail: liuxu@cs.umd.edu

D. Doermann
e-mail: doermann@umd.edu

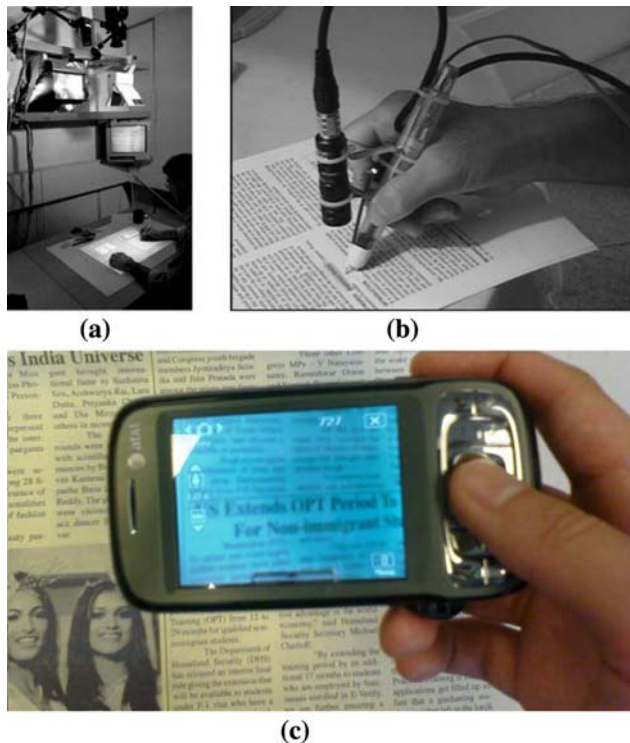


Fig. 1 Digital Desk (a), VideoPen (b) and Mobile Retriever (c)

overcoming the pain of reduced revenue caused by the WWW, and open up some very interesting research problems for document retrieval from printed material, and access to large document databases [3].

Historically, document retrieval may be categorized to text based or image based. Text based retrieval is relatively straightforward for known query terms because it is essentially a pattern matching problem which has been very well studied and is highly parallelizable. Image based methods are harder because there is no off-the-shelf meta-data to be used to form a query. Often the text is recognized with OCR and traditional text based methods are used, but image quality and articles with similar content can degrade accuracy. Alternatively unique image features can be extracted for the query and used to index the database. Associating an electronic text with the image is then trivial.

Consider the advantages of using a mobile camera phone as the potential link (Fig. 1c), where a snapshot provides access to the electronic source. With image acquisition capability, communication capability and portability, camera phones become a natural bridge between the physical and digital worlds. Document retrieval is one such “bridge” application which we call “Mobile Retriever”. The work flow of Mobile Retriever is illustrated in Fig. 2 with the green portion showing the physical world. The user snaps a picture of a piece of an article of interest and the pre-installed

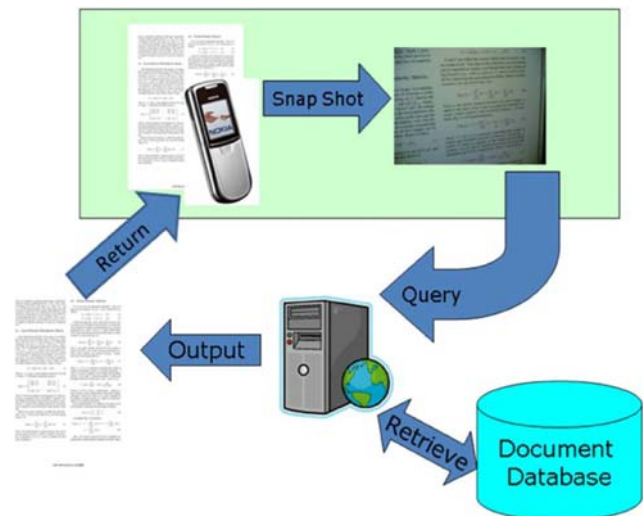


Fig. 2 Overview of Mobile Retriever

software sends the picture (or features extracted from the picture) to the server as a query. If the picture matches a document in the database, a visual or textual representation will be returned to the user. Sophisticated retrieval and matching algorithms may be applied on the server side, but all the user has to do is to capture a part of the document.

1.2 Application scenarios

Using mobile document retrieval we can link virtually any physical document to its digital version and promote many interesting applications. For example, one can tag an article that s/he has not finished reading, have it available for access later. Retrieving and downloading a desired article may cost a few cents providing alternative revenue, while providing targeted access that benefit both the reader and publisher. Considering related advertising opportunities this service could even be free. A similar goal can be achieved by attaching pages with “barcodes” [7] and have the user scan the barcode, but such solutions have met resistance because they require changes to the status quo. More importantly, by retrieving according to the piece of document captured, the system knows which part of the page the user is targeting at and can provide more targeted feedback according to the query such as keyword explanations or hyperlinked references. We can also help the visually impaired read newspaper articles using mobile document retrieval plus text-to-speech (TTS) technology. The visually impaired user can take a not necessarily perfect snapshot from the newspaper and have the handset read the retrieved article. Likewise, a retrieved digital document may be translated to any language using machine translation and be presented electronically.

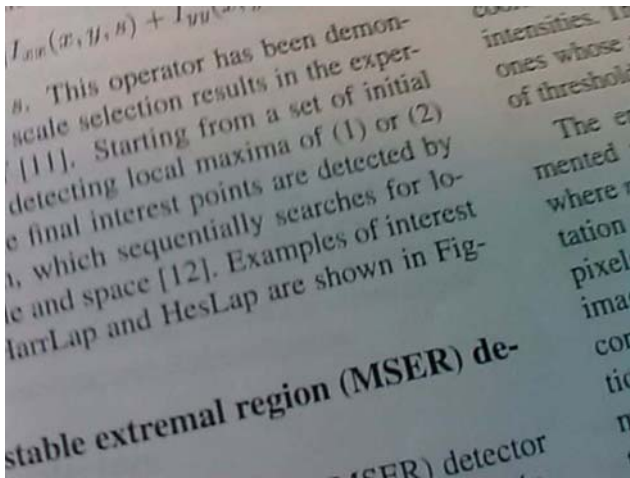


Fig. 3 A typical query image of Mobile Retriever

1.3 Challenges

A typical query to the Mobile Retriever system is shown in Fig. 3 and it illustrates three major challenges. *First*, the picture may only cover a small portion of the page because it is impractical to force the user take a full page snapshot. Therefore any retrieval technique that relies on the global structure of whole page may not be applicable. *Second*, the picture is taken by a low resolution camera. The contrast and sharpness is largely affected by the environment lighting. Typically the embedded camera has a fixed focal length which is set for shooting general scenes or portraits, so a document that is captured too close to camera may not be well focused. Character recognition will have low accuracy on these images and therefore text based retrieval will not be practical. *Third*, the captured image is subject to perspective distortion (including rotation, translation) and therefore the retrieval must be based on features that are invariant to these distortions.

Given these challenges, some specific questions need to be answered to build the Mobile Retriever system. From a degraded image with perspective distortion, how can one extract stable information which can be used for retrieval? How can one efficiently index large numbers of such documents? Since the captured image covers only a small portion of the page, what can be used to guarantee the accuracy of retrieval? From the user's point of view, if the captured document is not in database, when shall we reject the query? For evaluation, what is the false negative rate? These are the questions that we will address in the remainder of this paper.

1.4 Related work

Researchers have presented document image retrieval techniques using the query of layout [6], signatures [11] and logos [5] but these may not be suitable for our Mobile Retriever

concept because the global structure (layout) may not be captured in the partial snapshot and signatures and logos are of ten absent. As stated above, the query we used for retrieval is a low quality image with perspective distortion. It is impractical to use features that rely on the fine structures of the characters or are vulnerable to perspective distortion. Nakai and Kise [8] proposed locally likely arrangement hashing (LLAH) which computes affine invariant from the coordinates of four nearest neighbors of each word as a hash-code for indexing and retrieval. But the nearest neighbors may vary as view point changes. Liu and Doermann [14] proposed a Layout Context feature extracted from the camera captured document image. Layout Context can be used to efficiently index and retrieve documents but is not invariant to strong perspective distortion. It has not been quantitatively estimated in previous researches how many words should be captured for a success retrieval from a large database ($>10^5$ pages). More importantly, existing document image retrieval systems seldom discuss when to reject a query if it is not in the database.

For Mobile Retriever we use “token pairs” defined as words with close proximity in the image. Shape coding is used for indexing and geometrical invariant “token triplets” are used for verification to prune the false positives. Compared to existing approaches our shape coding and token-pair-triplet approach is more tractable because the features are extracted from the words and their relative positions. We test its performance on a database of 100K pages, which is ten times larger than the latest result reported in [8].

In Sect. 2, we present our solution to index the documents using a query of a partial document image with degradation. We also define the token pairs and triplets. In Sect. 3, we show how they are used for retrieval and verification. In Sect. 4, we briefly describe the system architecture of Mobile Retriever and data collection. Details of our experiments measuring performance and accuracy of retrieval are presented in Sect. 5. We conclude and discuss possible extensions in Sect. 6.

2 Indexing

As stated in the challenges, a camera captured document is highly vulnerable to character recognition. Shape coding was introduced as a document retrieval technique which is robust against character recognition errors [10, 12]. The basic idea is to reduce the surface form of glyphs using only low level features such as width, height, peaks and valleys. One way to implement this idea is to represent characters which may be confused with each other with a single representative symbol. For example, letter c and e are often confused because of their similar appearance, so we can remove letter e from the alphabet and replace letter e with letter c in the indexed

document. Table 1 shows part of the reduced alphabet we use in Mobile Retriever. Each group of characters is substituted by a representative symbol. This alphabet is built using the confusion matrix obtained from the OCR software which processes the query image. Having fewer letters in the alphabet may result in some information loss and confusion for some words, e.g. “cat” and “eat”, but with a sufficient number of tokens in the query accurate retrieval can still be achieved.

Bigrams and trigrams [2] are intensively used in language models. To utilize the geometrical relationship of words we generalize the concept of bigram to a pair of tokens which are geometrically close to each other (not necessary in reading order). A token pair is much more unique than a single token because it captures the layout context.

Figure 4 shows five token pairs centered at the word “searches”: searches-detected, searches-Examples searches-points, searches-are, and searches-for. K-nearest (we use $K = 5$) neighbors are indexed but only the nearest neighbor is used for retrieval. The nearest neighbor may vary as point of view changes but since we index K-nearest neighbors it will be covered by one of the indexed pairs. For a page with N words there will $K \cdot N$ token pairs in the index and the indexed tokens are substituted using the reduced alphabet.

A token triplet is an even stronger geometric structure. A token triplet consists of three words in the image and an orientation (clockwise or counterclockwise) of the three

Table 1 Reduced alphabet for indexing

ceoCDQ0→c	IijJLl1→i	f,t→f	68→6
7Zz→z	9gq→g	BEF→B	ri→n
YVyv→v	li→h	tn, rn, nn→m	cl, el→d

Example:
 In this paper we describe an image based document retrieval system
 →
 in this paper we descnbe an image bascd documcnf rcmevai svsfem

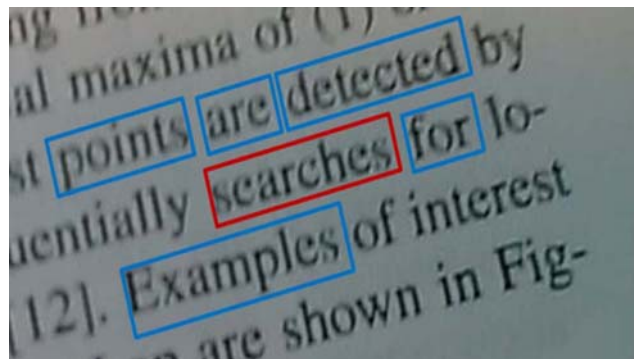


Fig. 4 Token pairs

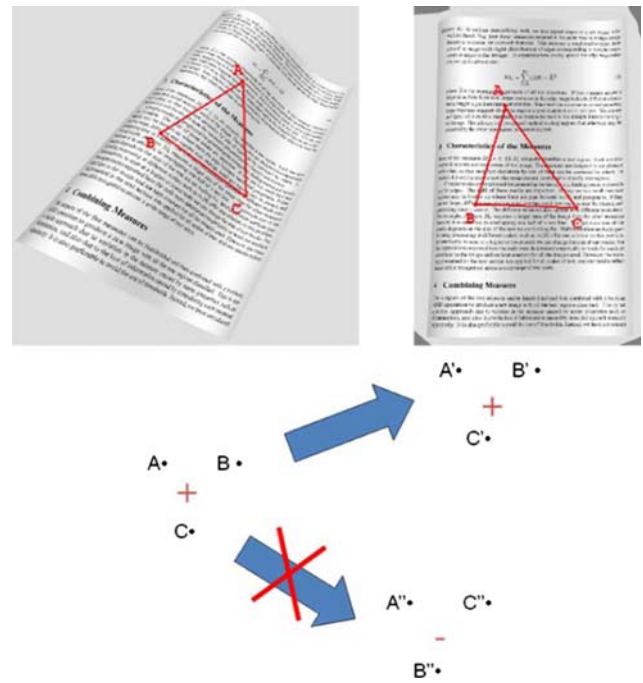


Fig. 5 Token triplets

words. It is invariant against viewpoint change or even page distortion. As shown in Fig. 5, the orientation of the triplet (A, B, C) on a page is defined as

$$\text{Sign} \left(\begin{bmatrix} X_A & Y_A & 1 \\ X_B & Y_B & 1 \\ X_C & Y_C & 1 \end{bmatrix} \right), \quad \text{Sign}(T) = \begin{cases} 1 \cdots T \geq 0 \\ -1 \cdots T < 0 \end{cases} \quad (1)$$

When the page is viewed from another angle, these three points appears as A', B' and C' and we have

$$\text{Sign} \left(\begin{bmatrix} X_A & Y_A & 1 \\ X_B & Y_B & 1 \\ X_C & Y_C & 1 \end{bmatrix} \right) \times \text{Sign} \left(\begin{bmatrix} X'_A & Y'_A & 1 \\ X'_B & Y'_B & 1 \\ X'_C & Y'_C & 1 \end{bmatrix} \right) = 1 \quad (2)$$

The orientation uniquely defines the geometrical relationship of these three words (A, B, C) and is robust against perspective distortion or surface bend.

When a point set S in the captured image is matched to another point S' in the indexed page, we define the match score as

$$\sum_{A, B, C \in S} \left(\text{Sign} \left(\begin{bmatrix} X_A & Y_A & 1 \\ X_B & Y_B & 1 \\ X_C & Y_C & 1 \end{bmatrix} \right) \times \text{Sign} \left(\begin{bmatrix} X'_A & Y'_A & 1 \\ X'_B & Y'_B & 1 \\ X'_C & Y'_C & 1 \end{bmatrix} \right) \right) \quad (3)$$

For a page with N words there are $\binom{N}{3}$ token triplets which is too large for indexing for $N > 200$. Therefore we use the token triplet for verification instead of indexing. The token triplet is a strong cue to reject queries that are not in the document database.

3 Retrieval

The input to Mobile Retriever is an image captured at an arbitrary viewing angle as shown in Fig. 6a. We enhance the input image to facilitate tokenization. The enhancement consists of two steps. We first enhance the contrast using an adaptive Niblack [9] binarization (Fig. 6b). We then compute the approximate rotation using the bottom line of each word (Fig. 6c) with a Hough transform. Strictly speaking the captured image is a perspective image, but we are not using the exact position of each word. The image is de-skewed using the estimated rotation.

Finally the enhanced image is sent to an OCR engine and the recognized characters are transformed to the reduced alphabet. With the coordinate of each word, we extract token pairs and use these pairs to query from the document database. The pages in the database are ranked according to the hits of token pairs. Those pages with more hits of token pairs in the query are ranked higher. However we cannot simply return the highest ranked page as the result because: (1) There can be multiple highest ranked pages. (2) The query may not be in database. The highest ranked page is just the one that shares most token pairs with the query. (3) Due to recognition error or incomplete words (on the boundary of the captured image), the highest ranked page may not be the right one to retrieve.

We collect the H highest ranked pages and verify that the orientation of token triplets are consistent with the query. This step identifies the page if it exists in the database and rejects the query if it is not. We score the page candidates and increase its score by 1 if a triplet has the same orientation as in the query and decrease its score by 1 otherwise. If there

Table 2 Document database

CVPR2007	3741	CVPR2006	4001	CVPR2005	3501
CVPR2004	3397	CVPR2001	1934	ICCV2003	2719
ICCV2005	3376	ICCV2007	3160	ECCV2006	2676
ICASSP 2001	4140	KDD2007	1403	ICME2000	2045
ICME2003	2412	ICPR2002	4024	ICPR 2004	4638
ICPR2006	5660	SPIEPW2001	8381	SPIEPW2005	8975
SPIEPW2006	8959	SPIEPW2006	7593	SPIEPW2008	7460
WACV05	898	Other EBooks	6000	Total=	100093

are N words in the query, the highest score a page could get is $\binom{N}{3}$ and the lowest score is $-\binom{N}{3}$. Two questions are raised here. How many page candidates (H) need to be collected to cover the one to retrieve? How many words (N) are required for a successful retrieval? We will briefly describe our implementation and database and then experimentally answer these questions in the section of evaluation.

4 Implementation

4.1 Data collection

We collected 100,093 pages from the proceedings of the recent year computer vision and image processing conferences and evaluate the accuracy, usability and speed of Mobile Retriever on this data set.¹ The token pair index of these 100,093 pages is 974 M bytes. We do not build index with token triplets, as the token triplets are only computed when verifying the top pages retrieved using token pairs.

4.2 Client

Our Mobile Retriever client is implemented on Windows Mobile platform with at least a VGA (640×480) camera. The client uploads the captured image to the server and image processing and retrieval are performed on the server side. The reason we do not process image on the mobile device is that it is not yet powerful enough for intensive computing. It will take more time to process the image on device than to upload it to the server. This may be changed as more powerful CPUs are widely installed on devices.

The results of retrieval can be shown via a WWW browser on the mobile device. We have not designed a specific user interface for the Mobile Retriever because there are so many different platforms which are evolving way rapidly. The WWW browser is a standard component that is available

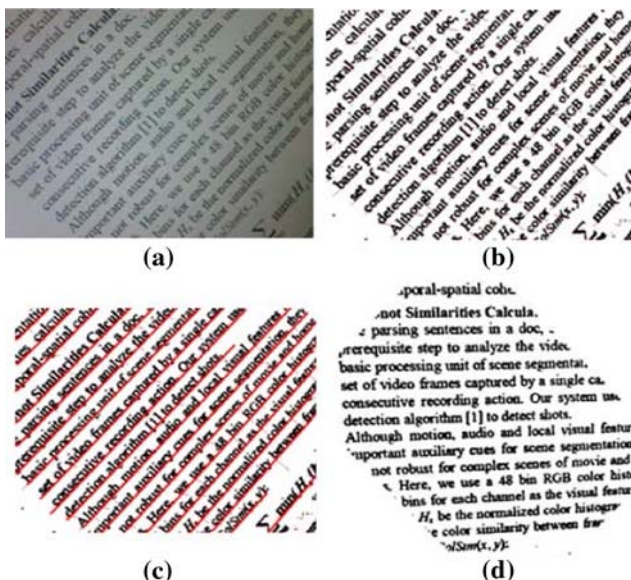


Fig. 6 Image enhancement

¹ A complete list of our collection is in Table 2.

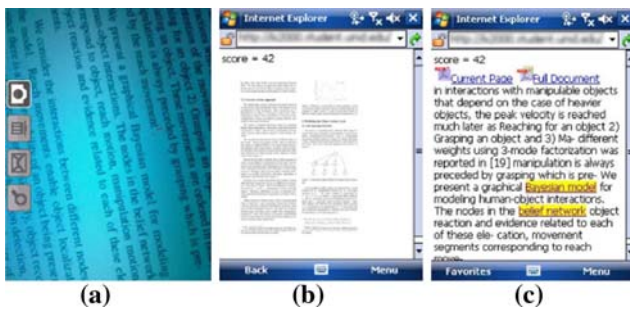


Fig. 7 Mobile Retriever on Windows Mobile 6: (a) snapshot, (b) thumbnail view of the full page, (c) detailed view of the captured texts

on almost all the mobile handsets and the appearance will be, in general, consistent. A typical retrieval result is shown in Fig. 7. It not only shows which page is retrieved but also shows the specific area of text in the query to demonstrate the capability of knowing exactly which part the user is looking at.

4.3 Server

Our server listens to TCP port 21 and use standard FTP protocol for receiving the query image. The server creates a unique ID for each request and redirect the client browser to the URL pointing to a dynamic page generated as the response of the request. The response either contains the retrieved page or a message of rejection.

5 Evaluation

We have tested the accuracy, usability and speed of Mobile Retriever on the collected data set with AT&T Tilt—A Windows Mobile 6 Pocket PC device at a camera resolution of 800×600 with auto-focus.

5.1 Retrieval using token pairs

5.1.1 Pages in the database

We first tested the accuracy of retrieval using only the hits of token pairs. We prepared 10 pages that are known to be in the database and captured 10 snapshots from each page. These 100 images (ID:1..100) are sent as queries to the Mobile Retriever system and rank the pages in database only according to hits of token pairs. In Fig. 8, we scatter plot the top ten candidates of each query. The pages that match the queries are marked with red stars. From this plot we can see that the correct page may not rank the highest among all pages. A histogram of the ranks of the correct pages is shown in Fig. 9a, where we can see that by returning the highest ranked page we only have an accuracy of approximately 18%. We draw the integral line of Fig. 9a in Fig. 9b. Figure 9b gives an esti-

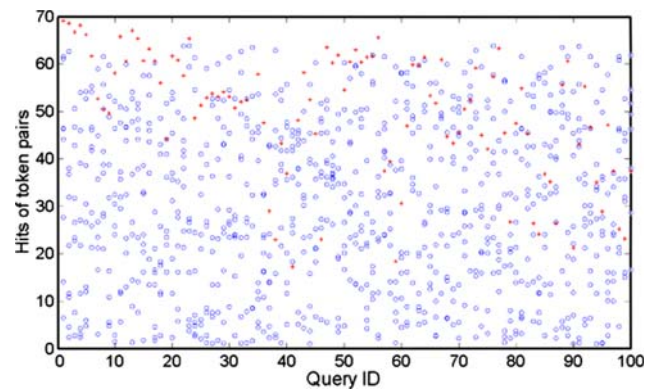


Fig. 8 Retrieving using token pairs

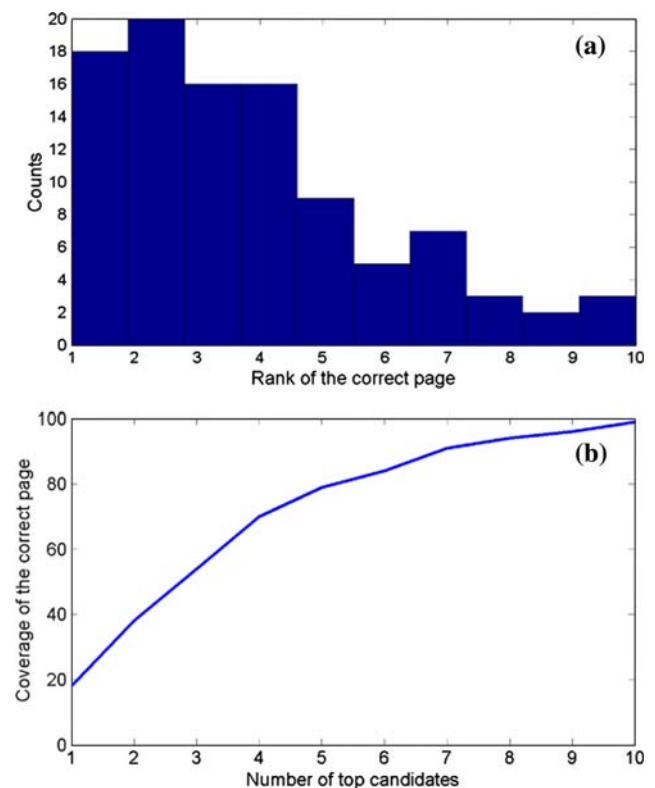


Fig. 9 The page that matches query may not be ranked 1st

mate of how many top candidate pages (H) we need to cover the one that matches the query. By taking the top five candidates we cover 80% of the correct pages and the coverage is increased to 99% if we take top 10.

5.1.2 Pages not in the database

Another issue which has not been given enough attention in the literature is when the query is not in the database. Some similar pages may be returned instead of being rejected. Although the inexact matches may provide information, for Mobile Retriever we would like the user to know if the exact

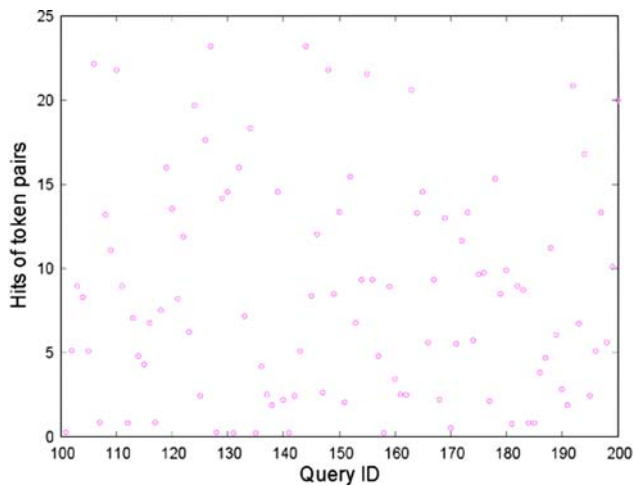


Fig. 10 Queries have no match in database still have hits of triplet pairs

document is present or not. This is important when related information (e.g. translation) is required. To test how token pairs work in this situation we collected 10 pages that are not in the database and took 10 snapshots from each page. We query the system using these 100 images (ID:101..200) and plot the topmost candidates for each query in Fig. 10. Comparing Fig. 10 (query not in DB) to Fig. 8 (query in DB), we find that using only the token pairs we cannot tell if a query is in database or not. Some of the queries that are not in database get more hits (>20) than the queries that are in database (<20).

The testing of token pairs shows that the hits of token pairs is insufficient for retrieval because the page that matches the query may not be ranked highest and those queries that are not in database cannot be rejected using only token pairs. These are the reasons that we need the token triplet verification.

5.2 Token triplet verification

5.2.1 Pages in the database

After ranking the pages using the hits of token pairs, we take the top 10 pages and apply token triplets verification. The scores of token triplets verification are computed using Eq. (3) and are shown in Fig. 11. Compared to Fig. 8, those pages that matches the query (red star) are distinguished from other candidates (blue circle). Therefore token triplets verification is able to pick the right one from the top pages ranked by hits of token pairs.

5.2.2 Pages not in the database

For those queries not in database, we also apply token triplets verification of the top 10 pages. Figure 12 shows the score

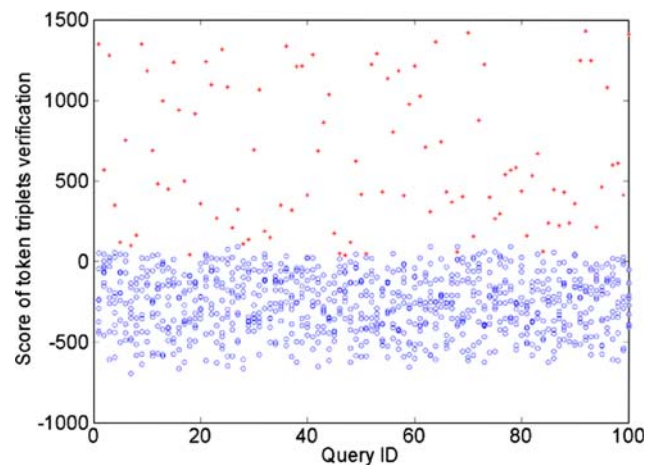


Fig. 11 Correct retrievals are distinguished after token triplet verification

of 100 queries that are known not to be in database. Their scores after triplets verification are in general below zero. Although these queries may have some common token pairs with a page in database, when it comes to the triplets, many of them are in the reversed orientation and therefore will result in negative score. Hence, using the triplets verification, we solve both the problem of picking the correct page and rejecting the queries that are not in database. Synthesizing Figs. 11 and 12 we draw the curve of false positive and false negative rate at different thresholds in Fig. 13. As the threshold goes higher, the false positive rate decreases and the false negative rate increases. An optimal choice of threshold is at 55 with false negative rate = 1% and false positive rate = 0. We normalize the score of triplets verification to [0..100] and return it to the user as the confidence of the retrieval.

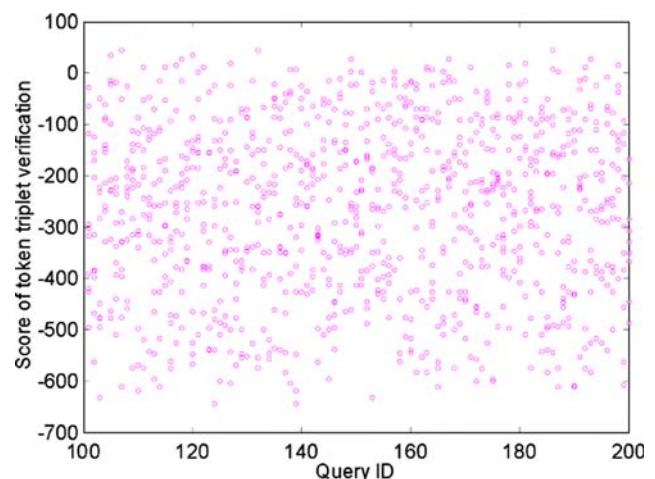


Fig. 12 Pages not in database have low scores after triplets verification

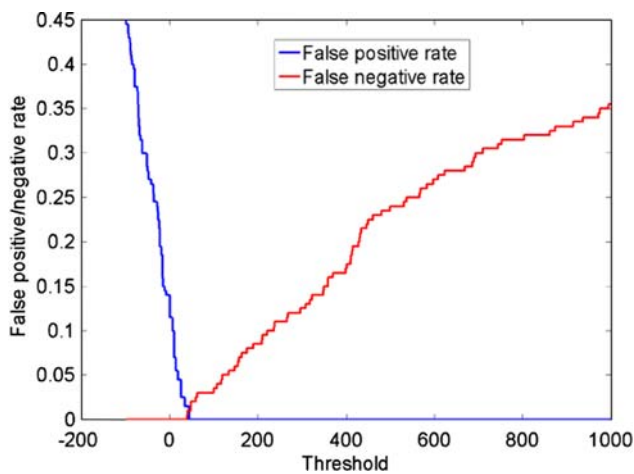


Fig. 13 False positive and false negative rate at different thresholds

5.3 Usability

5.3.1 Requirement of texts

The Mobile Retriever system is based on the token pairs and triplets which are geometrical relationships in the text context. It is important for a query to contain enough words for successful retrieval. It is not possible to estimate the number of words from the area of the captured image because different typographical settings result in different densities of words. To estimate how many words are required for a successful retrieval we ran a simulation on 100 randomly selected pages from the database. A random sized rectangle area is cut from each page as the query. The width of the rectangle is uniformly distributed between 0 and 1/2 page and the aspect ratio of the rectangle is set to 4:3 which is the same as the camera phone image. We record how many words are in each query and whether the retrieved page exactly matches the query. A histogram showing both successful retrievals and rejected queries is in Fig. 14. According to Fig. 14, 50 or more words in the query image will lead to a successful retrieval. Based on our statistic, there are approximately 750 words per page in conference proceedings. Considering image distortion and possible recognition errors, a capture of 1/3 page width will lead to a successful retrieval. This is fairly easy to achieve by a cooperative user.

5.3.2 Overall success rate

Ultimately Mobile Retriever will be evaluated on overall accuracy. When a user queries the system by taking a picture of a document, what is the chance that the system finds exactly the page that matches the query? We asked 10 users to choose 10 pages from our 100K-page database and perform a query for each page. We recorded how many pages were correctly retrieved. The only instruction they were given was

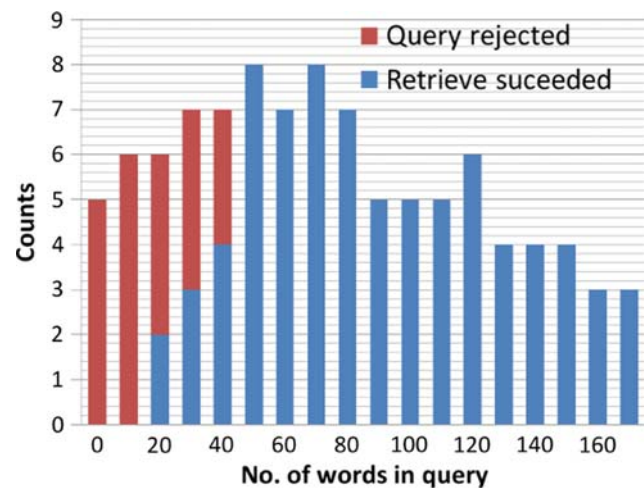


Fig. 14 Number of words in query vs. success rate

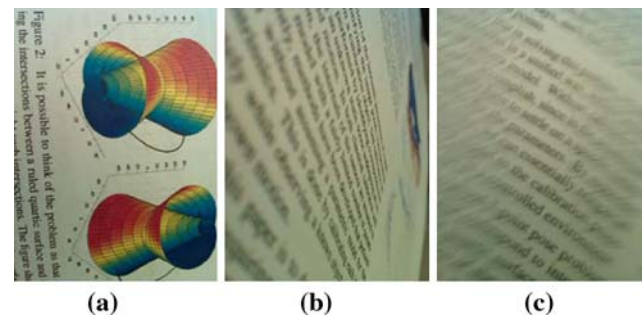


Fig. 15 Images failed to retrieve

to cover 1/3 page or more. Among these 100 queries only 4 of them are failed due limited text (Fig. 15a), too strong perspective distortion (Fig. 15b) motion blur (Fig. 15c) or network failure, resulting in a success rate of 96%.

5.3.3 Speed

Unlike a desktop/laptop environment, when users interact with mobile devices they always expect instant response. The speed of retrieval is a practical issue, therefore we measure the time spent on each retrieval task. The timer starts when the picture is taken and stops when the result of retrieving shows up. On average each query takes 3.9 s. From the server side, we also measure the time spent on each step of retrieval (Fig. 16). It turns out that most of the time is spent on image processing which is a constant factor. Second is sending the image to the server (we are using 10 Mbps campus wireless network). The actual retrieval of the page candidates using token pairs costs least and the verification is also a constant factor. It indicates that the time spent on each query will not increase much as the scale of document database grows large.

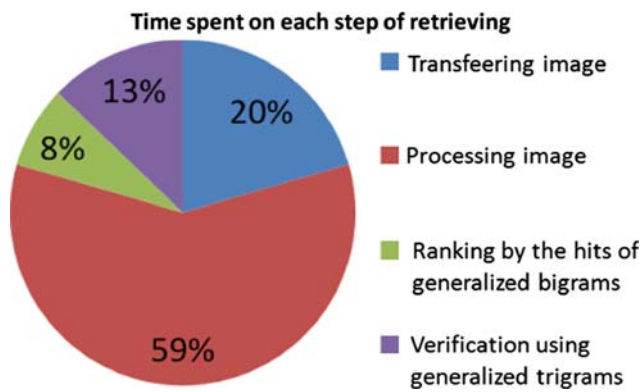


Fig. 16 Time spent on each step of retrieving

6 Conclusion and future work

In this paper, we have presented a camera based document image retrieval system, the Mobile Retriever, which employ the pervasive camera phone device as a client. Using Mobile Retriever one can retrieve the digital version of a document by taking a snapshot of a page, as if every page is given a hyper link. We propose token pairs and triplets which respectively retrieve and verify the candidates of retrieval at high speed and accuracy. We evaluate our system on a document database with 100K pages and the system can respond in <4 s at a success rate of 96%.

It worth mentioning that the concept of token pair and triplets is not bound to any language. Latin language has the natural unit of a word. If we apply shaping coding to any non-Latin language such as Chinese or Japanese, e.g. give a stable shape code to every character, the Mobile Retriever system can smoothly extend to non-Latin language documents. This will be the next step of our research.

References

1. Arai, T., Aust, D., Hudson, S.: PaperLink: a technique for hyper-linking from real paper to electronic content. In: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 327–334 (1997)
2. Brown, P., Mercer, R., Della Pietra, V., Lai, J.: Class-based n-gram models of natural language. *Comput. Linguist.* **18**(4), 467–479 (1992)
3. Doermann, D.: The indexing and retrieval of document images: a survey. *Comput. Vis. Image Understand.* **70**(3), 287–298 (1998)
4. Doermann, D., Rivlin, E., Rosenfeld, A.: The function of documents. *Int. J. Comput. Vis.* **16**, 799–814 (1998)
5. Doermann, D., Rivlin, E., Weiss, I.: Applying algebraic and differential invariants for logo recognition. *Mach. Vis. Appl.* **9**(2), 73–86 (1996)
6. Herrmann, P., Schlageter, G.: Retrieval of document images using layout knowledge. *Document Analysis and Recognition*. In: Proceedings of the Second International Conference on, pp. 537–540 (1993)
7. Kato, H., Tan, K.: Pervasive 2D barcodes for camera phone applications. *IEEE Pervasive Computing*, pp. 76–85 (2007)
8. Nakai, T., Kise, K., Iwamura, M.: Use of affine invariants in locally likely arrangement hashing for camera-based document image retrieval. *Lecture Notes in Computer Science 7th International Workshop DAS2006*, 3872:541–552 (2006)
9. Niblack, W.: An introduction to digital image processing (1990)
10. Smeaton, A., Spitz, A.: Using character shape coding for information retrieval. In: Proceedings of the 4th International Conference on Document Analysis and Recognition, p. 974 (1997)
11. Srihari, S., Shetty, S., Chen, S., Srinivasan, H., Huang, C., Agam, G., Frieder, O.: Document image retrieval using signatures as queries. In: Proceedings of the Second International Conference on Document Image Analysis for Libraries (DIAL'06), vol. 00, pp. 198–203 (2006)
12. Tan, C., Huang, W., Yu, Z., Xu, Y.: Imaged document text retrieval without OCR. *Pattern Anal. Mach. Intell. IEEE Trans.* **24**(6), 838–844 (2002)
13. Wellner, P.: Interacting with paper on the DigitalDesk. *Commun. ACM* **36**(7), 87–96 (1993)
14. Liu, X., Doermann, D.: Mobile Retriever—Finding Document with a Snapshot. *CBDAR 07*, pp. 29–34, September 2007