Hashing with Local Combinations of Feature Points and Its Application to Camera-Based Document Image Retrieval

— Retrieval in 0.14 Second from 10,000 Pages —

Tomohiro Nakai, Koichi Kise, Masakazu Iwamura
Graduate School of Engineering, Osaka Prefecture University
1-1 Gakuen-cho, Sakai, Osaka, 599-8531 Japan
nakai@m.cs.osakafu-u.ac.jp, {kise, masa}@cs.osakafu-u.ac.jp

Abstract

This paper presents a new method of indexing and retrieval of planar objects based on feature points and its application to document image retrieval using cameras. As the indexing method we propose a method based on local combinations of projective invariants calculated from feature points. As the retrieval method we employ a voting technique for efficiency and robustness against erasure of feature points. Experimental results on 10,000 images with 50 queries show that the method is effective (98% accuracy; the remaining query was ranked at the 5th position among 10,000) and efficient (0.14 second per query).

1. Introduction

Document image retrieval is a task of searching document images relevant to a user's query. For meeting diverse needs from users, a wide variety of queries have been employed [1]. With document images as queries, the task of finding similar or equivalent document images has been considered. For scanned documents it is called "document image matching" or "duplicate detection" [2, 3]. This paper concerns a kind of document image matching with camera captured documents as queries. We call this task "camera-based document image retrieval".

In order to deal with camera captured images, various kind of problems including perspective distortion, uneven lighting and focusing should be solved [4, 5]. We are concerned here with the problem of perspective distortion. An ordinary way of dealing with the distortion is to normalize the image by estimating parameters of projective transformation. In this paper we employ a different approach to this problem with the help of *invariants* and *hashing*.

In the field of computer vision, a method called geometric hashing [6] is well-known as an effective way of indexing and retrieval of images. In geometric hashing, images are represented as a collection of points, and images in the database or *models* are indexed with invariants calculated from their points. The voting technique is employed for distinguishing models based on a query image. It is difficult, however, to apply geometric hashing to camera-based document image retrieval since ordinary geometric hashing can deal with similarity or affine transformation; it cannot handle perspective distortion in an efficient way.

To solve this problem, this paper presents a new method of indexing and retrieval for images of planar objects using techniques of hashing and voting for feature points of images. As the feature points we utilize centroids of word regions. The main contribution of this paper is the proposal of a new hash key that is effective and efficient even under perspective distortion as well as erasure of some feature points. A projective invariant called "cross-ratio" is employed for the robustness to perspective distortion. The hash key is defined based on *local combinations* of feature points. The locality allows us to make the method insensitive to point erasure. The discriminability of the hash key is boosted by combining the feature points. From the experimental results on 10,000 document images, it is shown that the method can achieve almost perfect retrieval (only 1 of 50 queries is missed) within a short period of time (0.14 second per query in average).

2. Proposed method

2.1. Fundamental ideas

There are some problems to be solved for achieving camera-based document image retrieval: images captured by cameras can be projectively transformed, images may not include whole text regions, and resolution and illumination of images may be different from those in the database. Basic ideas for solving these problems are as follows:

(1) Hash based indexing and retrieval as voting

In order to make retrieval computationally feasible, we employ hashing and voting for documents. In the proposed method, a document image with the largest number of votes is selected as the result.

(2) Invariant-based hash key

In order to make the hash keys of document images projectively invariant, we calculate them using cross-ratios. As feature points from which cross-ratios are calculated, centroids of word regions are utilized, since they are robust to projective transformation and noises.

(3) Local combinations of feature points

In order to make the hash key insensitive to point erasure as well as to improve its discriminability, we locally combine feature points.

2.2. Cross-Ratio

The cross-ratio is known as an invariant of projective transformation. It is calculated using coordinates of five coplanar points on an image. For five points ABCDE, the cross-ratio is calculated as

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

where P(A,B,C) is the area of a triangle with apexes A, B, and C [7]. Since the cross-ratio is a projective invariant, its value keeps unchanged even if coordinates of points ABCDE change by perspective distortion.

Although the values of cross-ratios obtained from feature points are continuous, they must be converted to k discrete values in order to be used as indices. Values should be discretized by taking into account their frequency: the discretization step should be finer for values occurring more frequently. In the proposed method, discrete values are assigned in proportion to the frequency of values of cross-ratios using a histogram of values of cross-ratios obtained in a preliminary experiment.

2.3. Overview of processing

Figure 1 shows the overview of processing. At the step of feature point extraction, a document image is transformed into a set of feature points. Then feature points are inputted into the registration step or the retrieval step. These steps share the step of calculation of indices. In the registration step, every feature point in the image is registered into the document image database using its index. In the retrieval step, the document image database is accessed with indices to retrieve images by voting. We explain each step in the following.

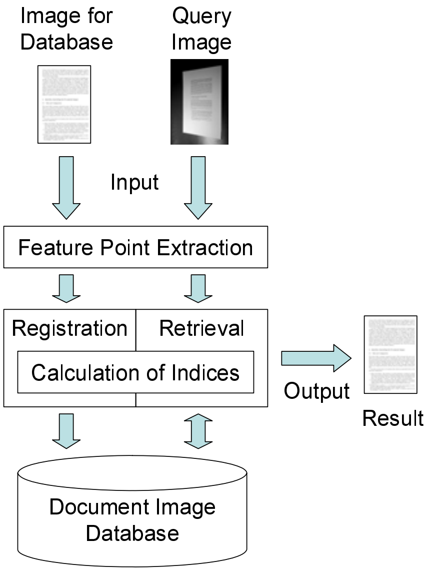
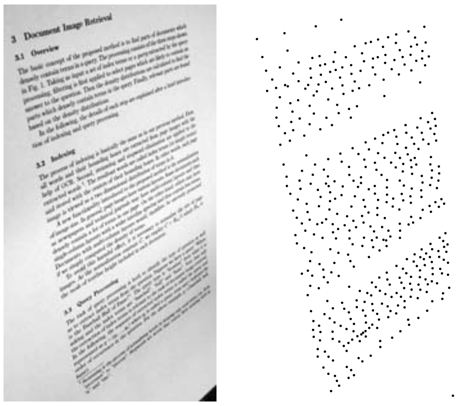


Figure 1. Overview of processing.



(a) input image (b) feature points

Figure 2. Feature point extraction.

2.4. Feature point extraction

Feature points should be obtained identically even under the perspective distortion, noise and low resolution. We employ centroids of word regions as feature points because they almost satisfy this requirement. First, input images (Fig. 2(a)) are adaptively thresholded into binary images. Next, the binary images are blurred using the Gaussian filter whose parameters are determined based on an estimated character size (the square root of a mode value of areas of connected components). Then, the blurred images are adaptively thresholded again. Finally, centroids of word regions (Fig. 2(b)) are extracted as feature points.

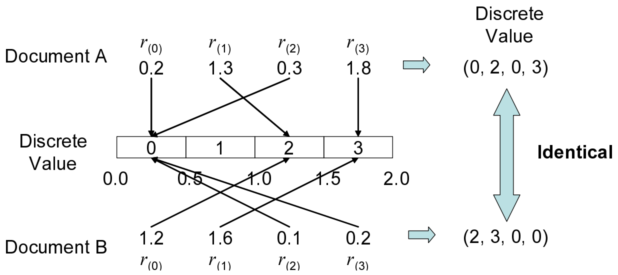


Figure 3. Discriminability of cross-ratios.

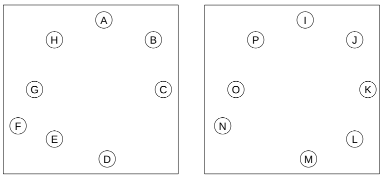


Figure 4. n points.

2.5. Calculation of indices

In the proposed method, each feature point is characterized by cross-ratios. Although it seems reasonable to calculate a cross-ratio for each feature point based on its five nearest feature points, it is not appropriate since in general the nearest points vary due to the projective distortion.

Another important problem is the discriminability of cross-ratios as illustrated in Fig. 3 that represents a case with quantization level $k = 4$. Suppose we have cross-ratios $r_{(0)},\cdots,r_{(3)}$ for documents A and B. Although the real values are different, their discrete versions are identical. Moreover, it is impossible to distinguish the documents A and B by counting the votes for each discrete value (0:twice, 2 and 3: once). Although the discriminability could be improved by increasing the level of quantization k, it sacrifices the robustness to noise.

In the proposed method, we attempt to solve the first problem using local combinations of feature points. The index of a feature point is calculated not just from the five nearest points but from the n nearest points. It is often the case that $m(< n)$ points in the n points are kept unchanged under ordinary perspective distortion.

Let us explain in more details with Fig. 4 which represents the $n(= 8)$ nearest feature points for a feature point in a document image and those for the corresponding feature point in a query image. In this figure, $m(= 7)$ points ABCDFGH and IJKMNOP are common. Thus the common combination of feature points can be obtained by examining all possible ${}_nC_m$ combinations. From the same combination of m points, the common cross-ratios are obtained by combining all possible ${}_mC_5$ points for calculating cross-ratios from points such as ABCDF and IJKMN, ABCDG and IJKMO.

The second problem of discriminability is solved by taking into account the order of cross-ratios. In the case of Fig. 3, the cross-ratios are different if we consider them as the sequences (0,2,0,3) and (2,3,0,0). Note that if a feature point in a database image corresponds to that in a query image, the sequence should be identical. Consider again the case in Fig. 4. A sequence of cross-ratios are calculated for every m points. Let a series of letters such as ABCDF represent the cross-ratio defined by these points. If the points correspond with each other, the sequence of cross-ratios from m points (ABCDF, ABCDG, ABCDH, BCDFG, BCDFH, ...) and its corresponding sequence (IJKMN, IJKMO, IJKMP, JKMNO, JKMNP, ...) become identical.

The following is the summary of calculation of indices. For each feature point, its n nearest points are obtained. Then all possible ${}_nC_m$ combinations of m points are generated from n points. Indices are defined as ordered cross-ratios by taking ${}_mC_5$ combinations from m points in the fixed order.

```
for each $p \in$ {All feature points in a database image} do
    $P_n \leftarrow$ The nearest $n$ points of $p$ (clockwise)
    for each $P_m \in$ { All $m$ points combinations from $P_n$} do
        for each $P_5 \in$ { All 5 points combinations from $P_m$} do
            $r_{(i)} \leftarrow$ The cross-ratio calculated with $P_5$
        end for
        $H_{\text{index}} \leftarrow$ The hash index calculated by Eq. (2).
        Register the item (document ID, point ID, $r_{(0)},\cdots,r_{({}_mC_5-1)}$) using $H_{\text{index}}$
    end for
end for
```

Figure 5. Registration algorithm.

2.6. Registration

Let us turn to the registration step. Figure 5 shows the algorithm of registration of document images to the database. In this algorithm, the document ID is the identification number of a document, and the point ID is that of a point.

Next, the index of the hash table H_{index} is calculated by the following hash function:

$$H_{\text{index}} = \left(\sum_{i=0}^{{}_mC_5-1} r_{(i)} k^i \right) \bmod H_{\text{size}} \qquad (2)$$

where $r_{(i)}$ is the discrete value of the cross-ratio, k is the level of quantization of cross-ratios and H_{size} is the size of the hash table.

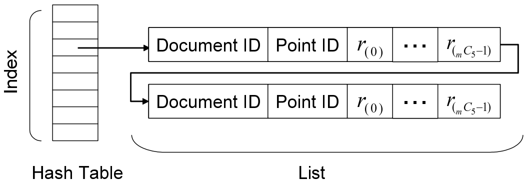


Figure 6. Configuration of the hash table.

```
 1: for each $p \in$ {All feature points in a query image} do
 2:   $P_n \leftarrow$ The nearest $n$ points of $p$ (clockwise)
 3:   for each $P_m \in$ { All $m$ points combinations from $P_n$} do
 4:     for each $P'_m \in$ { Cyclic permutations of $P_m$} do
 5:       for each $P_5 \in$ { All 5 points combinations from $P'_m$}
          do
 6:         $r_{(i)} \leftarrow$ The cross-ratio calculated with $P_5$
 7:       end for
 8:       $H_{\rm index} \leftarrow$ The hash index calculated by Eq. (2).
 9:       Look up the hash table using $H_{\rm index}$ and obtain the
          list.
10:       for each Item of the list do
11:         if Conditions 1 to 3 are satisfied then
12:           Vote for the document ID in the voting table.
13:         end if
14:       end for
15:     end for
16:   end for
17: end for
18: Calculate the score based on the votes.
19: Return the document image with the maximum score.
```

Figure 7. Retrieval algorithm.

The item (document ID, point ID, $r_{(0)}, \cdots, r_{({}_mC_5-1)}$) is registered into the hash table as shown in Fig. 6 where chaining is employed to collision resolution.

2.7. Retrieval

The retrieval algorithm is shown in Fig. 7. In the proposed method, retrieval results are determined by voting on documents represented as cells in the voting table.

First, the hash index is calculated at the lines 5 to 8 in the same way as in the registration step. At the line 9, the list shown in Fig. 6 is obtained by looking up the hash table. For each item of the list, a cell of the corresponding document ID in the voting table is incremented if the following conditions are satisfied.

Condition 1: All values of $r_{(0)} \cdots r_{({}_mC_5-1)}$ in the item are equal to those calculated at the lines 5 to 7 for P'_m.

Condition 2: It is the first time to vote for the document ID with the point p.

Condition 3: It is the first time to vote for the point ID of the document ID.

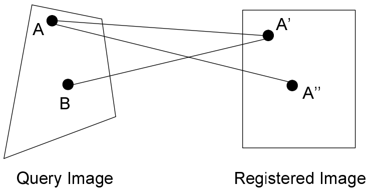


Figure 8. Incorrect correspondence.

The condition 1 aims to remove items with different sequences of cross-ratios. Note that a sequence of cross-ratios $r_{(0)} \cdots r_{({}_mC_5-1)}$ is not necessarily identical for items with the same value of the hash function $H_{\rm index}$.

The conditions 2 and 3 aim to limit votes caused by inconsistent correspondences. In the algorithm in Fig. 7 voting is to seek points which correspond to the point p. If only the condition 1 is employed, we face the following two types of inconsistency shown in Fig. 8: (Type 1) A point (A) in the query image corresponds to more than one point (A' and A") in a registered image. (Type 2) A point (A') in a registered image corresponds to more than one point (A and B) in the query image. In order to avoid such inconsistent correspondences, the conditions 2 and 3, which are for the types 1 and 2, respectively, are employed.

After repeating these steps for every point, the voting table with votes on every registered document is obtained. In spite of the above conditions 2 and 3, votes caused by incorrect point correspondences are generally obtained. The number of such incorrect votes is approximately in proportion to the number of feature points in a registered image. Hence registered images with a larger number of feature points tend to have unfairly larger votes. In order to compensate for the number of unfair votes, the following score $S(d_i)$ for a document d_i is calculated based on the numbers of votes $V(d_i)$ and feature points $N(d_i)$:

$$S(d_i) = V(d_i) - p_n \cdot N(d_i) \qquad (3)$$

where p_n is the proportionality constant of the number of feature points to those of incorrect votes, which is determined by a preliminary experiment. Finally, the document with the maximum score is determined as the result.

3. Experimental results

3.1. Overview

In order to examine effectiveness of the proposed method, we measured accuracy and processing time. Query images were captured from a skew angle using the digital camera CANON EOS Kiss Digital (also known as EOS-300D; 6.3 million pixels) with EF-S 18-55mm USM. The

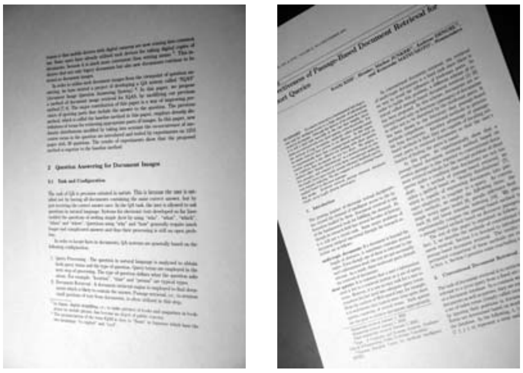


Figure 9. Examples of query images.

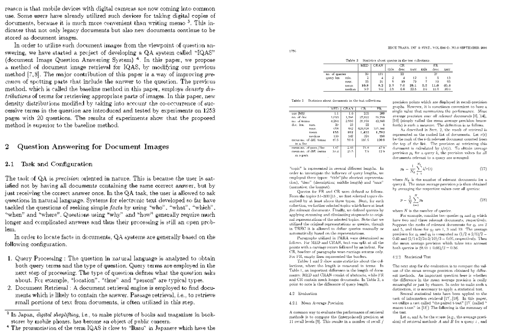


Figure 10. Examples of images in database.

Table 1. Contents of the database.

Title	Registered pages
CVPR 2001	1630
CVPR 1999	1211
ICCV 1999	1170
IDCAR 1997	609
ICPR 2002	2426
ICPR 2004	2724
IWFHR 2004	65
IEEE Transactions on Multimedia 1999	144
Others	21

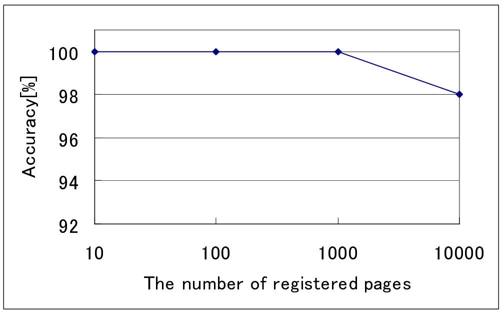


Figure 11. Accuracy of retrieval.

number of query images was 50. Figure 9 shows examples of query images whose size is 2,048 × 3,072. As documents in the database we employed 10,000 page images converted with 200 dpi from PDF files of single- and double-column English papers collected from CD-ROM proceedings shown in Table 1. Figure 10 shows examples of images in the database whose size is about 1,700 × 2,200. Note that the pages in the database look quite similar because all pages are from scientific papers. Experiments were performed on a workstation with AMD Opteron 1.8GHz CPUs and 4GB memory. Parameters described in Sect. 2 were set to $n = 8, m = 7, k = 10, H_{\mathrm{size}} = 128M, p_n = 0.022$.

3.2. Accuracy of retrieval

We first analyzed the relationship between the size of the database (the number of registered pages) and the accuracy of retrieval (the rate that the correct page receives the maximum score). The results are shown in Fig. 11: the accuracy of 100% was obtained for the sizes of 10 to 1,000 pages, and 98% for 10,000 pages. Figure 12 to 14 show some examples of a query image and 1st to 5th ranked images.

The query image that caused failure for the case with 10,000 pages is shown in Fig. 14(a): the correct page was ranked in 5th position. We consider that the reason of failure on this image is its narrow text region; it becomes more difficult to obtain correct correspondences between points if text regions are smaller since the number of feature points is small. Figure 15 shows successful and erroneous cases of point correspondences. As illustrated in Fig. 15(b), narrow text regions limit correct correspondences.

Figure 16 shows the relationship between the number of pages in the database and the average ratio of scores for the first to second ranked pages. As the size of the database grows, the difference between scores for the first and second ranked pages decreases. This is because expansion of the database increases the chance of having similar configuration of points.

3.3. Processing time

Next, we analyzed how the database size affects processing time. Figure 17 shows the results. The growth of the number of pages accompanies the increase of processing time. This figure also shows the average length of lists in the hash. The average list length is the average number of length of lists with at least one entry. The average list length, which means the number of collisions, increases as the number of registered pages increases. That is the reason

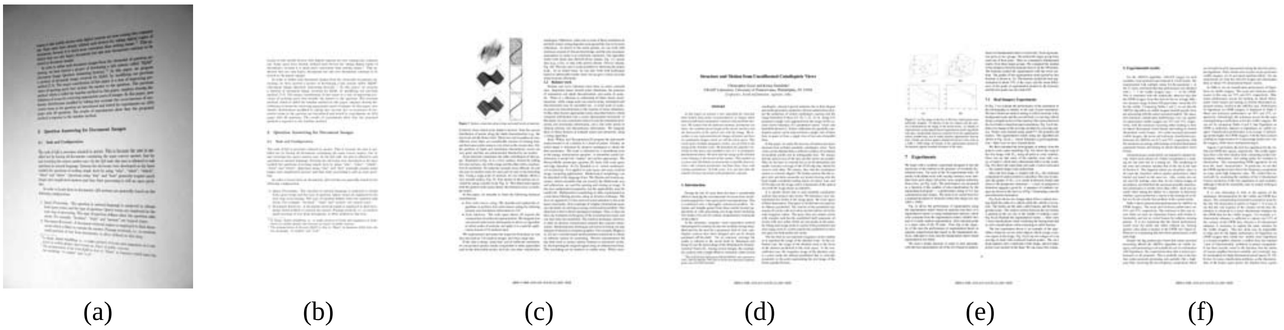


Figure 12. Successful case. (a) query image, (b) 1st, (c) 2nd, (d) 3rd, (e) 4th, and (f) 5th ranked image.

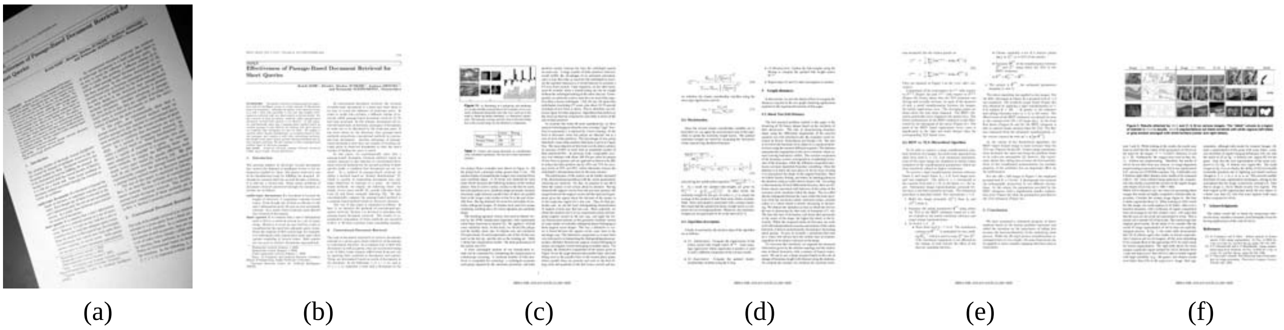


Figure 13. Successful case. (a) query image, (b) 1st, (c) 2nd, (d) 3rd, (e) 4th, and (f) 5th ranked image.

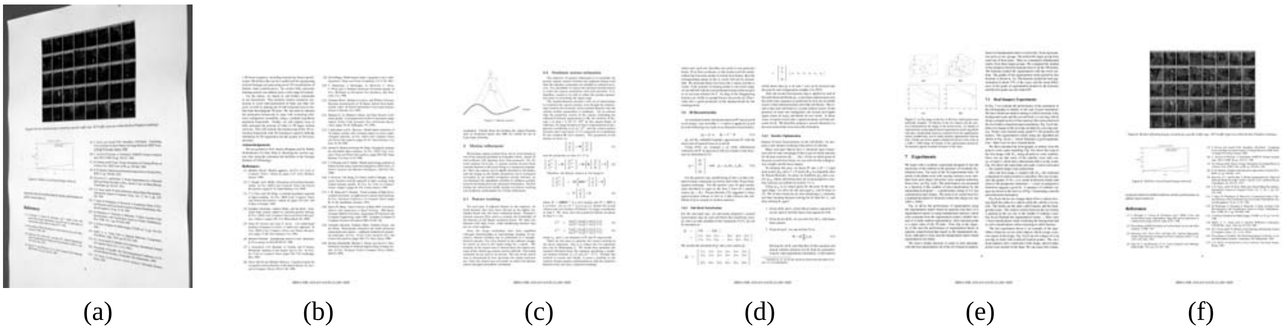


Figure 14. Failure case. (a) query image, (b) 1st, (c) 2nd, (d) 3rd, (e) 4th, and (f) 5th ranked image.

of the increase of processing time.

4. Related work

The proposed method can be said as a method for object recognition since it is for retrieval of the corresponding models to query images from a database. There have been many methods for object recognition which utilize invariants as the proposed method does. In this section, we describe similar methods and differences from them.

4.1. Geometric hashing

As mentioned above, the geometric hashing is a method of object recognition based on invariants. In the geometric hashing, all feature points of models are registered into a hash table using 2 to 4 selected points for defining a local coordinate basis. The number of points for the basis depends on the kind of invariance: 2 for similarity, 3 for

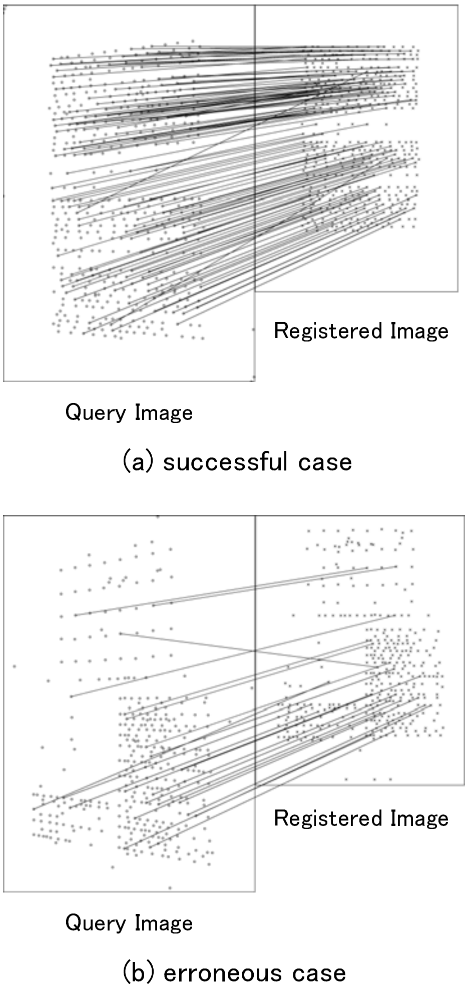


Figure 15. Point correspondences between a query and a database image.

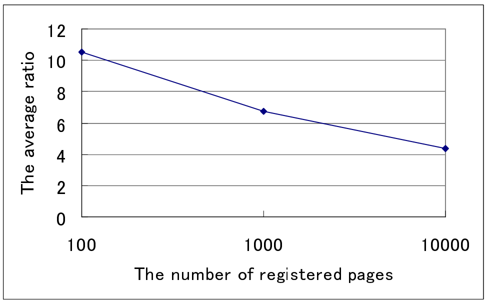


Figure 16. The average ratio of scores for the first to the second ranked pages.

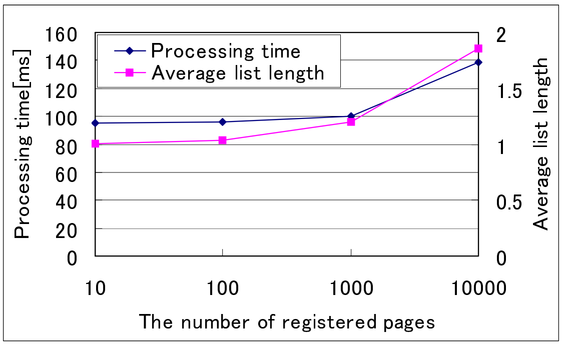


Figure 17. The relationship among the number of registered pages, processing time and the length of lists in the hash table.

affine, and 4 for projective transformation invariance. Registration is performed on every possible basis. Retrieval is performed by looking up from the hash table using an arbitrarily selected basis and voting. The geometric hashing is similar to the proposed method in the following points.

- Invariance for transformation
- Registration of each point
- Utilization of hashing

However, the proposed method is superior to the geometric hashing in terms of computational complexity. In the proposed method, features are calculated from limited neighboring points for each feature point in the registration and the retrieval processes. Hence the computational complexity of the proposed method is O(N) where N is the number of feature points in each model. On the other hand, in the geometric hashing each feature point is registered using every possible basis. Hence the computational complexity of registration is O(N^{b+1}) where b is the number of points for defining a basis. For example, for the case of projective invariants, the computational complexity of geometric hashing is O(N^5) since four points are necessary for the basis.

4.2. Other methods

Many invariant-based object recognition methods such as [8] and [9] have so far been proposed. However, improvement of discriminability by combinating invariants is not employed in these methods. For example, the feature is simply a cross-ratio of five connected line segments in [9]. It is difficult in our case to adopt such a simple indexing, because a huge number of points have similar cross-ratios.

In order to avoid the problem, the sequence of cross-ratios is employed in the proposed method; this high-dimensional feature realizes high accuracy and computational efficiency.

5. Conclusion

We have proposed a method of indexing and retrieval of planar objects based on feature points and its application to camera-based document image retrieval. The method is characterized by the hash key calculated from local combinations of projective invariants. High accuracy and efficiency of the proposed method were shown by the experimental results. Future work includes experiments with more queries and an extension of the method to object retrieval in scene images.

References

[1] D. Doermann: "The Indexing and Retrieval of Document Images: A Survey", Computer Vision and Image Understanding, **70**, 3, pp.287–298 (1998).

[2] J. J. Hull : "Document image matching and retrieval with multiple distortion-invariant descriptors", Document Analysis Systems, pp.379–396 (1995).

[3] D. Doermann, H. Li and O. Kia: "The detection of duplicates in document image databases", Proc. ICDAR'97, pp.314–318 (1997).

[4] D. Doermann, J. Liang and H. Li: "Progress in camera-based document image analysis", Proc. ICDAR'03, pp. 606–616 (2003).

[5] P. Clark and M. Mirmehdi: "Recognising text in real scenes", IJDAR, **4**, pp. 243–257 (2002).

[6] H. J. Wolfson and I. Rigoutsos: "Geometric hashing: an overview", IEEE Computational Science & Engineering, Vol. 4, No. 4, pp.10–21 (1997).

[7] T. Suk and J. Flusser : "Point-based projective invariants", Pattern Recognition 33, pp.251–261 (2000).

[8] B. Huet and E. R. Hancock: "Cartographic indexing into a database of remotely sensed images", WACV96, pp.8–14(1996).

[9] C. A. Rothwell, A. Zisserman, D. A. Fosyth and J. L. Mundy: "Using projective invariants for constant time library indexing in model based vision", Proc. BMVC, pp.62–70(1991).