A SCALABLE INDEXING METHOD FOR SIFT FEATURES

Wei-Chih Hung, Yi-Leh Wu, Tsung-Da Ho, Kuan-Yuo Lin, and Cheng-Yuan Tang

Department of Computer Science and Information Engineering
National Taiwan University of Science and Technology, Taipei, Taiwan
Department of Information Management, Huafan University, Taiwan
E-mail: ywu@csie.ntust.edu.tw

ABSTRACT

This paper presents a method for the image recognition in a large database. The local invariant features and two-tier hashing are used so that the time complexity of image matching is not affected by the size of the image database. We propose a method to simplify the PCA-SIFT descriptor via binarization. The binarized PCA-SIFT keypoint descriptors still keep very high distinctiveness for the image matching process. Furthermore, the binarized PCA-SIFT keypoint descriptors can be indexed by two-tier hashing method very efficiently. Moreover, to reduce the time complexity of matching keypoint descriptors, minimal perfect hashing is applied to build a number of external hash tables. Through computation we can determine which external hash table is hit by the distinct keypoint descriptor without searching. The experiment result suggests that the proposed method can effectively keep the time complexity to O(1) irrespective to the number of images in the matching applications.

Keywords PCA-SIFT, Indexing, Minimal perfect hashing, External hash tables, Digital right management

1. INTRODUCTION

The image recognition methods serve as the foundations of applications such as similar image searching and copyrighted images detection. In the past few years a considerable number of studies have been made on the image retrieval problems [5, 6, 7]. The image retrieval methods consist of several technologies from different research domains such as machine learning, data mining and image process, etc. We limit discussions are constrained here by methods which are able to speed up the searching process in a large scale image database.

The SIFT (Scale-Invariant Feature Transform) is a robust feature extraction method for various 3D and 2D image applications. Lowe first proposed the SIFT algorithm in 2004 [1, 2]. The most important property of SIFT is that the SIFT descriptors are invariant under many image transformations (e.g., rotation, scaling, translation etc.) so that the SIFT algorithm is very effective for image retrieving. We can decide whether the pairs of images are of the same source with only very few pairs of keypoint descriptors matched (usually 3) from two given images. PCA-SIFT [3] was proposed to reduce the dimension of the SIFT descriptors to speed up keypoint descriptors matching. The Principal Components Analysis (PCA) [4] is a typically approach to reduce dimension. The PCA-SIFT algorithm reduces the 128-dimension SIFT descriptors to 36-dimension by using the Principal Components Analysis (PCA).

However the SIFT algorithm consider only grey level images, the CSIFT [8] proposed to include color invariant features. The color invariant features are discussed in detail in [9]. The number of detected keypoints increases with the number of added color invariant elements. The extra detected keypoints will encumber the performance of the retrieving process.

Regardless of the extraction methods, the keypoint descriptors of images are commonly in high dimension which leads to the problem of the curse of dimensionality. This dimensionality problem occurs when a bulk of keypoint descriptors are distributed over boundary of the feature space and are far away from the center of the feature space. To solve this problem, there are many multi-dimension access methods proposed [10]. But even some well-known multi-dimension indexing methods (e.g., the R-tree [11], the KD-tree [12] etc.) can usually reduce to lower dimension space (less than 10 dimensions). The LSH (Locality Sensitive Hashing) methods, proposed by Indyk and Motwani [13, 14], are the multi-dimension indexing methods. There are several other applications that use the LSH method to solve the k-NN (k Nearest Neighbors) queries [13]. Ke proposed to index the PCA-SIFT descriptors using the LSH to solve the sub-image near-duplicate and part-based retrieval problem [15]. The general indexing structures using the LSH method can be found in [16]. Gong and Lazebnik proposed to minimize the quantization error using the iterative quantization (ITQ) method and the canonical correlation analysis (CCA) [19].

The general problem to solve is to match an image efficiently in a large image database. But the general indexing methods are not applicable in high-dimensional space. The main contributions are: first, we reduce the necessary number of keypoints in the image database and
second, we propose a simple and efficient two-tier hashing structure while maintaining high retrieval accuracy.

In Section 2, we discuss the PCA-SIFT method. We present the proposed DRM system and the two-tier hashing system employed in this work in Section 3. Experimental results using 1 million image dataset are presented in Section 4. Finally, we conclude this work and discuss future research directions in Section 5.

2. PCA-SIFT

The detail of the PCA-SIFT can be found in [3]. The Principal Components Analysis (PCA) [4] is a common technology for reducing dimension. The first three stages of PCA-SIFT and SIFT are identical. To pre-build an eigenspace, the PCA-SIFT method performs the first three stages of SIFT from some images and 21,000 patches. The final stage of the PCA-SIFT method extracts a 41*41 patch at the given scale of SIFT descriptors to estimate the principal components. The input vector contains 39*39*2 = 3042 elements which are produced by calculating the horizontal and the vertical gradient. The PCA is then applied to the covariance matrix of each vector and extracts the top n eigenvector as the projection matrix of PCA-SIFT. The normalized 3042-element gradient vectors are projected on low dimension space by using the eigenspace. Here we use n = 36 in the experiment to maintain high accuracy. We project the 128-dimensions SIFT descriptors onto the 36-dimensions PCA-SIFT descriptors. In [3], regardless of correctness or speed, it is shown that the PCA-SIFT method outperforms the SIFT method.

3. PROPOSED DRM SYSTEM

We propose a novel method for the Digital Right Management (DRM) to enhance the speed of image matching in a large image database. Fig. 1 shows the architecture of the proposed DRM method.

![Fig. 1. The architecture of the proposed DRM system.](image)

The proposed DRM method consists of three parts; the first part is the preprocessing, the second part is a two-tier hashing structure, and the final part is the method of matching a input query image.

3.1. Preprocessing Database

The dataset we used throughout this study is an image database with one million images from the Internet. First, all images are converted to the Portable Grey Map (PGM) format and then we extract the SIFT descriptors (128-dimension) from each converted image in the image database. Furthermore, each image is transformed to the PCA-SIFT descriptors (36-dimension). The keypoints are represented as vectors of values in the PCA-SIFT descriptors. To recognize which keypoint belongs to which image, each keypoint is assigned an image ID number. The preprocessing step helps to reduce the size of the keypoint database and to simplify the indexing structure.

3.2. Feature Reduction

Except from the reduction of the dimension of keypoints, it is also important to filter out unnecessary and distorted keypoints. The total number of keypoints extracted from the 1 million image dataset is 216,734,674. Ho [19] proposed methods to reduce the number of necessary keypoints by reserving only the robust keypoints with high geometric variances while keeping high retrieval accuracy, which is shown in Fig. 2. At first each image in the 1 million image dataset is rotated and scaled. There are four geometric transformations used: rotate 90 degrees, rotate 180 degrees, scale 2 times and scale 0.5 times. The keypoints extracted from the transformed images and the ones from the original images are compared. If the same keypoint pairs can be matched (with Euclidean distance less than 3000) from the transformed images and the original images it is considered that these keypoints are very robust against geometric variations. By keeping only the robust keypoints we can effectively reduce the size of the keypoint database. However, sometimes we can extract very few robust keypoints in some images, which makes the matching of those images difficult. Hence we set the constraint that for each image at least F keypoints will be kept in the keypoint database. The value of F is decided as shown in Equation (1). Let $\overline{X}$ is mean value of number of keypoints in training image,

$$\overline{X} = \frac{X_1 + X_2 + \ldots + X_n}{n} = \frac{1}{n} \sum_{i=1}^{n} X_i$$

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (X_i - \overline{X})^2}$$

Where $n$ is number of the training image and $\overline{X}$ is mean of the number of keypoints in all training images. For the Corel image dataset, $\overline{X}$ is 141 and $\sigma$ is 90. Note that these values
are decided after the keypoint reduction process. The \( F \) values for each training image are decided as below:

\[
\begin{align*}
\text{If } \bar{X} - \sigma & \leq R \leq \bar{X} + \sigma & F &= R \\
\text{If } R < \bar{X} - \sigma & & F &= R + \text{offset} \leq \bar{X} \\
\text{If } R > \bar{X} + \sigma & & F &= \bar{X} + \sigma
\end{align*}
\]  

(1)

In Equation (1), \( R \) is the number of keypoints in \( F \), as described in Fig. 2. The \textit{offset} is the number of keypoints that are picking randomly from the original training images until \( F = R + \text{offset} \leq \bar{X} \). Through the adjusting mechanism, we keep the number of keypoints of each training image within range of \( \bar{X} \) and one \( \sigma \). If \( R \) is between \( \bar{X} + \sigma \) and \( \bar{X} - \sigma \), we set \( R \) as \( F \). If \( R \) is more than \( \bar{X} + \sigma \), we reduce number of keypoints in \( R \) until \( F \) is equal to \( \bar{X} + \sigma \).

**Input:**
- \( T_i \) j-th keypoint in i-th training image.

**Variables:**
- \( rP_k \): k-th keypoint in i-th training image after rotated 90 degrees.
- \( r100_k \): k-th keypoint in i-th training image after rotated 100 degrees.
- \( r250_k \): k-th keypoint in i-th training image after rotated 2.5 times.
- \( r0.5_k \): k-th keypoint in i-th training image after scaling 0.5 times.
- \( P_k \): Reserve keypoints after reduced finally.

**Procedure:**
// \( P_k \) is number of keypoints in training image.
// \( G_k \) is number of keypoints in training image of different geometry
For \( i = 1 \rightarrow 1,000,000 \)
For \( j = 1 \rightarrow P_1 \)
For \( k = 1 \rightarrow G_k \)
For \( r = 1 \rightarrow 4 \)
If \( T_{ij} \) can find a match in \( rP_k, r100_k, r250_k \) and \( r0.5_k \) simultaneously then store \( T_{ij} \) to \( P_k \).

![Fig. 2. Keypoint reduction algorithm.](image1.png)

The number of training images in the experiment is 1,000,000 with total of 216,734,674 keypoints. After applying the proposed reduction process, we decrease the number of keypoint to 43,324,865, with the reduction rate of almost 80 percent. In the experiment section, we show experimental results that the proposed reduction process not only provides advantages in space and time but also without loosing retrieval accuracy.

### 3.3. Indexing Structure

We analyze the information about the training keypoints before selecting the proper indexing structure. The distribution of the keypoint values in each dimension of all training images is shown in Fig. 3. Note that the keypoints used here are before applying the reduction process discussed in Section 3.2. Fig. 3 shows the distribution of the 36-dimension PCA-SIFT descriptors of 216,734,674 keypoints with total number of descriptor values. From Fig. 3, we observe a highly skewed normal distribution with most of the descriptor values range from -1000 to 1000.

![Fig. 3. Distribution of the PCA-SAFT descriptor values in the training set.](image2.png)

Two matched keypoint pairs with theirs corresponding descriptor values are both illustrated in Fig. 4. The top of diagram shows the Euclidean distance. The X-axis is the 36 dimensions (descriptor values) used in PCA-SIFT. The Y-axis is the descriptor value of each dimension. Note that if the Euclidean distance is less than 3000 which is considered a match in the PCA-SIFT [15]. From Fig. 4, we observe that a highly similar distribution of the descriptor values occurs between the training keypoints and the query ones if there is a match. When the Euclidean distance is low it is expected that the two lines are closed to each other. However when the Euclidean distance is closed to 3000 it is can still observed the same behavior shown in the bottom of Fig. 4. Fig. 4 shows very little discrepancy in comparing a pair of training and query keypoint if a match occurs.

![Fig. 4. Two distribution diagram of testing and training line.](image3.png)
We proposed to binarize the PCA-SIFT descriptors and use the binarized descriptors as the hashing keys based on the previous observations. The formula of the binarization is shown in Equation (2).

\[
\begin{cases}
  \text{If } K_i \geq 0 & H_i = 1, \quad i = 1 \sim 36 \\ 
  \text{If } K_i < 0 & H_i = 0
\end{cases}
\]

In Equation (2), \(K_i\) is the value of the \(i\)th dimension in the keypoint descriptor. \(K_i\) is transformed into hash key as \(H_i\). If \(K_i \geq 0\), \(H_i\) is set to 1; otherwise is set to 0. Because the keypoint descriptors are 36-dimension the resulting hash keys consist of 36 digits of 1’s and 0’s with \(2^{36}\) possible combinations.

At first we transform all the training images into grey level as the input images. Second, the SIFT descriptors are extracted from the training images and then are reduced the dimension of the descriptors by using the PCA. The keypoint reduction process as described in Fig. 2 is applied to keep only the robust keypoints. Finally, we binarize each keypoint to binary hash key using Equation (2) and store the key and image ID pairs into an external hash table built by two-tier hashing system.

3.4. Two-tier Hashing

After the preprocessing is completed, to speed up the image matching in a large set of keypoints we need to build an external hash table which records the relationship between an image keypoint and the image ID number. Therefore, the time complexity of the image matching can be kept as constant via the external hash table.

To achieve the goal we propose a two-tier hashing structure. Let the number of total keypoints be \(n\). First, we run all keypoints through a minimal perfect hashing algorithm to obtain a minimal perfect hash table, where each keypoint generates a unique hash key, \(0 \leq \text{hash key} \leq n-1\). Minimal perfect hashing guarantees that \(n\) keys will map \(0 \sim n-1\) keys with no collisions at all.

\[
\text{Hash table name} = \left\lfloor k / 2000\right\rfloor, k \in \text{hash keys}
\]

We use the minimal perfect hashing method because the size of the data set is very large so that the size of the external hash tables must be very large too. To reduce the access I/O time of a large external hash table, the whole external hash table is split into smaller sub hash tables, and each sub external hash table is limited to a specific size. The experiment determines that each sub external hash table contains at most 2000 keypoints hence the number of total sub external hash tables is \(n / 2000\). As we know that each keypoint has a unique hash key value between 0 and \(n-1\) via minimal perfect hashing, we can directly compute the input keypoint belongs to which external table. If the minimum perfect hash key of the input keypoint is \(k\), the keypoint is stored in hash table numbered \(k / 2000\).

The minimal perfect hashing can only create the relation between keypoint and hash key but no relation about image ID number. To record which keypoint belongs to which image ID number, we have to perform a second reverse hashing. The GNU C++ hash_map provides the functionality to store value pairs of (key, value). All hash keys of keypoints and their image ID numbers are together hashed into the hash_map structures. Finally the hash_map structures are serialized as external hash tables so that we can access the target external hash tables according to the conversion rule in Equation 3.

3.5. Two-tier Hashing

We now discuss how to query if an input image is in the database. Step 1, step 2 and step 3 in Fig. 1 are the same for querying images. From the query image, we extract the 36-dimension descriptor of each keypoint to a hash key by using Equation (2). We then use the hash keys in the query image to check if we can find a match in the hash table. If the hash key of query is found in the hash table, the returned value is the image ID number in that bucket.

The No matched result indicates that this bucket does not appear in the hash table and the null value is simply discarded. If matched, we then store each returned value (image ID number) into a matched list. To decide the final matched image we use simple voting with the largest number of the same image ID to decide the training image that matches with the query image. Note that the retrieval process needs no calculation of the Euclidean distance among keypoint, which is an important factor for efficiency and scalability. In addition, the accuracy of the retrieval process is very high. The detailed procedure of the retrieval process is depicted in Fig. 5.
4. EXPERIMENTS

The following experiments try to consolidate previous conjectures: The proposed indexing structure is indeed accurate, scalable and efficient.

4.1. Experimental Environment

The hardware environment for all experiments is as followings: INTEL CPU Q9550 2.83 GHz, 4G bytes DDR2 800 memory. The number of the image data set is 1,000,000, which is used as the training dataset. The two-tier hashing is the main indexing structure. The first hashing is minimal perfect hashing, and the second one is external hashing. The two-tier hashing do not support duplicate data therefore each image is unique in the dataset. Also note that all the experiment results in this section are measured by cold-start queries of the external hash tables to eliminate the potential influence of the caching mechanism.

Table 1. Total space for keypoints of different image dataset size.

<table>
<thead>
<tr>
<th>Number of images</th>
<th>50,000</th>
<th>100,000</th>
<th>200,000</th>
<th>500,000</th>
<th>800,000</th>
<th>1,000,000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size of image (pixel)</td>
<td>256*384</td>
<td>256*384</td>
<td>1024*768</td>
<td>1024*768</td>
<td>1024*768</td>
<td>1024*768</td>
</tr>
<tr>
<td>Disk space of external hash tables (MB)</td>
<td>37.9</td>
<td>72.4</td>
<td>139.9</td>
<td>363.2</td>
<td>551.3</td>
<td>738.4</td>
</tr>
</tbody>
</table>

Fig. 6 and Fig. 7 shows that the total size of external hash tables and the total number of keypoints both increase linearly as the size of training dataset increases.

4.2. Experimental Results

We now experiment the retrieval time with different sizes of the training database. Fig. 8 shows the matching time of 200 image retrievals in different training dataset sizes. We observe that the time complexity of retrieval time is O(1) to the size of the datasets. The retrieval time only depends on how many external hash tables have been accessed via equation (3). The main time cost is the I/O time to access the external hash tables. This result suggests that the size of dataset does not affect the retrieval time and the proposed two-tier hashing method is indeed scalable and efficient.

Fig. 9 shows the query accuracy of 200 image retrievals in different training dataset sizes. We observe that the accuracy of retrieval decrease very gradually as the database size increases, which is as we expected. However, even with database of millions of images, the proposed system still has accuracy higher than 98%. The result suggests that the proposed method is indeed quite accurate and can be applied to real-world DRM systems.
5. CONCLUSIONS AND FUTURE WORK

The Scale Invariant Feature Transform (SIFT) has been widely used in many 2D and 3D image matching allocations. However, the large number of local invariant keypoints extracted by the SIFT poses a scalability problem when the number of images to be matched increases. We propose a method to simplified the SIFT descriptor matching processing. First we adopt the PCA-SIFT method to reduce the dimension of each descriptor from 128 to 36. We show that the binarized SIFT descriptors still maintain very high distinctiveness for the image matching process in two-tier hashing structure. The binarized SIFT descriptors can be indexed by the proposed two-tier hashing system very efficiently. Most important of all, we successfully build the proposed system to reduce the time complexity of querying image to O(1) in a large database with high accuracy. The experiment results suggest that the proposed method can effectively alleviate the scalability problem in large scale image matching applications. One of future research directions is to develop more robust and less overhead external hash structures to further increase the descriptor matching efficiency. Other research directions are to apply the matching algorithms on real world applications such as the Digital Right Management systems.

ACKNOWLEDGEMENT

This work was partially supported by the National Science Council, Taiwan, under the Grants No. NSC101-2221-E-011-141, NSC100-2221-E-011-121, and NSC101-2221-E-211-011.

REFERENCES