Augmenting Text Document by On-Line Learning of Local Arrangement of Keypoints

Hideaki Uchiyama* Hideo Saito†

Keio University

Abstract

We propose a technique for text document tracking over a large range of viewpoints. Since the popular SIFT or SURF descriptors typically fail on such documents, our method considers instead local arrangement of keypoints. We extends Locally Likely Arrangement Hashing (LLAH), which is limited to fronto-parallel images: We handle a large range of viewpoints by learning the behavior of keypoint patterns when the camera viewpoint changes. Our method starts tracking a document from a nearly frontal view. Then, it undergoes motion, and new configurations of keypoints appear. The database is incrementally updated to reflect these new observations, allowing the system to detect the document under the new viewpoint. We demonstrate the performance and robustness of our method by comparing it with the original LLAH.

Keywords: LLAH, on-line learning, pose estimation, paper registration, paper based augmented reality

Index Terms: K.5.1 [INFORMATION INTERFACES AND PRESENTATION]: Multimedia Information Systems—Artificial, augmented, and virtual realities; I.4.8 [IMAGE PROCESSING AND COMPUTER VISION]: Scene Analysis—Tracking

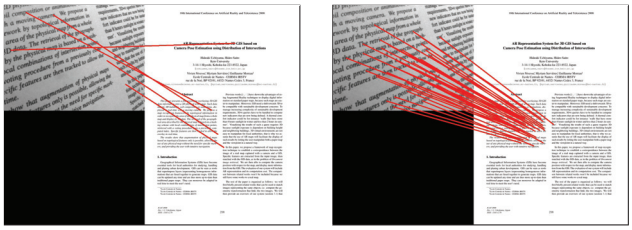
1 Introduction

A popular category of AR application uses physical paper documents. It is called *Paper-Based Augmented Reality* [5]. In this application, a user reads a document printed and can click words and logos using a mobile device displays for additional digital information. This enables document papers to be media for making links between physical and digital worlds. Paper based AR has much more potential possibilities for providing various kinds of services to the users.

For developing paper based AR applications, one of the most important issues is registration. Typical approaches to registration include feature point matching with descriptors such as SIFT [8] and SURF [2]. However, these methods fail to handle poorly textured objects or repetitive binary patterns such as text. Figure 1(a) represents a result of keypoint matching by SURF in a document. Even though 4419 keypoints were extracted in the query image, only 5 matches were established, including some outliers.

To overcome this limitation, Nakai et al. have proposed a method called Locally Likely Arrangement Hashing (LLAH) that uses the center of a word as a keypoint and computes descriptors from local arrangement of keypoints [10, 11]. It is utilized to retrieve the corresponding document of a query image from the document database. In LLAH, the movement of the camera is restricted, because viewpoint changes cause local feature patterns to change.

In our work, we propose to overcome this limitation by incrementally learning the new patterns as soon as they appear. In addition, our system learns new keypoints that appear in novel views, depending on their discriminative power. As a result, we are able to reliably and efficiently augment text documents as shown in Figure 1(b). We can apply our method to several 2D planar papers such as newspapers, documents, intersection map [17] for developing paper based AR applications.



(a) (b)

Figure 1: Keypoint matching for a document. (a) Matching by SURF in OpenCV [13]. SURF doesn't work in case of repetitive binary patterns. (b) Matching by our method. Our method works even under strongly tilted views by learning LLAH.

2 Related work

The process of feature point-based pose tracking can be divided into three parts: keypoint extraction, keypoint description and keypoint matching.

Harris corner [4] and FAST corner [14] have been proposed for extracting keypoints which have different appearance from other pixels. For taking into account the scale change, there has been several approaches such as difference of Gaussians [8], gradient location and orientation histogram [9] and basic Hessian-matrix approximation [2].

A descriptor is a high dimensional vector computed with local neighbor region of the keypoint in order to have a discriminative power. The well-known descriptors such as SIFT [8] and SURF [2] are well designed to be invariant to illumination change and viewpoint change. SIFT has been implemented on a GPU in order to use parallel processing and speed up the computational time [15]. Wagner et al. have modified SIFT to apply to a mobile phone within the limited resources such as memory and cache [18].

Matching of descriptors can be addressed as a nearest neighbor searching problem between high dimensional vectors. Approximate Nearest Neighbor (ANN) [1] and LSH (Locality Sensitive Hashing) [3] are well known methods in approximated nearest neighbor searching. Nister and Stewenius have proposed a recursive k-means tree in order to make a vocabulary tree [12]. Lepetit et al. have treated matching of descriptors as a classification problem [7].

Local descriptors such as SIFT and SURF are well suited to match keypoints with rich texture patterns, but they are not dicriminative enough to work on text documents. Instead, geometrical relationships of keypoints have been proposed for such documents. [5, 11].

*e-mail: uchiyama@hvrl.ics.keio.ac.jp

†e-mail: saito@hvrl.ics.keio.ac.jp

Hull et al. have proposed horizontal connectivity of word lengths as a descriptor [5]. The descriptor is valid only when a user captures an image parallel and close to the paper and words parallel to the lower side of the image. Nakai et al. have proposed point retrieval method using local patterns of keypoints for document image retrieval, which is called LLAH [11]. LLAH is an improved method of Geometric Hashing (GH) [6] in order to reduce computational cost and required amount of memory. In their system, the camera should be set almost parallel to and away from a paper for capturing as much part of the paper as possible.

In our previous work, we have applied LLAH to map image retrieval for a GIS data representation system [16] and proposed LLAH tracking for improving the constraint of a moving camera [17]. Keypoint extraction on a map was very simple and stable because the center of a colored intersection marker was a keypoint. Since the keypoint extraction in a document is much more unstable, some further improvements to LLAH tracking are necessary.

In the next section we explain the details of LLAH because we extend LLAH for flexible camera movement.

3 Original LLAH [10]

In LLAH, a descriptor is composed of local arrangement of nearest neighbor points. For a keypoint, several descriptors are computed using different contributions of nearest neighbor points. In order to retrieve the corresponding keypoint from the keypoint database quickly, each descriptor is indexed in a hash scheme.

The procedure of computing a descriptor and indexing is as follows: For a target keypoint, n neighbor keypoints are selected in a defined clockwise. Next, $m(with m < n)$ keypoints are selected from the n keypoints. Since a descriptor is computed from m keypoints, each keypoint has ${}_nC_m = \frac{n!}{m!(n-m)!}$ descriptors. For each one of the possible combinations of 4 points picked among the m keypoints, the ratio fo 2 adjoining triangles (defined by the 4 points) is computed as an affine invariant. As a result, a descriptor is composed of the ${}_mC_4$ ratios. The descriptor is converted into an index by using following equation:

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

where $r_{(i)}(i = 0, 1, \ldots, {}_mC_4 - 1)$ is a quantized value of an affine invariant, k is the quantization level and H_{size} is the hash size.

During an off-line pre-processing stage, each keypoint extracted from images in a database is uniquely numbered and stored as (Keypoint ID, x, y). For each keypoint, ${}_nC_m$ indices are computed and stored as (Index, Keypoint ID).

In the on-line retrieval processing stage, keypoints are extracted and their indices are computed in a query image as well as the off-line. For the indices, a histogram over keypoint IDs is computed, and the keypoint ID is taken as the mode of the histogram.

4 On-line update

4.1 On-line update of hash table

In the original LLAH, the update of the hash table and the keypoint database is not considered. In our proposed approach, we can incrementally achieve more flexible camera movement by learning new patterns and keypoints in the LLAH framework.

In the on-line retrieval process, keypoints extracted in a query image can be divided into two classes; retrieved (matched) keypoints and unretrieved (unmatched) keypoints. From the retrieved keypoints, the corresponding image of a query image is retrieved from the image database if there are enough retrieved keypoints. In addition, the homography between the query image and the retrieved image is computed since many 2D-2D correspondences have been established. By using this homography, the keypoints in the database can be reprojected onto the query image. From this reprojection, unretrieved keypoints might be matched with the keypoints reprojected from the database when the distance between the unretrieved keypoint and the reprojected keypoint is close. As a result, some of the unretrieved keypoints by LLAH become matched keypoints which can get a keypoint ID of the reprojected keypoint. For all retrieved (matched) keypoints, update of the hash table is performed.

Each retrieved (matched) keypoint in a query image has a histogram of keypoint ID as shown in Figure 2. In the histogram, there is the counted number of NULL which means the indices do not include a keypoint ID. By occupying NULL with the keypoint ID, the hash table is updated in order to enable keypoint retrieval by LLAH at the next frame or later. This process works effectively for incremental tracking.

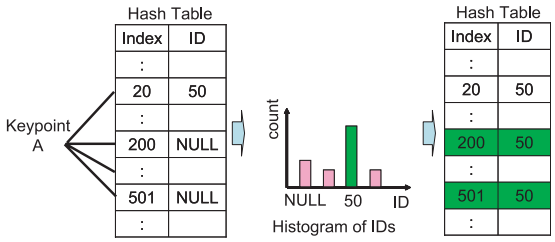


Figure 2: On-line update of hash table. Regarding keypoint A, several IDs are retrieved. If the ID is selected as No.50, No.50 is inserted at NULL.

4.2 On-line update of new discriminative keypoints at keyframes

There are unretrieved (unmatched) keypoints after the process of updating the hash table in Section 4.1. Since these keypoints could not be matched by the reprojection of the database, these keypoints might not be stored in the database beforehand. If these keypoints are stored in the database during the on-line processing, these keypoints can be useful at the next frame or later. We select new keypoints which have discriminative power compared with keypoints in the database.

These keypoints also have the histogram of keypoint ID. The discriminative power of each keypoint can be described by the number of indices of NULL. A keypoint does not have discriminative power when the histogram has already been full by other keypoints. For this reason, a keypoint will be discarded when the number of indices of NULL is less than a threshold.

When an unmatched keypoint is selected as a new keypoint, its x and y in the coordinate system of the database is computed by using the homography. Its new keypoint ID and the x and y coordinate are stored in the database as shown in Figure 3. Also, the new keypoint ID is inserted at the indices of NULL in the hash table.

This process is performed at keyframes. If the number of retrieved keypoints at a frame is enough, new keypoints do not have to be added. For this reason, we decide if a frame becomes a keyframe by thresholding the number of retrieved points by LLAH.

5 Details

5.1 Overview

The flowchart of our process is illustrated in Figure 4. As a preprocessing, we compute keypoints extracted from document images and their indices for making an initial keypoint database and a hash table as well as the original LLAH [10].

Our initialization is equivalent to the process of document image retrieval in LLAH. After the initialization is successfully done,

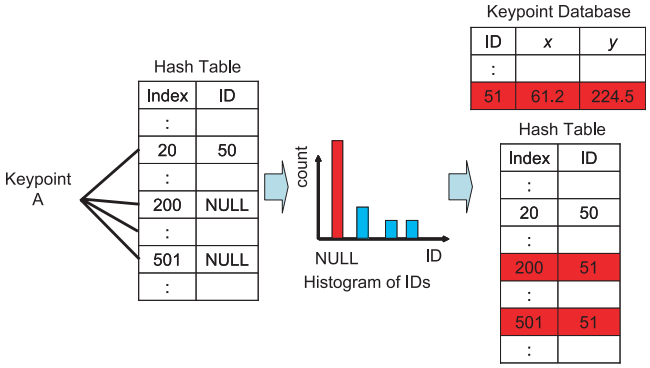


Figure 3: On-line update of new discriminative keypoints. Since keypoint A is new based on the number of NULL, new ID (No.51) and its xy are stored in the database. Also, No.51 is inserted at the indices of NULL.

we update hash table of keypoints and add new keypoints into the keypoint database.

In the initialization and pose estimation, keypoint extraction and selection of nearest neighbor points are performed. Since we improved these processes compared with the original LLAH, we describe the details from the next section.

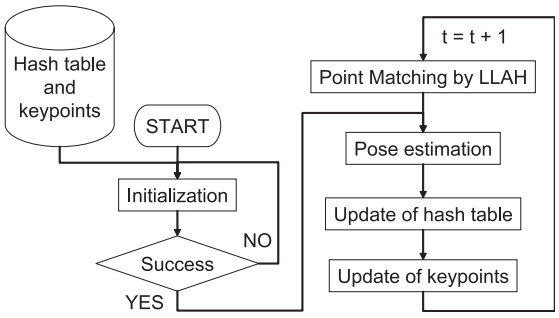


Figure 4: Process flow. On-line update of keypoints and the hash table starts after the initialization is finished.

5.2 Keypoint extraction

In our method, a keypoint is the center of a word as was given in [11]. For extracting keypoints, a captured image is blurred by using a Gaussian filter for removing noise. Next, the blurred image is adaptively thresholded to make a binary image. Finally, the dilation of white regions is performed once in order to make a word being one region by connecting the letters.

A camera might be close to a document compared with [11] because we do not restrict the movable range of a camera. In that case, each word may be clearly captured and cannot be one region by dilation once. For this reason, we change the number of the dilation times depending on the distance between a camera and a document (less than 100mm: 3 times, less than 150mm: 2 times, else: once).

5.3 Selection of nearest neighbor points

After keypoint extraction, nearest neighbor points of each keypoint are selected for computing the descriptors. In case that the distances with all keypoints are computed, the computational cost will be $O(N^2)$ where N is the number of keypoints. For searching neighbor points efficiently, we perform searching candidates of neighbor points within limited neighbor areas.

As a pre-process, a query image is divided into square regions by segmenting at an uniform interval as shown in Figure 5. Each region includes several keypoints after the square region of each extracted keypoint is computed. When we search nearest neighbor keypoints of a keypoint, we collect their candidates from the surrounding regions of the region to which the keypoint belongs. If the number of candidates is not enough, we collect the candidates from larger surrounding regions.

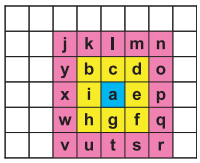


Figure 5: Selection of nearest neighbor points. If a keypoint belongs to **a**, the candidates are extracted from the region **b** to the region **i**. If the number of the candidates is not enough, more candidates are extracted from the region **j** to the region **y**.

5.4 Pose estimation

2D-2D correspondences between a query and the database are established by LLAH. Even though reliable IDs might be selected from the histogram, outliers are sometimes included. For estimating a camera pose without the influence of outliers, RANSAC based homography computation is performed.

After the homography computation, keypoints in the database can be reprojected onto a query image by using the homography in order to get more correct 2D-2D correspondences. The correct correspondences are established by thresholding the reprojection error between a reprojected keypoint from the database and the nearest extracted keypoint in the query image. We defined that the extracted keypoint was matched with the projected keypoint if the error was less than 2 pixels. From these new correspondences, the homography is computed again for refining pose estimation.

5.5 Update of hash table and keypoint database

After pose estimation, the update of the hash table and keypoint database is performed.

If a viewpoint changes, some keypoints can be retrieved by LLAH. On the other hand, the other keypoints can not be retrieved by LLAH because the local arrangement of neighbor points of a keypoint might be different. By updating descriptors by new patterns, point retrieval by LLAH can work better at the next frame or later. In the process of updating the hash table, we update the descriptors of matched keypoints by projection of keypoints in the database. Since the arrangement of neighbor points of retreived keypoints by LLAH slightly changes, we update the descriptors of the retreived keypoints.

New keypoints might be generated because the center of a word can not be stably extracted even though we change the dilation times of white regions as described in Section 5.2. Also, the center of a word might be shifted by the effect of perspective distortion with change of viewpoint. From these keypoints, discriminative keypoints are selected and added as new keypoints based on the histogram of retrieved keypoint IDs. The update of the keypoint database is performed at keyframes in order to avoid updating too much keypoints. If the number of retrieved keypoints by LLAH is less than a threshold, the update of the keypoint database is performed.

6 Evaluation

6.1 Setting

We implemented our system on a laptop PC with Intel Core 2 Duo 2.2GHz, 3GB RAM and 640 × 480 pixel camera under windows XP. Our system is written in C++ using OpenCV [13]. The intrinsic

parameters of the camera and the radial distortion of the lens are calibrated beforehand. We determined empirically paramters for LLAH described in Section 3 as well as [11]. In this experiment, we will show our robust registration and processing time compared with the original LLAH [11].

6.2 Registration

We extended the original LLAH to enable free camera moving. If update of the hash table and keypoints is removed from our method, the method is equivalent to the original LLAH. We applied our method and the original LLAH to a video stream for the comparison.

In Figure 6, the original LLAH worked only in case of (a). On the other hand, our method worked in all cases by overlaying several rectangles. Even though the occlusion area is quite large, our method still worked in case of (d). Update of the hash table and keypoints strongly contributed to free camera moving and solving the occlsion problem.

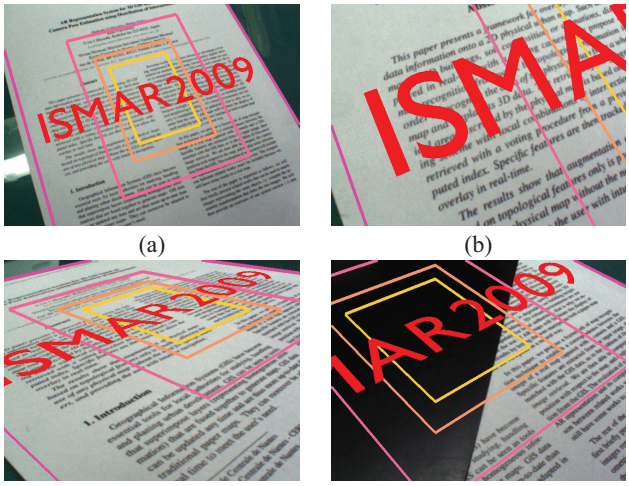


Figure 6: Adding a virtual 'ISMAR 2009' label. Our method can apply to the cases; (a) is far away from paper. (b) is very close to paper. (c) is captured by tilted camera. (d) is a large occlusion.

6.3 Processing time

Some part of our process is the same as the original LLAH. The difference is the update of the hash table and keypoints. In Table 6.3, the average computational cost of each process was computed by using 100 images. The computational costs of the update of the hash table and keypoints were only 4 msec. By adding the only two process, the tracking results were drastically changed as shown in Figure 6.

Table 1: Processing time. The difference between our method and the original LLAH is update of hash table and keypoints. Our method needs only more 4 msec than the original LLAH.

Process	msec
Point matching by LLAH	45
Keypoint extraction	23
Pose estimation	2
Update of hash table	2
Update of keypoints	2

7 Conclusion

We propose a method for text document tracking based on on-line update of geometrical relationship of local keypoints. We improved LLAH for free camera moving with additional few computational costs. Our method learns the behavior of keypoint patterns with change of viewpoints. In addition, new keypoints are updated depending on their discriminative power in order to overcome unstable keypoint extraction. As a result, our method can apply to several camera positions such as far away or close to the paper and tilted. Our method could still work even if most part of the paper was occluded in the experiment. Our improvement needs few computatinal cost and provides much more contributions for removing the constraint of the camera movement in LLAH.

Acknowledgements

We thank Dr. Julien Pilet for the discussion. This work has been supported by "Foundation of Technology Supporting the Creation of Digital Media Contents" project (CREST, JST), Japan.

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] Y. Lamdan and H. Wolfson. Geometric hashing: A general and efficient model-based recognition scheme. In *Proc. ICCV*, pages 238–249, 1988.

[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] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *PAMI*, 27:1615–1630, 2005.

[10] T. Nakai, K. Kise, and M. Iwamura. Use of affine invariants in locally likely arrangement hashing for camera-based document image retrieval. In *Proc. DAS*, pages 541–552, 2006.

[11] T. Nakai, K. Kise, and M. Iwamura. 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] OpenCV. http://sourceforge.net/projects/opencvlibrary/.

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

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

[16] H. Uchiyama, H. Saito, M. Servières, and G. Moreau. AR representation system for 3D GIS based on camera pose estimation using distribution of intersections. In *Proc. ICAT*, pages 218–225, 2008.

[17] H. Uchiyama, H. Saito, M. Servières, and G. Moreau. AR GIS on a physical map based on map image retrieval using LLAH tracking. In *Proc. MVA*, pages 382–385, 2009.

[18] 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.