Memory Reduction for Real-Time Document Image Retrieval with a 20 Million Pages Database

Kazutaka Takeda, Koichi Kise and Masakazu Iwamura
Dept. of CSIS, Graduate School of Engineering
Osaka Prefecture University
1-1 Gakuen-cho, Naka, Sakai, Osaka, 599-8531 Japan
takeda@m.cs.osakafu-u.ac.jp, {kise, masa}@cs.osakafu-u.ac.jp

***Abstract*—We have introduced the three improvements of Locally Likely Arrangement Hashing (LLAH) in ICDAR2011 to reduce a required amount of memory and increase discrimination power of features. In this paper, we show the experimental results which is obtained on a larger-scale database than that utilized for ICDAR2011. From experimental results, we have confirmed that the proposed method realizes 60% memory reduction and achieves 99.2% accuracy with 49ms/query processing time for the retrieval of a database of 20 million pages.**

***Keywords*-Document image retrieval, Real-time 20 million pages processing, LLAH, Large-scale database**

I. Introduction

Recently, portable devices called smartphones are attracting more attention all over the world. Smartphones combine the functions of a camera phone and personal digital assistant (PDA). They have advanced network capability compared to the traditional mobile phones. Owing to this function, we can utilize various information retrieval services, including "Google Goggles" [1], Kooaba [2] and Snaptell [3]. For example, "Google Goggles" returns related search results of the objects in the picture captured by a smartphone camera. When users capture a book cover, users can obtain the related information of the book. With the helpful of these services, users can easily obtain beneficial information about the objects. These applications are based on specific object recognition [4] [5] [6]. In this paper, we especially focus on document image retrieval which limits specific object recognition to printed documents.

Document image retrieval is a task to find from a database a document image corresponding to a query obtained by capturing a document. For this purpose, various methods have been proposed [7] [8]. However, these methods lack usability because they intend scanned documents. This paper concerns document image retrieval with camera captured documents as queries. This technique can easily provide users with the information associated with the retrieved document in the database. In other words, with the help of camera-based document image retrieval, paper documents can be viewed as media for accessing various information; images, movies, texts and more.

As the method to deal with camera-based document image retrieval, Locally Likely Arrangement Hashing (LLAH) has been proposed [9]. In this method, centroids of word regions are utilized as feature points. LLAH retrieves document images based on features which consist of geometric invariant. Note that features are calculated based on the arrangement of feature points. It is also known for its fast retrieval which enables a real-time processing. Moreover, LLAH has already been extended for retrieval of documents in various languages [10]. Owing to the above property, LLAH has been applied to Augmented Reality [11] and a Camera-Pen system [12].

If LLAH can realize real-time document image retrieval with a large scale database, which includes more than 10 million pages, it becomes more useful. However, to scale up the database, LLAH has a problem that a large amount of memory is required. In order to realize accurate retrieval on a 20 million pages database, 350GB memory is needed. Such heavy consumption of memory limits the scalability of LLAH. Moreover, as the database becomes large, retrieval accuracy decreases because similar features are more likely to be extracted from a larger database. Therefore, we need to increase the discrimination power of features. To address these problems, we have proposed three improvements of LLAH in ICDAR2011 [13]; the memory reduction by sampling feature points stored in the database, to improve discrimination power of features by increasing the number of dimension of features and to improve stability of features by reducing redundancy of features.

In this paper, we show experimental results with a larger-scale database than a database of 10 million pages used for ICDAR2011. From experimental results with a 20 million pages database, we have confirmed that the improvement of memory reduction is effective but the other two improvements are not necessarily effective for the 20 million pages database. In addition, by using the proposed method, we have realized demo system with 1 million pages database which runs on a laptop with 8GB of memory. We confirmed that this demo system allows us to retrieve a corresponding image at a rate of 15 frames per second (fps).

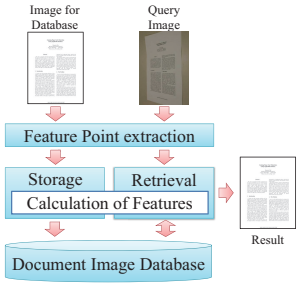

Figure 1. Overview of processing.

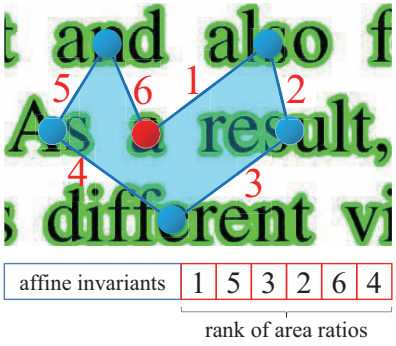

Figure 2. Additional feature.

II. Document Image Retrieval with the Original LLAH

A. Overview of processing

Figure 1 shows the overview of processing of LLAH. First, feature points are extracted from a document image. Then, features are calculated from arrangements of the feature points. In the storage step, every feature point in the image is stored into the document image database using its feature. In the retrieval step, the document image database is accessed with features to retrieve images by voting. We explain each step in the following.

B. Feature point extraction

An important requirement for the feature point extraction is that feature points should be obtained identically even under perspective distortion. To satisfy this requirement, we employ centroids of word regions as feature points.

The process is as follows. First, the input image is adaptively thresholded into a binary image. Next, the binary image is blurred using the Gaussian filter. The blurred image is adaptively thresholded again. Finally, centroids of word regions are extracted as feature points.

C. Calculation of features

A feature is a key to retrieve correspond point from the database. The feature of LLAH is defined for each feature point so as to realize robustness and availability under occlusion. In addition, the geometric invariants are employed as a feature. This is because the geometric invariant is stable under perspective distortion which occurs in camera-captured images. To be more precise, we utilize the affine invariant defined with four coplanar points ABCD as follows:

$$\frac{P(A,C,D)}{P(A,B,C)} \tag{1}$$

where $P(A,B,C)$ is the area of a triangle with apexes A, B and C.

In order to increase the discrimination power of a feature, a feature consists of multiple affine invariants calculated from multiple feature points. An arrangement of m neighboring points is described as a sequence of discretized invariants $(r_{(0)},\cdots,r_{(\binom{m}{4}-1)})$ calculated from all possible combinations of 4 feature points taken from m feature points. To improve stability of features, $\binom{n}{m}$ features are calculated for each point from all possible combinations of m points from n points. In the original LLAH, $n=7, m=6$ are utilized.

The rank of area ratios of word regions is also employed as an additional feature. In Fig. 2, the edges represent area ratios. For example, the edge 5 indicates the area ratio of the word region of "As" to that of "and". The additional feature is concatenated to the affine invariants. As a result, the number of dimension of the feature is $(\binom{m}{4}+m)$.

D. Storage

Every feature point is stored in the database in accordance with its feature. The index H_{index} of the hash table is calculated by the following hash function:

$$\left(\sum_{i=0}^{(\binom{m}{4}+m)} r_{(i)}d^i\right) = QH_{\text{size}} + H_{\text{index}} \tag{2}$$

where $r_{(i)}$ is a discrete value of the invariant, d is the level of quantization of the invariant, and H_{size} is the size of the hash table. Q is uniquely determined for a feature. Therefore, the item (document ID, point ID and Q) is stored in the hash table where chaining is employed for collision resolution. Consequently, we can make sure that each item of the list has the same feature by using Q in place of the feature in retrieval.

E. Retrieval

In LLAH, the result of retrieval is determined by voting on documents represented as cells in the voting table.

H_{index} and Q are calculated for each feature point of a query image in the same way as in the storage step. The list of items (document ID, point ID and Q) is obtained by looking up the hash table. For each item, a cell of the corresponding document ID in the voting table is incremented if it has the same Q. Eventually, the document obtained the maximum votes is returned as the retrieval result.

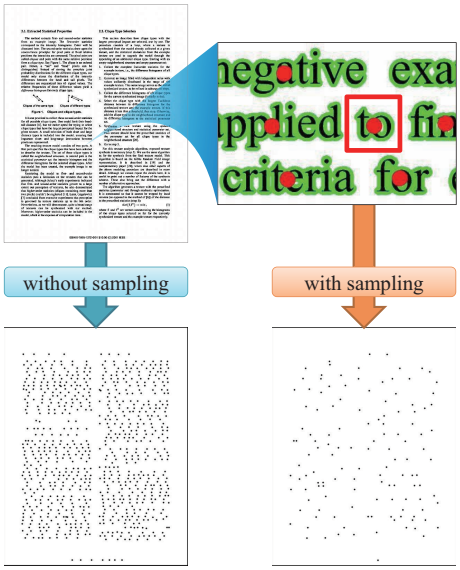

Figure 3. Example of word region selected to sample feature point.

III. Proposed Method

We explain the proposed method which is introduced in ICDAR2011.

A. Reduction of required amount of memory

In the original LLAH, about 600 feature points per page are extracted and 7 features per feature point are calculated. Therefore, about 4200 features per page are stored in the database. When we make 20 million pages database, the required amount of memory becomes large because about 84 billion features are stored in the database. However, there is no need to store all points since retrieval results are determined by voting. Therefore, we reduce memory consumption by sampling feature points stored in the database.

In sampling feature points, the important thing is to evenly select feature points in a page. If the distribution of sampled points is uneven, the page has both densely sampled and sparsely sampled regions. In the case that queries are captured at sparse regions, LLAH cannot realize accurate retrieval since the number of feature points is not enough.

To meet the requirement stated above, we focus on areas of word regions in order to sample feature points. In particular, we sample feature points extracted from word regions whose areas are smaller than those of surrounds. Figure 3 (top right) shows an example of the word regions selected to sample feature points. In this example, the area of "to" is smaller than those of surrounds. Therefore, the feature point extracted from "to" is selected. Figure 3 (bottom) shows a result of sampling. This represents a result with sampling and without sampling. From this example, it is confirmed that the proposed method selected feature points evenly. Moreover, when the area of the word region is smallest, it is more robust against perspective distortion since the distances to the n nearest points become shorter.

However, the number of sampled points in this way is too small. Thus we store k nearest points from each sampled point. In the propose method, k is determined so that the number of feature points in a document is 200. If the number of all extracted feature points is less than 200, we take all points.

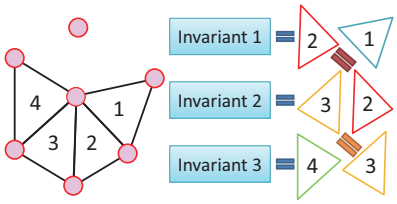

Figure 4. Invariants and their redundancy.

B. Improvement of the feature

In general, as the size of database becomes large, retrieval accuracy decreases. This also holds for LLAH. When the number of pages stored in the database increases, it is more likely that pages have similar arrangements of feature points. As a result, a number of erroneous votes occur in the retrieval process. Therefore, we need to modify features so that they can be more discriminative to address similar arrangement of feature points.

In order to increase the discrimination power, we increase the number of dimensions of features. In this paper, we increase the number of dimensions by modifying the values of parameters n and m. The feature of original LLAH has 21 dimensions when the parameters are $n = 7$, $m = 6$. In the proposed method, we increase the dimensions by modifying the values to $n = 8$, $m = 7$. As a consequence, the number of affine invariants is $\binom{7}{4} = 35$ and the number of area ratio features is 7, resulting in the total number of dimensions 42.

Although the discrimination power is enhanced by increasing the number of dimensions, the stability of features decreases. LLAH requires that all dimensions of features must be identical to determine the corresponding point in retrieval. However, an error occurs in the affine invariant due to position variations of feature points. For this reason, when the number of dimensions increases, there is an increasing possibility that different invariants are calculated. As a result, the stability of features decreases in exchange for increasing the discrimination power.

To solve this problem, we reduce the redundant dimensions of features. As shown in Fig. 4, we utilize the same triangle to calculate two different invariants. Triangle 2 is utilized to calculate the invariants 1 and 2. Moreover, triangle 3 is utilized to calculate the invariants 2 and 3. Therefore, the invariant 2 has a correlation with invariants 1 and 3. This redundancy can be resolved by deleting the invariant

Figure 5. Example of images in the database.

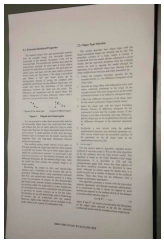

Figure 6. Example of query images.

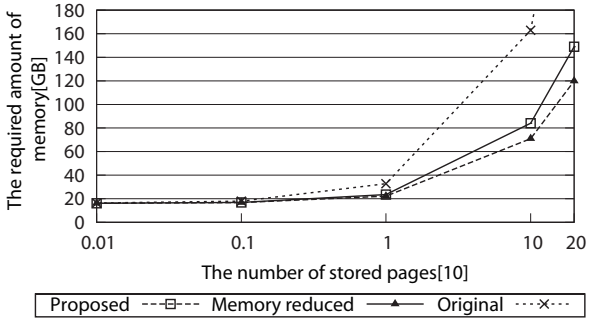

Figure 7. Relationship between the number of stored pages and the required amount of memory.

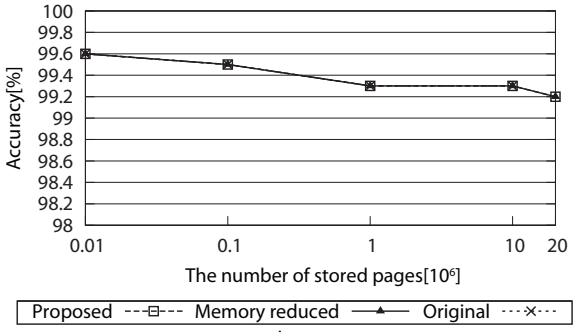

Figure 8. Relationship between the number of stored pages and accuracy.

2. As shown in the above example, we select triangles for calculating invariants in the way that no invariants share triangles. As a result, the number of dimensions becomes 24.

IV. Experiments

We experimented with the proposed method on a 10 million pages database in ICDAR2011. In this paper, we utilized the same method and experimented with a database of 20 million pages. We investigated the required amount of memory, processing time per query and retrieval accuracy of three versions of LLAH; the original LLAH, the memory reduced LLAH without increasing the dimensionality and the proposed LLAH.

For the experiments, we made five databases which include different numbers of pages: 0.01 million, 0.1 million, 1 million, 10 million and 20 million. By testing with the databases which included the various numbers of pages, we examined scalability of each method. Document images stored in the database were images converted with 200 dpi from PDF files collected mainly from the Internet. An example of images in the database is shown in Fig. 5. Query images were captured from an elevation angle of 60 degrees using a digital camera with 1,200 million pixels. The number of query images was 1,003, whose example is shown in Fig. 6. Since the angle with which query images are captured (60 degrees) is different from that of the images in the database (90 degrees), experiments performed with these query images demonstrate robustness of the proposed method to perspective distortion. Experiments were performed on a workstation with AMD Opteron 2.2GHz CPUs and 256GB memory. The required amount of memory represented all memory consumption needed in retrieval. H_{size} was set to $2^{31}-1$. Note that the original LLAH cannot deal with a 20 million pages database.

A. Required amount of memory

Figure 7 shows the relationship between the number of pages stored in the database and required amount of memory. These values include the size of hash table (16GB). The estimated value of original LLAH with a 20 million pages database is 350GB.

With the 10 million pages database, the memory reduced version of LLAH achieved about 60% memory reduction compared to the original LLAH. On the other hand, memory consumption of the proposed version is larger than that of the memory reduced version. The number of features per feature point stored in the database is $\binom{8}{7} = 8$ in the proposed method, while $\binom{7}{6} = 7$ in the memory reduced method. Therefore, the proposed method requires larger memory. In memory reduced version, 20 million pages database required about 120GB.

B. Accuracy

Figure 8 shows the relationship between the number of pages stored in the database and retrieval accuracy. Retrieval accuracy became same values from 0.01 million to 10 million for the all version of LLAH. Especially, the Memory reduced LLAH and the proposed LLAH have the same accuracy, which is 99.2%, on the 20 million pages database. This means that the original number of dimensions of features has enough discrimination power to deal with the 20 million pages database.

Figure 9 shows an example of query images which caused a failure. This query image consists of few text regions and many figures. Since the current feature point extraction utilizes centroids of word regions, it does not work well with such images. In order to address this problem, we need to implement a new feature point extraction process.

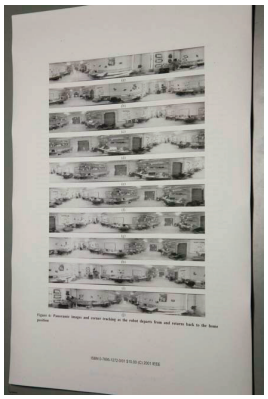

Figure 9. Example of images which caused retrieval errors.

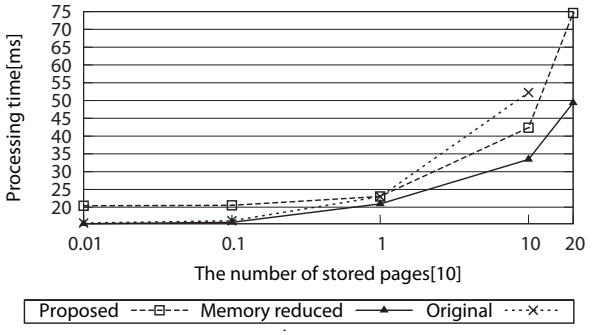

Figure 10. Relationship between the number of stored pages and processing time per query.

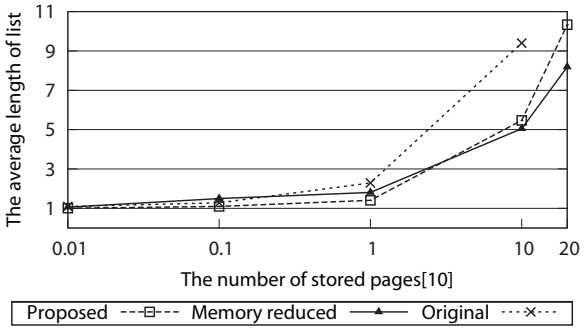

Figure 11. Relationship between the number of stored pages and the average length of lists.

C. Processing time

Figure 10 shows the relationship between the number of pages stored in the database and processing time. As the number of pages increased, processing time became longer for the all versions of LLAH. This is because the length of lists in the hash table increased as the database was scaled up. Figure 11 shows the average length of lists whose length is not zero. For example, with the memory reduced LLAH, the average length of lists of a 20 million pages database is about 8 times as long as that of a 0.01 million pages database. Since the number of accessing lists increased when the length of lists was long, processing time became long. Note that the reason why the length of list explodes on the 20 million pages database is that the hash table is occupied by a lot of entries. In the proposed method, more than 73% hash bins were occupied. Therefore, when we make a larger-scale database, we should employ a larger hash table.

The memory reduced LLAH allows faster processing than the proposed method by about 25ms. On a 20 million pages database, the average length of lists of the memory reduced LLAH is shorter than that of the proposed method. Therefore, the memory reduced LLAH has the small number of accessing lists. In addition, the computational complexity of both versions of LLAH to calculate features are different as shown in IV-A. From these reasons, the memory reduced version has faster processing time than the proposed method.

The memory reduced version of LLAH realized 49ms/query processing time on a 20 million pages database. Therefore, it can retrieve document images in real-time.

V. Real-time demo system using LLAH

Based on LLAH, we have built a real-time demo system of document image retrieval running on a laptop. A user can retrieve document images by capturing printed documents with a web camera. By using the proposed method, this system has achieved 15 frames per second (fps) with the 1 million pages database on a laptop with 8GB of memory. The retrieval results are almost always correct under severe occlusion and perspective distortion as well as non-linear deformation of the surface of pages.

Figure 12 illustrates the appearance of the real-time demo system. From the result of retrieval, we obtain the feature point correspondences between the query image and the corresponding image in the database. An example is shown in Fig. 12 with straight lines between images. From the correspondence, we can estimate the parameters of perspective transformation. By employing RANSAC for estimating the parameters, we increase the precision of the parameters. The parameters are utilized to estimate the surface angle of the captured page as well as the captured region of the corresponding document.

Let us explain that the demo system is realized as a client-server system. Figure 13 shows the flow of the processing. The method simply repeats the cycle of image capture, feature point extraction, retrieval and display a result. The client system repeats the following five processes in order.

1) Capture the query image with a web camera.
2) Extract feature points from the query image.
3) Receive the retrieval result and the parameters.
4) Send the extracted feature points to the server system.
5) Display the result.

The server system repeats the following four processes in order.

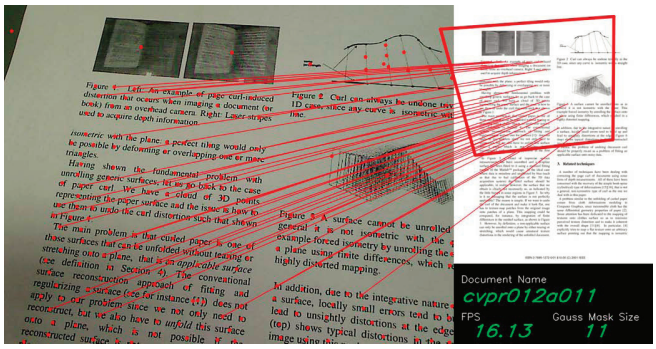

Figure 12. Appearance of the real-time demo system.

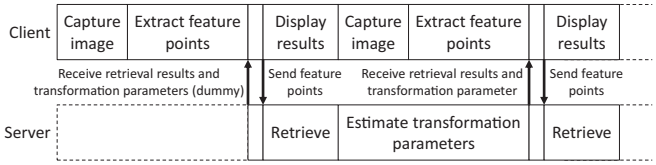

Figure 13. Pipeline processing with a client-server system.

1) Send the retrieval result and the parameters.
2) Receive feature points.
3) Retrieve the corresponding page.
4) Estimate the parameters based on the point correspondence.

As shown in Fig. 13, the parallel processing of the client and server is employed. The parallel processing allows us to achieve a high frame rate. In the current implementation, the client requires longer processing time than the server. Therefore, 15fps processing time is identical to that of the client.

We introduce applications of the real-time demo system. The one of the applications is augmented reality (AR) to documents. The AR is to display relevant information as an image overlapped with a captured query image. The image representing the relevant information is perspectively transformed using the parameters estimated during the retrieval. We currently consider as relevant information annotations by text, images, underlines and handwriting. The other is to obtain words and text in the captured region without character recognition. We can get the position of words from the original PDF of the captured document. Therefore, the user can search the meaning of words included in the captured region.

VI. Conclusion

We have proposed three improvements of the original LLAH in order to reduce the memory consumption and increase the discrimination power of features. In this paper, we experimented with the same method for a larger-scale database. From the experimental results, we have confirmed that the memory reduction was effective for the 20 million pages database. However, we could not confirm the effectiveness of the method of increasing discrimination power.

Our future task includes improvement of feature point extraction process. In addition, further reduction of memory requirement for making a large library available for document image retrieval.

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://www.google.com/mobile/goggles/.

[2] http://www.kooaba.com/.

[3] http://www.snaptell.com/.

[4] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *Proc. CVPR2006*, 2006, pp. 775–781.

[5] M. Özuysal, M. Calonder, V. Lepetit, and P. Fua, "Fast keypoint recognition using random ferns," *IEEE transactions on pattern analysis and machine intelligence*, pp. 448–461, 2009.

[6] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," 2003.

[7] D. Doermann, "The indexing and retrieval of document images: A survey," *Computer Vision and Image Understanding: CVIU*, vol. 70, no. 3, pp. 287–298, jun 1998.

[8] D. Doermann, H. Li, and O. Kia, "The detection of duplicates in document image databases," *Proc. 4th International Conference on Document Analysis and Recognition (ICDAR1997)*, pp. 314–318, 1997.

[9] 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, feb 2006.

[10] ——, "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, jul 2009.

[11] R. T. Azuma, "A survey of augmented reality," *Presence*, vol. 6, no. 4, pp. 355–385, 1997.

[12] K. Kise, M. Chikano, K. Iwata, M. Iwamura, S. Uchida, and S. Omachi, "Expansion of queries and databases for improving the retrieval accuracy of document portions — an application to a camera-pen system," *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems (DAS2010)*, pp. 309–316, jun 2010.

[13] 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)[to appear]*, 2011.