2012 10th IAPR International Workshop on Document Analysis Systems

Real-Time Document Image Retrieval on a Smartphone

Kazutaka Takeda, Koichi Kise and Masakazu Iwamura
Dept. of CSIS, Graduate School of Engineering
Osaka Prefecture University
takeda@m.cs.osakafu-u.ac.jp, {kise, masa}@cs.osakafu-u.ac.jp

***Abstract*—This paper presents a novel interface running on smartphones which is capable of seamlessly linking physical and digital worlds through paper documents. This interface is based on a real-time document image retrieval method called Locally Likely Arrangement Hashing. By just only pointing a smartphone to a paper document, the user can obtain its corresponding electronic document. This can easily provide the user with the information associated with the retrieved document. This relevant information can be superimposed on the display of smartphones. Therefore, we consider that with the help of this interface, the user can utilize paper documents as a new medium to display various information.**

***Keywords*-document image retrieval, reai-time processing, paper document, smartphone, human-computer interface**

I. Introduction

Recently, portable devices such as tablet PCs and smartphones become widespread rapidly. These devices create the worldwide demand for digital books. Along with this, various services for digital books have been provided. For example, "Qlippy" [1] provides the service by which readers of books can share their opinion and comments on favorite places of books. The opinion and comments are displayed on the electronic documents, which have high usability. On the other hand, many people still feel that the traditional paper is more comfortable because they are easier to read and more portable. In spite of this fact, there is no service for the physical documents; if the services such as "Qlippy" can be applied to paper documents, we think they become more useful.

In order to satisfy the requirement stated above, at least the following problem has to be solved. There is a big gap between physical documents and computers. We cannot add new digital information to paper documents themselves. In order to bridge this gap seamlessly, we must externally influence documents. In addition, in terms of usability, we should utilize portable and popular devices to satisfy this necessity.

As a method which can add the information to documents and utilizes the potable and popular devices, we consider the augmented reality (AR) system for the paper document using smartphones. This retrieves the electronic document corresponding to the captured page and superimposes on the captured image the information associated with the retrieved document. Some researchers have proposed the document image retrieval methods running on the mobile phone [2][3][4]. However, these methods do not have enough processing speed to realize AR. As a method solving this problem, we employ the document image retrieval method called Locally Likely Arrangement Hashing (LLAH) [5]. LLAH is known for its fast retrieval which enables a real-time processing. Additionally, LLAH can estimate the position and posture of the query image. These characteristics allow us to naturally display the relevant information. Therefore, we consider LLAH is usable for AR.

In this paper, we propose a real-time document image retrieval method using a smartphone. By using LLAH, we can realize AR to the physical documents. However, the problem about processing speed occurs when we run LLAH on smartphones. Due to this problem, the relevant information cannot be superimposed smoothly. To solve this problem, we implement two methods. One is to utilize tracking of captured paper documents, which is much faster than LLAH. By tracking, we can adjust the position and posture of the relevant information at a faster pace. The other is to employ multithread processing. For this system, we utilize four threads: capturing and display the query image, LLAH, tracking and drawing AR. From experimental results, we have confirmed that this system runs with retrieval accuracy of 95% and processing time of 10 fps. In addition, we propose various services using this system.

II. Related work

In this section, we overview previous methods of document image retrieval, some of which run on smartphones.

"Mobile Retriever [2]" is a retrieval method based on the token pairs and the token triplets. The token pair is defined as some shape codes of words whose characters are recognized by OCR. The token triplet consists of three words and an orientation of them. In this method, a snapshot of a page is captured by a smartphone and sent to the server to retrieve the corresponding document. Mobile Retriever has been confirmed to perform high accuracy with a large size of databases. However, it takes much processing time, about 4 seconds per query. This slow speed cannot satisfy our requirement to display AR naturally. This is because the position and posture of the smartphone change all the time.

"HotPaper" [3] is another method using features based on document text, which is called Brick Wall Coding Features

978-0-7695-4661-2/12 $26.00 © 2012 IEEE
DOI 10.1109/DAS.2012.71

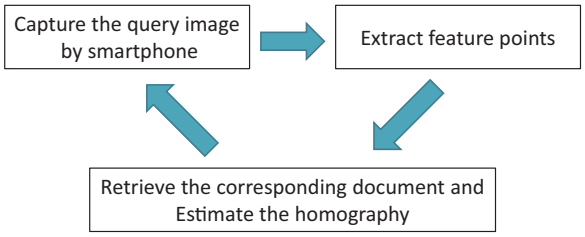

Figure 1. Processing of real-time retrieval.

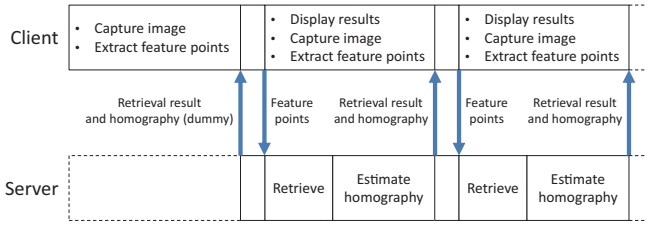

Figure 2. Client-server system.

(BWC). BWC is a local feature which represents bounding boxes of the words. This is scale invariant and robust to slight perspective distortion. This method has characteristics of fast processing, which is about 300[ms] per query. On the other hand, the size of database is very small, less than 5K pages. Moreover, the accuracy rate is only about 60% or better. This low scalability restricts the services.

In addition, Mobile Retriever and HotPaper have a shared problem. Because these methods use the spatial layout of words, they cannot work well for the documents written in languages such as Japanese and Chinese in which words are not separated.

"PaperUI" [4] is the document image retrieval method which is capable of solving this problem. PaperUI has seven approaches to identify a document; barcode, micro optical patterns, encoding hidden information, paper fingerprint, character recognition, local features such as SIFT [6], and RFID. Owing to utilize SIFT, PaperUI can retrieve a document without the influence of the difference of languages. However PaperUI is not capable of handling large-scale databases because SIFT features requires a large amount of memory. In order to supply the service to a huge number of pages, the ability to address large-scale databases must be needed.

In this paper, we employ Locally Likely Arrangement Hashing (LLAH) [5] which is capable of solving all the problems stated above. LLAH has high scalability and performs fast. We have confirmed that LLAH realizes the accuracy of 99% and the processing time of 50[ms] with a 20 million pages database [7]. Additionally, LLAH has already been extended for retrieval of documents in various languages [8]. Moreover, through the retrieval process, LLAH can estimate the position and posture of the query image on the electronic document. This allows us to easily decide the position and posture to display the relevant information. In terms of the characteristics stated above, we think LLAH is suitable for the requirement to display relevant information on smartphones naturally. However, since the smartphone does not have high processing power, the processing load of LLAH must be reduced to realize AR.

III. Software Design

A. Client-Server system

Real-time document image retrieval is realized by repeating the process as shown in Fig. 1. First, the query image is captured by the smartphone. Second, feature points are extracted from the query image. Third, the corresponding document to the query image is retrieved based on the feature points. For each feature point, a feature is calculated from the arrangements of surrounding feature points. The retrieval is realized by searching the corresponding feature points based on the features. In addition, we can estimate the homography from the correspondences of feature points. By using this homography, we can locate the captured region on the original page.

Because these steps can be executed independently, we can parallelize the processing by using a client-server system. In the implementation of this system, the client is the smartphone. Figure 2 shows the roles of the client and the server. At the client, the query image is captured and feature points are extracted from the query image. Note that these processes are executed in parallel. We explain it in detail in the next section. These feature points are sent to the server by using the TCP socket. In the server, based on the received feature points, the document is retrieved and the homography is estimated. At the same time, the client captures the new query image and extracts feature points. When the both processes of the client and the server are finished, the retrieval result and the homography are sent back to the client and the feature points are sent to the server again. By parallelizing the repeating process, this system is capable of realizing high frame rate.

B. Improvements

Next, we explain how to solve the problem about processing speed.

Adjusting the position and posture of an imaginary object to the real-world is important in realizing AR. Moreover, real-time processing is also significant to display the relevant information smoothly. By using the homography calculated through the retrieval process of LLAH, we can exactly decide the position and posture. However, in spite of the fact that LLAH is much faster than other existing methods, it still has a problem of processing time in running on the

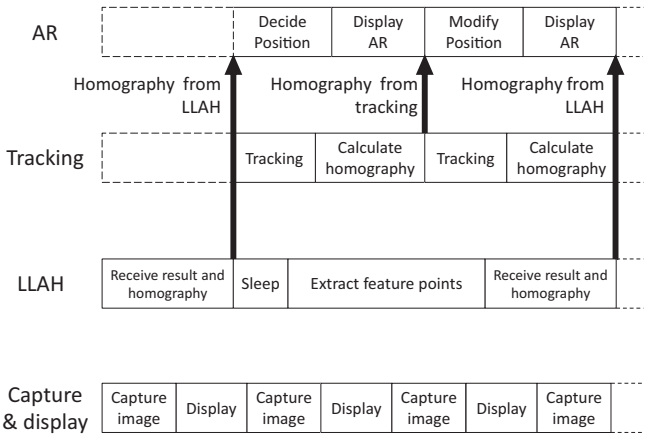

Figure 3. Multithreading and adjusting of position and posture by tracking.

smartphone. Because the smartphone is not powerful enough for intensive computing, we cannot display the relevant information naturally only by using LLAH. To solve this problem, we employ tracking of captured paper documents as well as multithread processing. These techniques allow us to realize AR to the document smoothly. We explain the details below.

1) Multithread processing: In order to improve processing speed, we employ multithread processing. In this improvement, as shown in Fig. 3, we utilize four threads: capturing and displaying the query image, LLAH, tracking and drawing AR. The efficiency to parallelize capturing the query image and LLAH is to smooth displaying the captured image. In addition, in order to reduce processing time per frame, we also parallelize the process of displaying the captured image and drawing AR. Note that in order to spare the time for other threads, the process of LLAH has the sleep time after each execution.

2) Tracking: The process of tracking is much faster than that of LLAH. Thus, by tracking the captured paper document in parallel with the process of LLAH, we can adjust the position and posture of the relevant information at a faster pace.

The way to track the paper document is as follow. First, we extract feature points to track the position of the paper from the initial frame. These feature points are extracted by using the Harris operator [9]. Figure 4 shows examples of feature points. Next, the optical flow of these feature points is calculated by using Lucas-Kaneda method [10] and they are tracked. Figure 5 shows the appearance of tracking. In this process, feature points that cannot be tracked are discarded. In addition, we set the minimum threshold of corresponding feature points between the previous and the present frame. When the number of the corresponding feature points becomes smaller that this threshold, we remove the previous feature points and extract new initial feature points. This threshold is determined experimentally.

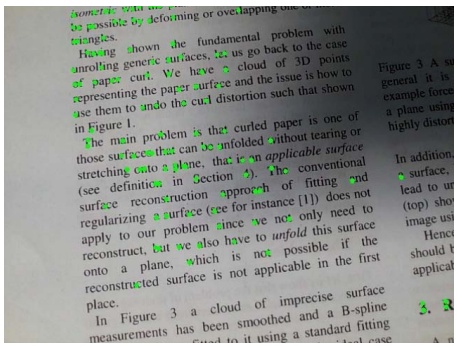

Figure 4. Examples of feature points for tracking.

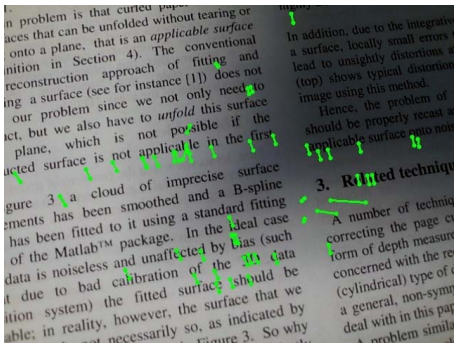

Figure 5. Appearance of tracking.

Tracking enables us to obtain the correspondences of feature points in successive frames. Then we can find out the homography from a set of the corresponding points by applying RANSAC [11]. Owing to this homography, we can modify the position and posture to superimpose the relevant information.

Figure 3 shows how to modify the position and posture by tracking. First of all, we decide the position and posture of the relevant information by using the calculated homography. Next, we also calculate the homography from the result of tracking and adjust the position and posture. Since tracking is fast but error-prone, we cannot apply this adjustment for many times. This problem is simply solved by applying more reliable estimation of homography by LLAH once in a few applications of tracking. By repeating the process as shown in Fig. 3, we can realize AR to the document naturally.

C. Application

We introduce some applications of this system.

1) Augmented Reality: As stated above, this system can realize AR to paper documents. An example of AR for paper documents is shown in Fig. 6. The AR is to display relevant information as an image superimposed with a captured query image. We currently consider as relevant information annotations by text, images, highlights, underlines and handwriting.

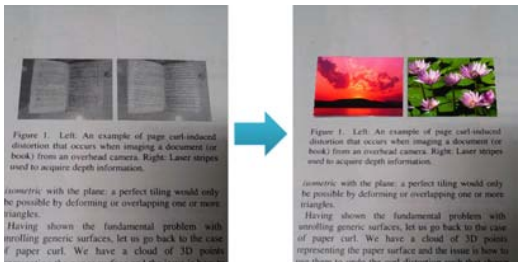

Figure 6. Example of AR for paper documents.

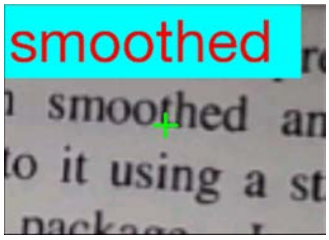

Figure 7. Example of word extraction.

Moreover, we can also superimpose 3D objects on the printed document. We consider that this technique allows us to utilize paper documents as a new medium to obtain extra information.

2) Word/Text Extraction: Another application is to obtain words and text in the captured region. We can get the position of words from the original PDF of the retrieved document. Because this system estimates the captured region on the retrieved document, the user can obtain the words in the captured region. Figure 7 shows an example of word extraction. We can see that the word centered at the image is extracted correctly. The merit of this technique is that the user does not need the character recognition. This allows us, for example, to search the meaning of words and obtain the explanation of the word.

3) Retrieval of the electronic documents: As a retrieval result of LLAH, we can obtain the document name of the query. Based on this document name, we can also obtain the original PDF and display it. Owing to this service, the user can save time to search the PDF file from the large number of files. Moreover, the user can write their opinion and the comments on this PDF. These annotations are shared and reflected in AR system.

IV. Experiments

We implemented the above services using the method proposed in this paper. In order to make such services usable, the system must realize high accuracy and fast processing. In the experiments, we investigated retrieval accuracy and processing time.

For the experiments, we made the database which includes 1 million pages. These documents were mainly collected from the Internet. The smartphone utilized in the experiments was with the 1.2GHz Dual core CPU and the 1GB memory.

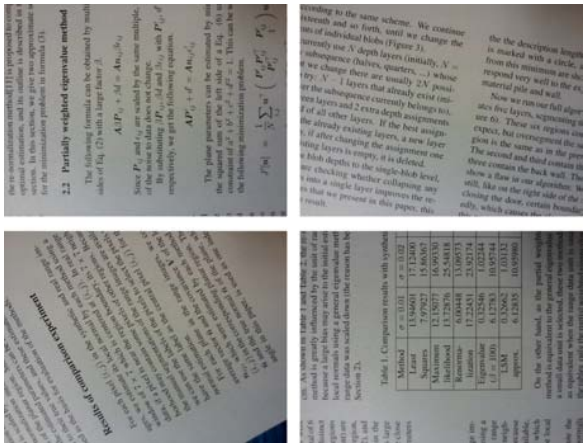

Figure 8. Examples of the query images that were correctly retrieved.

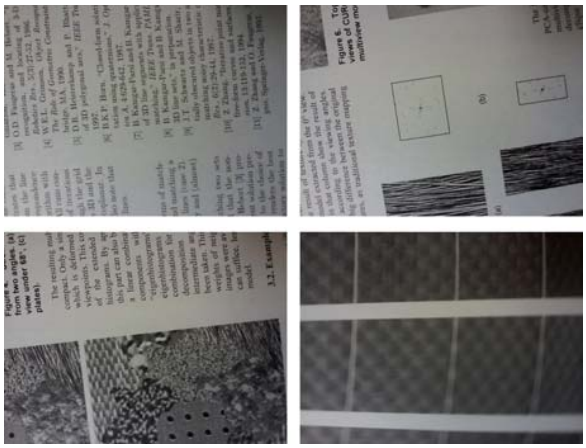

Figure 9. Examples of the query images that were not retrieved.

A. Retrieval Accuracy

We first tested the retrieval accuracy. For this experiment, we printed out 100 pages selected randomly from the database and captured each page from 5 viewing angles as the query images. Therefore, the number of the query images was 500. These query images include small parts of the entire pages.

From this experiment we obtained the accuracy of 95.2%. Figure 8 shows examples of correctly retrieved query images, which tended to include many words. On the other hand, Fig. 9 shows the examples of the queries that were not retrieved. Except for the left top image, these have less text. Because LLAH requires many characters to be captured to realize the stable retrieval, these images cannot be retrieved. The top left image could not be retrieved due to the instability of feature points; many dots and quotations

included in the query made the feature point extraction unstable [12].

In addition, we tested whether queries that do not exist in the database are rejected. We prepared 5 pages which was not contained in the database and captured them as queries. If the number of corresponding feature points is less than 4 the query is rejected. As a result of the retrieval, these queries were all rejected.

B. Processing Time

We also tested the processing time. In this experiment, we calculated the time for various processes.

First, time for capturing and displaying the image took only 40[ms], which is ignorably short as compared to other processes.

Next, we show the result of the process of LLAH. In LLAH, the client, which is the smartphone, has two processes; extraction of feature points and communication with the server. Note that we did not include the capturing the query image because it was in another thread. The feature point extraction required about 100[ms] and the connection took 2[ms]. Moreover, in order to spare the time for other threads, the process of LLAH has the sleep time of 100[ms] after each execution. As a result, the total amount time of the process of LLAH became 202[ms].

The process of tracking has the extraction of feature points and the tracking. The feature point extraction needed 278[ms] and the tracking took about 70[ms]. Because the feature point extraction is executed just only for initializing the tracking, and thus rarely applied, the time for tracking dominates the process. Thus it runs in 1000/70 fps, which is more than 10 fps.

Finally, we show the processing time of AR. The process includes the decision of the position and posture of the relevant information and drawing them. The decision of the position needed little time, but the drawing took about 70[ms]. As a result, the processing time of drawing AR was about 70[ms]. From the result of the tracking, this system can display the relevant information with the processing time of over 10 fps.

The processing time the user can perceive is from the AR. Since time for drawing AR was not faster than other processes, it dominated the user's perception. Thus, this system has the processing time of more than 10 fps.

V. Summary

In this paper, we introduced a real-time document image retrieval running on the smartphone by applying LLAH. In order to solve the problem about the processing time, we employed the tracking of the paper document as well as multithreading. From the experimental results, we have confirmed that this system realize the accuracy of 95.2% and can draw AR with over 10 fps. In addition, we also proposed some applications realized by using this system.

Our future work includes the improvement to speed up the processing time and the invention of more attractive services.

Acknowledgment

This work was supported in part by CREST, and the Grant-in-Aid for Scientific Research (B) (20300049) from Japan Society for the Promotion of Science (JSPS).

References

[1] http://qlippy.com/.

[2] X. Liu and D. Doermann, "Mobile retriever: access to digital documents from their physical source," *Int. J. Doc. Anal. Recognit.*, vol. 11, pp. 19–27, September 2008.

[3] B. Erol, E. Antúnez, and J. J. Hull, "Hotpaper: multimedia interaction with paper using mobile phones," *Proceeding of the 16th ACM international conference on Multimedia*, pp. 399–408, 2008.

[4] Q. Liu and C. Liao., "PaperUI," *Proceeding of the 4th International Workshop on Camera-Based Document Analysis and Recognition*, pp. 3–10, September 2011.

[5] T. Nakai, K. Kise, and M. Iwamura, "Use of affine invariants in locally likely arrangement hashing for camera-based document image retrieval," *Lecture Notes in Computer Science (7th International Workshop DAS2006)*, vol. 3872, pp. 541–552, February 2006.

[6] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, pp. 91–110, November 2004.

[7] K. Takeda, K. Kise, and M. Iwamura, "Memory reduction for real-time document image retrieval with a 20 million pages database," *Proceedings of the 4th International Workshop on Camera-Based Document Analysis and Recognition*, 2011.

[8] T. Nakai, K. Kise, and M. Iwamura, "Real-time retrieval for images of documents in various languages using a web camera," *Proceedings of the 10th International Conference on Document Analysis and Recognition (ICDAR2009)*, pp. 146–150, July 2009.

[9] C. Harris and M. Stephens, "A combined corner and edge detector," *Proc. Alvey Vision Conf*, pp. 147–151, 1988.

[10] B. Lucas and T. Kanade, "An iterative image registratio technique with an application to stereo vision," *Proceedings of 7th International Joint Conference on Artificial Intelligence*, pp. 674–679, 1981.

[11] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, pp. 381–395, 1981.

[12] K. Takeda, K. Kise, and M. Iwamura, "Real-time document image retrieval for a 10 million pages database with a memory efficient and stability improved LLAH," *Proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR2011)*, 2011.