Efficient Recognition of Planar Objects Based on Hashing of Keypoints — An Approach Towards Making the Physical World Clickable

Koichi Kise, Tomohiro Nakai, Masakazu Iwamura, Satoshi Yokota
Dept. of Computer Science and Intelligent Systems,
Graduate School of Engineering School of Engineering, Osaka Prefecture University
{kise, masa}@cs.osakafu-u.ac.jp, {nakai, yokota}@m.cs.osakafu-u.ac.jp

Abstract

This paper presents a method of planar object recognition for aiming at accessing information about objects by taking pictures of them. For this purpose efficiency of processing is the central issue because current state-of-the-art technologies with tree structures do not necessarily work well with a large amount of data represented as high dimensional vectors. To solve this problem, we employ hashing of keypoints extracted from images of objects. With the help of hash keys obtained as integers converted from the real valued vectors, keypoints are stored with object IDs and retrieved with no search process. Voting for object IDs is employed to determine a recognized object as the one with the largest vote. Experimental results show that the proposed method is at least 400 times faster than a brute-force method while 90% of objects were correctly recognized.

1. Introduction

Suppose you would like to retrieve additional information about a poster you are looking at. Besides typing in URLs or keywords to your mobile terminal, recent information technologies are offering different ways. One way is to use a mobile terminal with character recognition capability. However it is not an easy task to recognize correctly all characters in the 3D environment. Another possibility would be to use RFID tags. In addition to their costs, they have a problem about discriminability. It is problematic if there exist several other objects with RFIDs near the object of interest. Within currently available technologies, the most promising way would be to use barcodes for recording URLs and other information. Their well known shortcoming is that they spoil appearance of objects on which they are printed. For example, if you would like to access information when you look at a catalog, each item should have its barcode that makes the total design of the catalog ruined.

We are concerned here with a completely different way of achieving the above goal using *images* of objects taken by digital cameras under the assumption that objects are planar such as posters and signs. All that have to be done for accessing information is just to take a picture of the object of interest. The proposed method can recognize the object in the picture by matching it against all images stored in the database. Because the user's action of taking pictures seems to click the object of interest by the shutter button, the proposed method is considered to be a step towards making the physical world clickable.

Reseachers in the field of computer vision have tackled the task of planar object recognition. Recent progress in indexing images based on keypoints such as SIFT [5] and PCA-SIFT [3] allows us to consider the clickable physical world. A major obstacle is the burden of computation required for matching keypoints. Although several sophisticated data structure such as ANN [1] have been proposed, they are commonly incompetent with a larger number of high dimensional data that are typical to our case.

This paper presents a method of solving the above problem by using hashing techniques. The use of hashing for finding similar data was first introduced in the field of databases [2] and recently applied to near-duplicate detection [4] of still images. In this paper we apply a different method called "locally likely arrangement hashing (LLAH)" proposed by Nakai et al. [6] for achieving higher discriminability of similar objects under perspective distortions efficiently. From the results of prelminary experiments, it is shown that the recognition time was reduced less than 0.3% of a simple brute-force matching while keeping 90% of the original accuracy.

2. Recognition method

Figure 1 shows a configuration of the system consisting of a database which stores images of objects as well as information associated with them, and a retrieval engine to

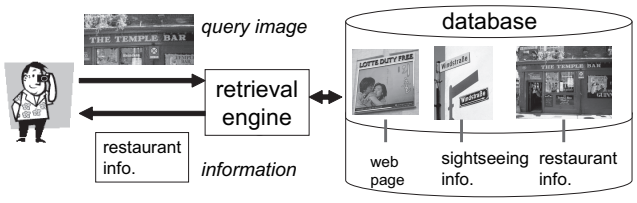

Figure 1. System configuration.

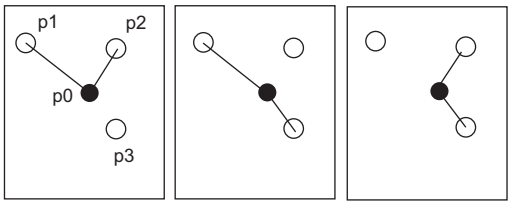

Figure 2. Arrangements of $m(=2)$ points out of $n(=3)$ points.

find an image of the corresponding object in response to a query image given by the user. The key technologies here are (1) how to create the databse with images, and (2) how to retrieve corresponding images.

2.1. Creation of the database

Images of objects are stored into the database using the extracted keypoints. Problems of creation of the database are threefold: (1) how to achieve efficiency to keep the system scalable, (2) how to realize robustness against noise and other fluctuation of values, (3) how to improve the discriminability.

To solve the first problem it is inevitable to employ hashing techniques. Let $\mathbf{v} = (v_1, ..., v_d)$ be a d-dimensional real valued vector that represents a keypoint. Hashing requires to convert it into an integer called the hash key by quantization. If the quantization steps are large enough to absorb the fluctuation of values, the second problem is eased.

In our method we utilize vectors produced by PCA-SIFT. Since the mean of the value v_i of each dimension is around zero, we simply convert it into a bit-vector $\mathbf{w} = (w_1, ..., w_d)$ where $w_i = 1$ if $v_i \geq 0$; otherwise $w_i = 0$. The vector $\mathbf{w}$ obtained by the quantization is converted into an integer by the following hash function:

$$H_{\text{key}} = \left(\sum_{i=1}^{d} w_i 2^{i-1} \right) \bmod H_{\text{size}} \tag{1}$$

where H_{size} is the size of the hash table.

For the third problem, however, larger quantization steps generally spoil the discriminability of vectors. This is because different real valued vectors can be converted into the same quantized vector. In order to improve the discriminability, LLAH employs not only the vector of a point of interest p_0 but also vectors of points around p_0. Let $\mathbf{w}_0$ be the quantized vector of p_0, $\mathbf{w}_1, ..., \mathbf{w}_n$ be the vectors of the n nearest neighbors $p_1, ..., p_n$ of p_0 where $\mathbf{w}_j = (w_{1j}, ..., w_{dj})$. In order to cope with the case that some of the n nearest neighbors are disappeared, possible arrangements of m points out of n nearest neighbors are utilized to form concatenated vectors whose dimensions are $d \times m$.

Figure 2 illustrates an example for the case of $n = 3$ and $m = 2$ where lines indicates the selected points for concatenations. In LLAH, all possible combinations $\{p_0, p_1, p_2\}$, $\{p_0, p_1, p_3\}$, $\{p_0, p_2, p_3\}$ are considered to form concatenated vectors. For the first combination, for example, the concatenated vector is $(\mathbf{w}_0; \mathbf{w}_1; \mathbf{w}_2) = (w_{10}, ..., w_{d0}, w_{11}, ..., w_{d1}, w_{12}..., w_{d2})$. Thus the point p_0 is indexed using these three combinations separately. Note that we can achieve the robustness against missing points using the above technique of taking combinations of points. In the case of Fig. 2, missing of one point out of three has no harmful effect when p_0 is retrieved.

In the proposed method, all concatenated vectors are calculated and utilized for indexing of each point. A hash key H_{key} calculated from a concatenated $d \times m$ dimensional bit vector is utilized to store the corresponding point with the object ID. Thus the hash table stores the point ID with its object ID indexed by the hash key. At the process of creation, collision is simply handled by chaining.

2.2. Retrieval

The retrieval process shares most of the processing steps with the creation. From a query image given by the user, keypoints are extracted and quantized by the same process. In order to form the concatenations, however, it is necessary to determine the order of points to be concatenated. In LLAH, except for the point of interest p_0, all cyclic permutation of points are simply examined. For the leftmost example in Fig. 2, possible concatenations are $(\mathbf{w}_0; \mathbf{w}_1; \mathbf{w}_2)$ and $(\mathbf{w}_0; \mathbf{w}_2; \mathbf{w}_1)$.

Through the process of looking up the hash table followed by the voting for objects, we can determine the corresponding object as the one with the largest vote.

3. Experimental results

3.1. Overview

Experiments were performed using a workstation with AMD Opteron 2.8GHz with 16GB memory. The proposed

(a) 45 deg. (b) 60 deg. (c) 75 deg. (d) large (e) small

Figure 3. Examples of processing results. $d = 16$**,** $n = 28$ **and** $m = 1$ **were employed.**

method is evaluated as compared to the methods SIFT and PCA-SIFT using a brute-force matching with the Euclidean distance as a measure of dissimilarity between keypoints [1]. Because the matching is without any sophisticated mechanisms such as k-d trees, it represents the upper bound of the processing time required for recognition. For the proposed method we tested various parameter values within the following ranges: the number of dimensions: $9 \leq d \leq 36$, the number of nearest points: $n = 0$, $5 \leq n \leq 30$, and the number of combined points : $0 \leq m \leq 3$. Note that $m = 0$ is meaningful only with $n = 0$, which indicates only the central point is utilized for indexing.

As the data for experiments we prepared 40 planar objects including 35 covers of "Comm. of the ACM", 3 posters, a journal cover and a book cover. These objects were captured using a digital camera with 6.3 million pixels. Original color images of size 3042×2048 were converted to 8 bit gray-level images of size 1024×683. Frontal views of objects, such as shown in the lower parts of Fig. 3, were stored in the database

We prepared query images taken from five different views: 45, 60, 75 degrees as well as large and small. The first three views indicate the supplement of the angle between the normal of the object surface and the optical axis. In these images, the whole objects were captured. For the views large and small, on the other hand, images were taken from the angle of 90 degree, but their sizes are different from the ones in the database. Large indicates that only a part (about 50%) of the whole object was in the images. Small means the objects were taken from a larger distance. Upper parts of Fig. 3 show examples of views 45, 60, 75 degrees as well as large and small.

[1]Source codes of SIFT and PCA-SIFT are available at `http://www.cs.ubc.ca/~lowe/keypoints/` and `http://www.cs.cmu.edu/~yke/pcasift/`, respectively.

3.2. Results and discussions

Figure 3 also shows processing results by the proposed method with the parameters $n = 28$ and $m = 1$: the lower parts represent the retrieved image in the database and the white lines between the upper and the lower parts indicate matching between keypoints. Although the matching includes some errors, all images were correctly recognized for the case of this figure.

The overall processing results are shown in Table 1 that lists the average accuracy and the processing time for 200 queries. As shown in this table, SIFT and PCA-SIFT worked well in terms of the accuracy. However, the processing time was unsatisfactory for both methods: 3.3 min./query for SIFT and 26 sec./query for PCA-SIFT. Although the efficiency can be improved using sophisticated data structures, it is not so easy for such technologies to handle tens of thousands of images in real time.

On the other hand, although the proposed method was inferior to SIFT and PCA-SIFT with respect to the accuracy, it achieved much faster processing time: 26 msec. with $n = 0$, $m = 0$ and 61 msec. with $n = 28$, $m = 1$, which are less than 0.04% of SIFT and 0.3% of PCA-SIFT. Such fast processing can be achieved with the help of hashing techniques: the proposed method does not "search" corresponding points but just find it by looking up the hash table.

For the experiments with document images [6], it is reported that the processing time stays almost constant to the size of the database. For example, documents can be retrieved within 100 msec. from the database of size 10 document images, and 140 msec. from the database of size 10,000 document images. The proposed method is expected to be likewise scalable, since it is based on the same technology.

The proposed method with $d = 24, n = 0, m = 0$ was the fastest among the tested combinations of parameter values. The method with $n = 28, m = 1$ was posed the high-

Table 1. Processing results.

method	accuracy	time /query [msec]		
	[%]	ave.	max.	min.
SIFT	100	2.0×10^5	1.2×10^6	1.8×10^4
PCA-SIFT	100	2.6×10^4	1.6×10^5	2.3×10^3
Proposed ($d = 24, n = 0, m = 0$)	89.0	26.3	170	<10
Proposed ($d = 16, n = 28, m = 1$)	90.5	61.2	380	10

Table 2. Analysis of errors.

	ave. rank of	no. of incorrectly retrieved images					ave. ratio of votes	
	failed correct images	45°	60°	75°	large	small	succeeded	failed
$d = 24, n = 0, m = 0$	5.6	21	1	0	0	0	10.18	1.14
$d = 16, n = 28, m = 1$	3.8	13	0	0	0	6	7.08	1.18

est accuracy with $d = 16, 18, 20, 22, 24$. The combination $d = 16, n = 28, m = 1$ was the fastest with the highest accuracy.

Some aspects of erroneous retrieval are shown in Table 2. The average rank means the average rank of correct images that were not correctly retrieved. For the parameters $d = 24, n = 0, m = 0$, correct images were at the second rank for 27% of total errors, and 64% were at less than or equal to the fifth rank. For the parameters $d = 16, n = 28, m = 1$, 1/3 of total errors were due to the correct images at the second rank, and 89% were within the fifth rank. The latter is more suitable to users who select corresponding images from candidates.

The distribution of incorrectly retrieved images indicate that the former will be close to 100% accuracy if we do not take the angle 45 degree into consideration.

The average ratio of votes indicates the ratio of votes for the first and second ranked images: a method is more stable with a larger ratio of correct retrieval. The ratio for failed retrieval indicates that we can reject some erroneous retrieval with a threshold of the ratio. For example, all recognition errors can be eliminated by thresholding the ratio. In these cases the accuracy of retrieval was dropped to 75%.

4. Conclusions

We have presented an efficient method of object recognition for the purpose of accessing information using camera-captured images of planar objects. The characteristic point of the proposed method is the use of hashing techniques for shortening drastically the processing time of recognition. Experimental results show that the proposed method is more than 400 times faster than the brute-force method of comparing keypoints, though there remains some room for improvement in the accuracy.

Future work includes experiments with a larger number of images as well as various imaging conditions such as blurring and uneven lightening.

Acknowledgment

This work was supported in part by JST Innovation Plaza Osaka-Grant, The research grant of the Okawa Foundation for Information and Telecommunication, and Faculty Innovation Research Grant, Graduate School of Engineering, Osaka Prefecture University.

References

[1] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM*, 45(6):891–923, 1998.

[2] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proc. of 30th STOC*, pages 604–613, 1998.

[3] Y. Ke and R. Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *Proc. of CVPR*, volume 2, pages 506–513, 2004.

[4] Y. Ke, R. Sukthankar, and L. Hunston. Efficient near-duplicate detection and sub-image retrieval. In *Proc. of ACM International Conference on Multimedia*, pages 869–876, 2004.

[5] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[6] T. Nakai, K. Kise, and M. Iwamura. 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 —. In *Proc. of the Camera-Based Document Analysis and Recognition*, pages 87–94, 2005.