On-line Document Registering and Retrieving System for AR Annotation Overlay

Hideaki Uchiyama, Julien Pilet and Hideo Saito
Keio University
3-14-1 Hiyoshi, Kohoku-ku
Yokohama, Japan
{uchiyama,julien,saito}@hvrl.ics.keio.ac.jp

ABSTRACT

We propose a system that registers and retrieves text documents to annotate them on-line. The user registers a text document captured from a nearly top view and adds virtual annotations. When the user thereafter captures the document again, the system retrieves and displays the appropriate annotations, in real-time and at the correct location. Registering and deleting documents is done by user interaction. Our approach relies on LLAH, a hashing based method for document image retrieval. At the on-line registering stage, our system extracts keypoints from the input image and stores their descriptors computed from their neighbors. After registration, our system can quickly find the stored document corresponding to an input view by matching keypoints. From the matches, our system estimates the geometrical relationship between the camera and the document for accurately overlaying the annotations. In the experimental results, we show that our system can achieve on-line and real-time performances.

Categories and Subject Descriptors

K.5.1 [INFORMATION INTERFACES AND PRESENTATION]: Multimedia Information SystemsArtificial, augmented, and virtual realities; I.4.8 [IMAGE PROCESSING AND COMPUTER VISION]: Scene AnalysisTracking

General Terms

Algorithms

Keywords

Augmented reality, Document retrieval, LLAH, Feature matching, Poes estimation

1. INTRODUCTION

Augmented reality is one of the popular research categories in computer vision and human computer interaction. AR applications are widely developed for game, education, industry, communication and so on. They usually need to estimate geometrical relationship between the camera and the real world to overlay virtual objects with geometrical consistency. One of the traditional approaches to estimate the geometry is to use fiducial markers [6]. In recent years, the research direction of AR is going towards using natural features in order to reduce the limitations of a practical use [16].

Nowadays, augmenting documents is gaining in popularity and is called Paper-Based Augmented Reality [5]. The purpose of this research is to enlarge the usage of a physical document. For example, a user can click words on a physical document through a mobile phone. This enables a document paper to be a new tangible interface for connecting physical to digital worlds. Hull et al have proposed a clickable document, which has some colored words as a printed hyperlink [5]. When the user reads the document, the user can click the colored word to connect the Internet. Also, the user can watch the movie instead of the printed picture on the document. Their application was designed for extending the usage of existing newspapers or magazines.

As a novel application for document based AR, we propose a system that registers a document image with some user annotations for later retrieval and augmentation on new views. Our system is composed of a single camera mounted on a handheld display, such as a mobile phone, and of the physical documents the user selects. No other special equipment such as a fiducial marker is necessary. The user captures the document, writes some annotations on the document through the display, and registers them in our system. When the user captures the registered document again, the annotations of the document are retrieved from the database and overlaid at the position selected by the user beforehand. Our system can be useful in case that the user does not want to write annotations directly on valuable documents such as ancient books.

The rest of the paper is organised as follows: we review keypoint matching based registration for augmented reality in the next section. In Section 3, we introduce the usage of our system. Then, we explain the detailed algorithm of our system in Section 4. We evaluate the way of capturing documents and processing time in Section 5, and conclude in Section 6.

2. RELATED WORKS

The process of registration by keypoint matching between

Augmented Human Conference April 2-3, 2010, Megève, France.

two images can be divided into three parts; extraction, description and matching.

As a first step, we extract keypoints which have distinctive and different appearance from other pixels in each image. By using these distinctive keypoints, it is easier to establish correspondences between two images. Harris corner [4] and FAST corner [13] are widely used and keep the repeatability of the extraction under different viewpoints.

Next, these keypoints are described as a high dimensional vector for robust and stable matching. The vector is usually computed with local neighbor region of the keypoint. The well-known descriptors such as SIFT [8] and SURF [2] with 128 dimensional vector are well designed to be invariant to illumination, rotation and translation change. Since the computational cost of SIFT is not sufficient for real-time processing, several attempts to reduce the cost have been done [14, 16].

Matching of descriptors can be addressed as a nearest neighbor searching problem. KD-tree based approach [1] and a hash scheme [3] are typical as an approximated nearest neighbor searching. Though the nearest neighbor cannot be always searched, the computational cost is drastically reduced compared to full searching. Nister and Stewenius have proposed a recursive k-means tree as a tree structure for quick retrieval [12]. Lepetit et al. have proposed another approach by treating the keypoint matching as a classification [7].

The descriptors such as SIFT and SURF are well suited to match keypoints with rich texture patterns. However, documents have generally repetitive patterns composed of text. Since the local region of documents may be similar and not be distinctive, these descriptors do not work well. Instead of them, geometrical relationship of keypoints have been proposed [5, 11].

As a descriptor for a document, Hull et al. have proposed horizontal connectivity of word lengths [5]. Nakai et al. have proposed LLAH (Locally Likely Arrangement Hashing), which uses local arrangement of word centers [11]. Uchiyama and Saito extended the framework of LLAH for more wide range tracking [15]. LLAH is applied to annotation extraction written in a physical documents [9] and extended to a framework for augmented reality [15].

Since LLAH can achieve real-time processing thanks to hashing scheme [11, 15], we develop the system based on LLAH as described in Section 4.

3. PROPOSED SYSTEM

The configuration of the system is only a camera mounted handheld display such as a tablet PC or a mobile phone. The user prepares text documents in which the user wants to write some annotations electronically. No other special equipment is used.

At the registration of the document into our system, the users capture the documents from nearly top view as shown in Figure 1. While our system shows the captured document on the display, the user can write annotations on the document through the display. We prepare two modes; text and highlighter. In the text mode, the users can write down several sentences at specified positions on the document as shown in Figure 1(a). This mode works as memos and can be replaced with handwriting. In the highlighter mode, the users can highlight the text on the document as shown in Figure 1(b). Since the highlighted areas are semi-transparent, this mode can be considered as a virtual color highlighter pen.

After the registration, the retrieval stage starts. When the same document is captured again, the annotations are overlaid at the specified position. While rotating and translating a camera, the users can watch the overlaid annotations as written on the document. Since many documents can be registered in our system, our system can identify which document is captured now and overlay each annotation.

The operations for registering and deleting documents are by user's click. First, our system starts the capturing process. If the users register a document, the users click a button. Then, our system switches to the registration stage and waits for user's annotation input. After the input, the users click the button again to switch to the retrieval stage. During the retrieval stage, the users can watch the annotations on the document captured. When the users delete a document from the database, the users click the button while watching the annotations of the document. This operation is designed in order to avoid registering the same document. By using these user interactions, the users can register and delete documents.

Our system is designed for the people who do not want to write annotations on the documents directly and can be also considered as an electronical bookmarking. In the previous related works, the document database was prepared from the digital documents [5, 11, 15]. Since it is difficult to prepare the digital version of books and newspapers, our system can be easier and more practical in terms of the usage because our system uses physical documents the user has on hand.

4. DETAILS

4.1 LLAH

LLAH is a document image retrieval method [11]. Since our system relies on it, we briefly describe the method here for completeness.

First, the center of each word is extracted from the captured document image as a keypoint. The image is blurred by using a Gaussian filter and binarized by using adaptive thresholding as shown in Figure 2. Since the filter size of both processing affects our result, we will discuss their effects in Section 5.2.

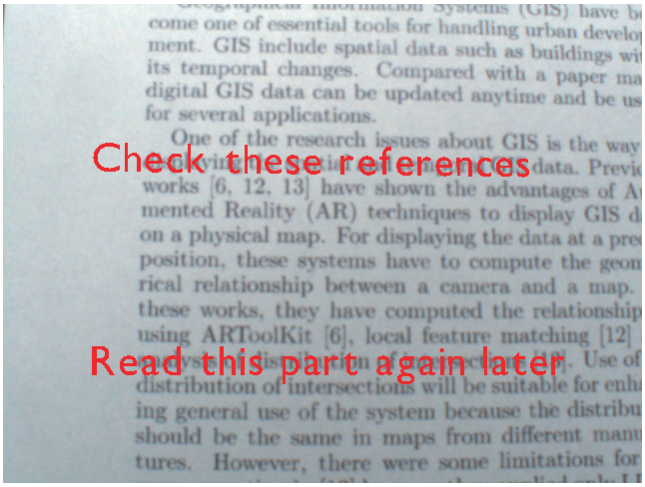
Next, descriptors are computed for each keypoint. In Figure 3, x is a target keypoint. First, n nearest points of the target are selected as abcdefg ($n = 7$). Next, m points out of the n points are selected as abcde ($m = 5$). From m points, a descriptor is computed as explained in the next paragraph. Since the number of the selections is ${}_nC_m = \frac{n!}{m!(n-m)!}$, one keypoint has ${}_nC_m$ descriptors.

From m points, 4 points are selected as abcd. From 4 points, we compute the ratio of two triangles. Since the number of the selections is ${}_mC_4$, the dimension of the descriptor is ${}_mC_4$.

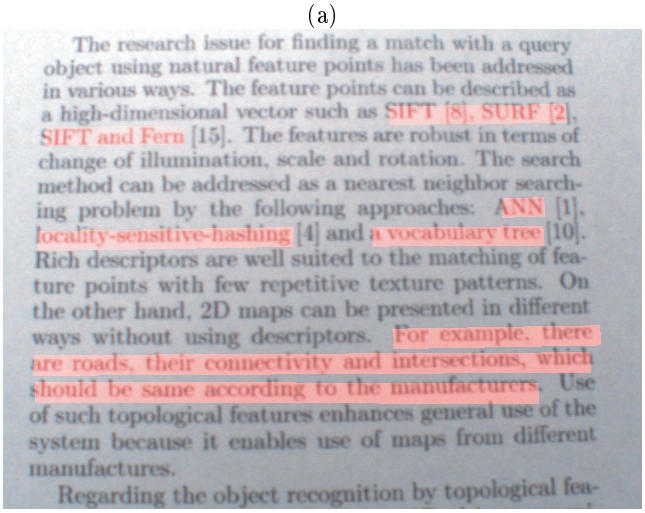
For quick retrieval in keypoint matching, the descriptor is transformed into an index by using the following hash function:

$$Index = \left(\sum_{i=0}^{{}_mC_4-1} r_{(i)} k^i \right) \bmod H_{size} \qquad (1)$$

where $r_{(i)} (i = 0, 1, ..., {}_mC_4 - 1)$ is a quantized ratio of two triangles, k is quantization level and H_{size} is the hash size.



(a)



(b)

Figure 1: Annotation Overlay. (a) Red text is written as a memo. (b) Semi-transparent rectangle is highlighted as written by a color marker pen.

These descriptors allow matching keypoints of an input image with those of a reference image in the database.

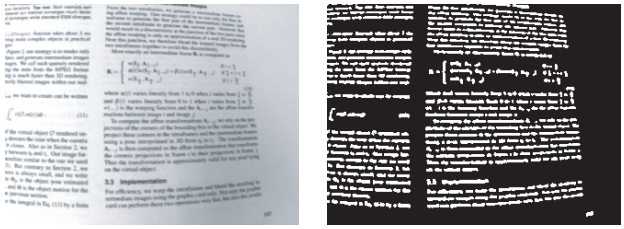
4.2 Document registration

When the user captures a document, our system extracts its keypoints and computes their descriptors. For each document, our system stores keypoints in a table as follows:

Document ID	Keypoint ID	(x, y)	Descriptors

The document ID is numbered by captured order. The keypoint ID is also numbered by extracted order from the image. (x, y) is the coordinate in the image. This allows our system to estimate the geometrical relationship between the coordinate system of the stored image and the one of the input image, making possible accurate annotation overlay. Previous method do not store descriptors [11, 15]. In contrast, we need to keep them for the deletion process described in Section 4.4.

For document retrieval, our system has a descriptor database as follows:



(a) (b)

Figure 2: Keypoint extraction. (a) The document is captured from nearly top viewpoint. (b) The white regions represent the extracted word regions. The keypoint is the center of each region.

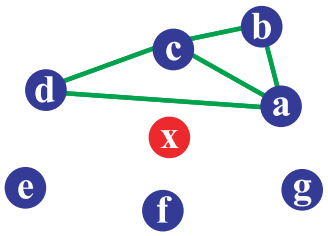


Figure 3: Descriptor. (1) Selection of n points. (2) Selection of m points out of n. (3) Selectiong of 4 points out of m (4) Computation of the two triangles area ratio.

Descriptor	(Document ID + Keypoint ID), ...

As described in Section 4.1, the descriptor is an index. At each index, the set of document ID and keypoint ID is stored. In our system, we use 16 bits for document ID and 16 bits for keypoint ID, and store them as 32 bits integer. Since the same descriptor can be computed, we use a list structure for storing several sets of document ID and keypoint ID at each index.

The descriptor database was generated as a hash table in previous works [11, 15]. If the database can be generated beforehand as in [11, 15], the hash size can be optimized and optimally designed by using all document images. Since our system starts from an empty database, it is difficult to determine the appropriate hash size. For avoiding large empty spaces in the hash table, we use a sparse tree structure for the descriptor database. Even though the computational cost of a binary tree for searching will be $O(\log_2 N)$ compared with $O(1)$ of a hash table, it is enough for our purpose, as discussed in Section 5.3.

4.3 Document retrieval

In this process, keypoints are extracted, and their descriptors and indices are computed as described in Section 4.1. For each keypoint, the several sets of document ID and keypoint ID are retrieved from the descriptor database. If the retrieval is relatively succeeded, the same set of document ID and keypoint ID often appears for a keypoint. By selecting maximum number of the counted sets, one set (document ID and keypoint ID) is assigned to a keypoint.

After assigning one set to each keypoint, we count assigned document ID of each keypoint in order to determine the document image captured currently. The document captured is also identified by selecting maximum number of the

counts.

For verifying that the selected document is correct, we compute geometrical constraints such as fundamental matrix and homography matrix. Since the paper is put on the table, we can use RANSAC based homography computation for the verification [15].

From the computed homography, we can overlay some AR annotations at specified positions on the document. The document retrieval and annotation overlay can be simultaneously done in the same process.

4.4 Document deletion

As described in Section 3, the users can delete the document while watching the annotation of the document. This means that the users delete the current retrieved document.

When the document is deleted, its document data such as the sets of document ID and keypoint ID and their descriptors should be deleted. First, we delete the sets of document ID and keypoint ID from the descriptor database. Since we keep descriptors (indices) for each keypoint in the registration, we can delete the sets by accessing each index. After deleting the sets, we delete other document data.

5. EXPERIMENTS

5.1 Setting

The parameters in LLAH affect the performance and accuracy of document image retrieval. Since the influence of the parameters has already been discussed in [11], we do not discuss it here and fix the parameters through our experiments. Instead, we will discuss about the way of capturing a document and the processing time for our purpose.

In LLAH, the parameters are described in Section 4.1 as follows: n, m, k and H_{size}. Since we set $n = 6$ and $m = 5$, the number of descriptors for one keypoint is ${}_6C_5 = 6$. The quantization level is $k = 32$ and the hash size is $H_{size} = 2^{23} - 1$. As described in Section 4.2, the hash size is used only for computing descriptors. Each descriptor is stored in a binary tree structure. The quantization method of descriptors is the same as [11].

In our current implementation, we use a laptop with a fire wire camera for a device. The laptop has Intel Core 2 Duo 2.2 GHz and 3GB RAM. The size of the input image is 640×480 pixels, and the size for the keypoint extraction is 320×240 pixels for fast computation. The focal length of the lens is fixed as 6mm.

5.2 Image capture

In LLAH, the keypoint extraction is composed of smoothing by a Gaussian filter and binarization by adaptive thresholding. The filter size of both methods needs to be determined beforehand.

Since the filter size affects the result of keypoint extraction, we have tested the keypoint extraction to images captured from different position as shown in Figure 4. The Gaussian filter is 3×3 and the filter for adaptive thresholding is 11×11. The character size is 10 pt written in a A4 paper with two column format.

If the camera is close to the document as 3cm, the each character is individually extracted as shown in Figure 4(a). On the other hand, the word region cannot be extracted from the image captured far from the document (20cm) as shown in Figure 4(b). The word regions are desirably extracted in case of Figure 4(c).

The result of keypoint extraction can be influenced by image size, character size in a physical document, distance between a camera and a document, two filters' size and a lens. These parameters should be optimized by considering the use of the application. In our application, examples of a captured image are as shown in Figure 1. The user captures a A4 paper with 10 pt size's character from around 10 cm high.

5.3 Processing time

We have measured the processing time with 200 small parts of documents. The size of each small part is as shown in Figure 1. In this region, the average number of keypoints was around 180.

The average processing time of each process is shown in Table 1. The document registration without user's annotation took 1 msec. The document deletion also took 1 msec. From these result, user interactions can be done with no stress.

Regarding the document retrieval including the annotation overlay, the average time was 30 msec. Compared with the previous related work [15], the computational cost was reduced because the number of keypoints in a smaller image was fewer. Even though we use a tree structure for searching, we can still achieve about 30 fps and enough processing time for AR.

Table 1: Processing time

Process	msec
Registration	1
Retrieval	30
Deletion	1

6. CONCLUSIONS AND FUTURE WORKS

In this paper, we presented an on-line AR annotation system on text documents. The user can register text documents with annotations virtually written on the document. Then, the user can watch the annotations by AR while capturing the same document again. Our system provides the user interaction for registering and deleting documents. The algorithm of our system is based on LLAH. Our system stores keypoints with their descriptors in the captured image. By using LLAH, our system can quickly identify which document is captured and overlay its annotations. In the experiments, we showed that our system could work real-time.

In our current system, target documents are European documents such as English and French. As a future work, we will apply to any language by changing the keypoint extraction method depending on the language [10]. Also, multiple documents may be detected for showing many annotations simultaneously. For handling a large scale change, keypoint extraction on a pyramid image may be another direction.

7. ACKNOWLEDGMENT

This work is supported in part by a Grant-in-Aid for the GCOE for high-Level Global Cooperation for Leading-Edge

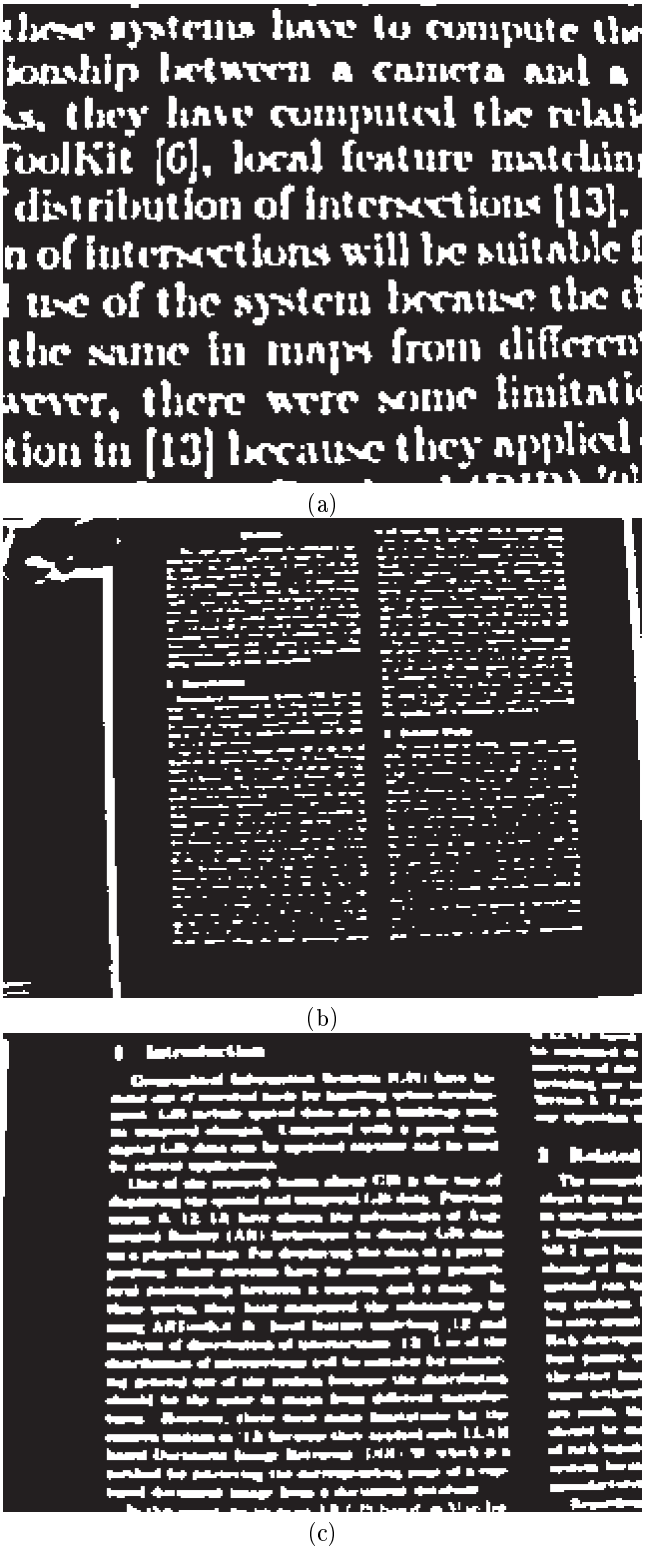


Figure 4: Keypoint extraction at a distance. (a) The camera is set near the document(3cm). (b) The camera is set far from the document (20cm). (c) The distance is between (a) and (b) (10cm).

Platform on Access Spaces from the Ministry of Education, Culture, Sport, Science, and Technology in Japan and Grant-in-Aid for JSPS Fellows.

8. REFERENCES

[1] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. J. of the ACM, 45:891–923, 1998.

[2] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. SURF: Speeded up robust features. CVIU, 110:346–359, 2008.

[3] M. Datar, P. Indyk, N. Immorlica, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In Proc. SCG, pages 253–262, 2004.

[4] C. Harris and M. Stephens. A combined corner and edge detector. In Proc. AVC, pages 147–151, 1988.

[5] J. Hull, B. Erol, J. Graham, Q. Ke, H. Kishi, J. Moraleda, and D. Van Olst. Paper-based augmented reality. In Proc. ICAT, pages 205–209, 2007.

[6] H. Kato and M. Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In Proc. IWAR, 1999.

[7] V. Lepetit, J. Pilet, and P. Fua. Point matching as a classification problem for fast and robust object pose estimation. In Proc. CVPR, pages 244–250, 2004.

[8] D. G. Lowe. Distinctive image features from scale-invariant keypoints. IJCV, 60:91–110, 2004.

[9] T. Nakai, K. Iwata, and K. Kise. Accuracy improvement and objective evaluation of annotation extraction from printed documents. In Proc. DAS, pages 329–336, 2008.

[10] T. Nakai, K. Iwata, and K. Kise. Real-time retrieval for images of documents in various languages using a web camera. In Proc. ICDAR, pages 146–150, 2009.

[11] T. Nakai, K. Kise, and K. Iwata. Camera based document image retrieval with more time and memory efficient LLAH. In Proc. CBDAR, pages 21–28, 2007.

[12] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In Proc. CVPR, pages 2161–2168, 2006.

[13] E. Rosten and T. Drummond. Machine learning for high speed corner detection. In Proc. ECCV, pages 430–443, 2006.

[14] S. Sinha, J. Frahm, M. Pollefeys, and Y. Genc. GPU-based video feature tracking and matching. In Proc. EDGE, 2006.

[15] H. Uchiyama and H. Saito. Augmenting text document by on-line learning of local arrangement of keypoints. In Proc. ISMAR, pages 95–98, 2009.

[16] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg. Pose tracking from natural features on mobile phones. In Proc. ISMAR, pages 125–134, 2008.