This article has been accepted for publication in a future issue of IEEE Transactions on Pattern Analysis and Machine Intelligence. Citation information: DOI:10.1109/TPAMI.2014.2382092

Mixture of Subspaces Image Representation and Compact Coding for Large-Scale Image Retrieval

Takashi TAKAHASHI† and Takio KURITA‡

† Department of Applied Mathematics and Informatics, Ryukoku University,
Otsu, Shiga, 520-2194, Japan.
E-mail:takataka@math.ryukoku.ac.jp
‡ Department of Information Engineering, Hiroshima University,
1-4-1 Kagamiyama, Higashi-Hiroshima-shi, Hiroshima, 739–8527, Japan.

Abstract There are two major approaches to content-based image retrieval using local image descriptors. One is descriptor-by-descriptor matching and the other is based on comparison of global image representation that describes the set of local descriptors of each image. In large-scale problems, the latter is preferred due to its smaller memory requirements; however, it tends to be inferior to the former in terms of retrieval accuracy. To achieve both low memory cost and high accuracy, we investigate an asymmetric approach in which the probability distribution of local descriptors is modeled for each individual database image while the local descriptors of a query are used as is. We adopt a mixture model of probabilistic principal component analysis. The model parameters constitute a global image representation to be stored in database. Then the likelihood function is employed to compute a matching score between each database image and a query. We also propose an algorithm to encode our image representation into more compact codes. Experimental results demonstrate that our method can represent each database image in less than several hundred bytes achieving higher retrieval accuracy than the state-of-the-art method using Fisher vectors.

1 Introduction

This paper addresses the problem of content-based image retrieval based on local image descriptors. By using expressive local image descriptors such as SIFT [1], even simple descriptor-by-descriptor matching can achieve high retrieval accuracy. However, there are two major difficulties when applying this approach to a large-scale problem. One is the computational cost for matching every local descriptor in the database with those extracted from a query image. The other is the memory cost for storing immense numbers of local descriptors in the database. To solve the former problem, many studies have been devoted to developing methods that apply approximate nearest neighbor (ANN) search [2, 3, 4, 5]. On the other hand, one approach to avoid the later problem is to develop a compact global image feature representation that preserves the information of the set of local descriptors extracted from each image. The bag-of-features (BoF) or bag-of-visual-words is the most common of this type of approach [6]. This paper investigates a high-performance image retrieval method that utilizes an efficient and compact image feature representation.

In a typical BoF-based image retrieval method, the distributions of local descriptors are summarized into histograms that count the occurrence of visual words. These histograms are used as image-wise features to measure the distance between images. Given a query image, this distance is used to compute the matching score between the query and each of the images in the database. This BoF image representation is rather compact compared to the set of raw local descriptors, and thus reduces the memory costs. However, the BoF-based method is not competitive with ANN-based methods in terms of retrieval accuracy. Jégou et al. showed that the BoF approach can be interpreted as the voting of local descriptors and proposed a method that combines the BoF approach with an ANN search [7, 8]. They reported that their method shows significant improvement in image retrieval accuracy compared to the conventional BoF-based method.

There are also various methods that represent image features in distinct ways from the BoF approach. Jégou et al., for instance, proposed the image feature representation called vector of locally aggregated descriptors (VLAD) [9, 10]. They also investigated the Fisher Vector (FV) method [11, 12] for image retrieval. It has been demonstrated that the VLAD and FV methods give higher accuracy than the conventional BoF approach [9, 10]. It has also been demonstrated that these image features can be encoded into less than several hundred bytes per image without degrading performance by using

their product quantization method [5, 9, 10]. To the best of our knowledge, the combination of FV image representation and this encoding method by Jégou et al. is a state-of-the-art approach that achieves both high accuracy and reduced memory costs. Recently, an extension of VLAD called the vector of locally aggregated tensors (VLAT), which obtains improved accuracy, has been reported [13, 14]. However, the VLAT method did not involve the encoding of image features. In this paper, we examine a method for attaining superior performance in terms of both retrieval accuracy and the length of codes for representing image features.

Higher retrieval accuracy achieved with the FV image representation is considered to be due to the fact that the FV can describe the distribution of local descriptors in greater detail than BoF. In a typical FV set up, the generation process of local descriptors is modeled by a Gaussian mixture model (GMM). Then, the FV is defined such that the dot product between the two vectors becomes a similarity measure between the model parameters of two corresponding images, such as the mixture weights, means, and covariance matrices of the GMM. This is in contrast to the BoF representation, in which the distribution is described only by counting the number of local descriptors assigned to each cluster.

Both of these approaches adopt a common scheme for representing image features in identical dimensions for both database images and queries. In image retrieval, there exists asymmetry such that the feature of each database image is desired to be described in compact dimensions; however, this is not the case for a query. We can distinguish database features from query features so long as some similarity can be evaluated between a query and each database image. If we can combine exclusive compact feature representation for database images with a rich representation for queries, such as asymmetric image representation, we could achieve both high retrieval accuracy and low memory costs.

In this study, we examine an asymmetric approach in which the probability distribution of local descriptors is modeled for each individual database image while the local descriptors of a query are used as is. We adopt a mixture model of probabilistic principal component analysis (probabilistic PCA or PPCA) as the model. Then, the likelihood function of each model is employed as a matching score to measure the similarity between a query and each database image. In this approach, the model parameters of each PPCA mixture model constitute a feature of database images. We further introduce some constraints and approximations into the model for improving computational efficiency. We refer to the image feature representation in this approach as a *mixture of subspaces image representation*. We demonstrate that the image retrieval method utilizing this representation outperforms the method using FV representation. In addition, we investigate how to encode the image features of mixture of subspaces image representation. Experimental results demonstrate that the database image features can be encoded into less than several hundred bytes per image without significantly degrading accuracy. These results are comparable to those obtained by the state-of-the-art encoding method for FVs [10].

The remainder of this paper is organized as follows. Section 2 introduces the mixture of subspaces image representation and proposes an image retrieval method that utilizes it. Section 3 presents experimental results for demonstrate the validity of the proposed method for some large-scale public datasets. Section 4 investigates the method for encoding the mixture of subspaces image representation. Section 5 evaluates the performance of the encoding method, and Section 6 presents conclusions and suggestions for future work.

2 Mixture of Subspaces Image Representation

2.1 Defining a Matching Score Based on PPCA Mixture Models

Let $\mathcal{I}_i$ denote the i-th image to be stored in a database and

$$X_i = \{\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}, \ldots, \boldsymbol{x}_{i,N_i}\} \tag{1}$$

denote the set of local image descriptors extracted from $\mathcal{I}_i$, where N_i is the number of descriptors. Each local descriptor is assumed to be a D-dimensional vector. Now, assume that the local descriptors of $\mathcal{I}_i$ are distributed according to a GMM with K mixtures:

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_{i,k} \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_{i,k}, \boldsymbol{\Sigma}_{i,k}), \tag{2}$$

where $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes a Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. The parameter $\pi_{i,k}$ is the mixture weight of the k-th Gaussian mixture component of $\mathcal{I}_i$, which satisfies $0 \leq \pi_{i,k} \leq 1$ and $\sum_{k=1}^{K} \pi_{i,k} = 1$. The parameters $\boldsymbol{\mu}_{i,k}$ and $\boldsymbol{\Sigma}_{i,k}$ are the mean and covariance matrix corresponding to the k-th mixture component of $\mathcal{I}_i$, respectively. We can then employ the simple idea of using the likelihood function of the model (2) as a matching score $S(\mathcal{I}^{(\mathrm{Q})}, \mathcal{I}_i)$ to rank $\mathcal{I}_i$ with respect to any query image $\mathcal{I}^{(\mathrm{Q})}$. Let $X^{(\mathrm{Q})} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{N^{(\mathrm{Q})}}\}$ denote the set of local descriptors of $\mathcal{I}^{(\mathrm{Q})}$, where $N^{(\mathrm{Q})}$ is the number of descriptors. Provided that $\boldsymbol{x}_n$ are assumed to be independent of each other, we employ the log-likelihood

function as the score:

$$S(\mathcal{I}^{(\mathrm{Q})}, \mathcal{I}_i) = \sum_{n=1}^{N^{(\mathrm{Q})}} \log \left\{ \sum_{k=1}^{K} \pi_{i,k} \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_{i,k}, \boldsymbol{\Sigma}_{i,k}) \right\}. \quad (3)$$

In this approach, the model parameters $\pi_{i,k}, \boldsymbol{\mu}_{i,k}$ and $\boldsymbol{\Sigma}_{i,k}$ $(k = 1, 2, \ldots, K)$ constitute a feature to be stored into the database for each database image. There are no constraints on the choice of covariance for mixture components of the model (2). It may be unfavorable, however, to adopt the full-covariance Gaussian model, as it requires a considerable memory capacity to store the covariance matrices for each database image. Conversely, the diagonal-covariance model seems to be insufficient for representing the local descriptor distribution. Therefore, we employ the PPCA model [15, 16].

In the PPCA model, the covariance matrix $\boldsymbol{\Sigma}$ of (2) is given as (subscripts are omitted)

$$\boldsymbol{\Sigma} = \boldsymbol{W}\boldsymbol{W}^{\top} + \sigma^2 \boldsymbol{I}, \quad (4)$$

where $\boldsymbol{W}$ is a $D \times H$ matrix ($H < D$) and $\sigma^2 > 0$. Consequently, the memory cost for representing each database image is reduced in comparison with the full-covariance GMM. For given samples, the maximum likelihood solution of $\boldsymbol{\mu}$ is the sample mean, while those of $\boldsymbol{W}$ and σ^2 are given as the following forms [15, 16]:

$$\boldsymbol{W} = \boldsymbol{U}(\boldsymbol{\Lambda} - \sigma^2 \boldsymbol{I})^{\frac{1}{2}} \boldsymbol{R} \quad (5)$$

$$\sigma^2 = \frac{1}{D-H} \sum_{d=H+1}^{D} \lambda_d, \quad (6)$$

where $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_D$ are the eigenvalues of the sample covariance matrix, $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_H)$, $\boldsymbol{U}$ is the $D \times H$ matrix composed of the H eigenvectors corresponding to the H largest eigenvalues, and $\boldsymbol{R}$ is an arbitrary $H \times H$ orthogonal matrix.

2.2 Mixture of Subspaces Image Representation

In the previous section, we defined a matching score based on PPCA mixture models. While this model might be appropriate to represent the local descriptor distribution, the score (3) is too complex to compute for every database image. Accordingly, we simplify (3) by introducing some constraints and approximations into the model.

Under mixture models such as (2), the mixture components to which $\boldsymbol{x}$ belongs are generally unknown. By denoting $z_k \in \{0, 1\}$ as the random variable representing whether $\boldsymbol{x}$ belongs to the k-th mixture ($\sum_{k=1}^{K} z_k = 1$) and $\boldsymbol{z} = (z_1, z_2, \ldots, z_k)$, $\boldsymbol{z}$ can be considered a latent variable that should be inferred. Due to this nature of mixture models, all of the K likelihoods $\mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_{i,k}, \boldsymbol{\Sigma}_{i,k})$ must be evaluated for each $\boldsymbol{x}$. This is the most computationally demanding part for our model, especially in the case of large K. To alleviate the computational requirements, we treat $\boldsymbol{z}$ as an observable according to the following assumptions.

- The value $\boldsymbol{z}$ is only dependent on $\boldsymbol{x}$ for all database images.
- The mean of the k-th mixture component is identical for all database images: $\boldsymbol{\mu}_{i,k} = \boldsymbol{\mu}_k$.

Such conditions can be satisfied by applying a clustering algorithm to local descriptors and fixing the corresponding parameters before estimating the model parameters of each database image. We adopt the K-means algorithm to estimate $\boldsymbol{\mu}_k$ and compute $\boldsymbol{z}$. The mean $\boldsymbol{\mu}_k$ is estimated as a cluster centroid using a distinct dataset from the database. The value $\boldsymbol{z}$ is determined for any $\boldsymbol{x}$ by assigning $\boldsymbol{x}$ to one of the K clusters. Under these conditions, the matching score, that is, the log-likelihood function, is given as

$$S(\mathcal{I}^{(\mathrm{Q})}, \mathcal{I}_i) = \sum_{k=1}^{K} \left(N_k^{(\mathrm{Q})} \log \pi_{i,k} + \sum_{n:z_{n,k}=1} \log \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_{i,k}) \right), \quad (7)$$

where $N_k^{(\mathrm{Q})}$ denotes the number of descriptors assigned to the k-th cluster. In (7), the first term in the summation of k penalizes the discrepancy in the allocation of local descriptors to each cluster between the query and the i-th database image. Thus, this score can be regarded as an extension of the conventional BoF approach, which measures the distance between two histograms with K-bins.

We further simplify (7) by introducing some approximations. If the model parameters of each PPCA component are estimated by maximum likelihood estimation and hence the parameter $\boldsymbol{W}$ has the form of (5), the log-likelihood function for the PPCA model is given as follows:

$$\log \mathcal{N}(\boldsymbol{x} | \boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\frac{1}{2} \left(D \log 2\pi + \log |\boldsymbol{\Sigma}| + \frac{1}{\sigma^2} \left(\|\boldsymbol{x} - \boldsymbol{\mu}\|^2 - \|\boldsymbol{\Lambda}^{-\frac{1}{2}} \boldsymbol{W}^{\top} (\boldsymbol{x} - \boldsymbol{\mu})\|^2 \right) \right). \quad (8)$$

For simplicity, we have chosen $\boldsymbol{R} = \boldsymbol{I}$. For the two terms $\log |\boldsymbol{\Sigma}|$ and $\boldsymbol{\Lambda}^{-\frac{1}{2}} \boldsymbol{W}^{\top}$ in (8), the following equations (9)

and (10) hold.

$$\log|\boldsymbol{\Sigma}| = \sum_{d=1}^{H} \log\lambda_d + (D-H)\log\sigma^2 \tag{9}$$

$$\boldsymbol{\Lambda}^{-\frac{1}{2}}\boldsymbol{W}^\top = \mathrm{diag}\left(\sqrt{1-\frac{\sigma^2}{\lambda_1}},\ldots,\sqrt{1-\frac{\sigma^2}{\lambda_H}}\right)\boldsymbol{U}^\top \tag{10}$$

First, let us assume that σ^2 is equal, and $\sum_{d=1}^{H}\log\lambda_d$ takes an identical value for every image. Then, $\log|\boldsymbol{\Sigma}|$ becomes a constant; therefore, this term can be dropped. In addition, let us also assume that $\lambda_H \gg \sigma^2$. Then, (10) is approximately equal to $\boldsymbol{U}^\top$. Therefore,

$$\log\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\Sigma}) \approx \frac{1}{2\sigma^2}\|\boldsymbol{U}^\top(\boldsymbol{x}-\boldsymbol{\mu})\|^2 + \text{const.} \tag{11}$$

holds. The term $\|\boldsymbol{x}-\boldsymbol{\mu}\|^2$ is constant since it takes an equal value for every image. Hence, by substituting (11) into (7) and omitting the constant terms, we obtain

$$S(\mathcal{I}^{(\mathrm{Q})},\mathcal{I}_i) = \sum_{k=1}^{K}\left(N_k^{(\mathrm{Q})}\log\pi_{i,k} + \frac{1}{2\sigma_k^2}\sum_{n:z_{n,k}=1}\|\boldsymbol{U}_{i,k}^\top(\boldsymbol{x}_n-\boldsymbol{\mu}_k)\|^2\right). \tag{12}$$

The parameter σ_k^2, corresponding to σ^2 of (6), is assumed to have an identical value for all database images; therefore, it can be determined using a distinct dataset from the database as $\boldsymbol{\mu}_k$. We propose to use (12) as the matching score for image retrieval. As mentioned above, the first term in the summation of k in (12) is a penalty for the discrepancy of the distribution of local descriptors to K clusters between the query and the i-th database image. The second term is considered to measure the similarities of the query descriptors $\boldsymbol{x}_n$ with respect to the k-th subspace of the i-th database image. In this approach, the features of each database image are represented by $\pi_{i,k}$ and $\boldsymbol{U}_{i,k}$. We refer to such image feature representation as a *mixture of subspaces image representation*. In Section 3.2, we experimentally confirm that the introduced constraints and approximations do not degrade retrieval accuracy of our approach.

The above treatment simplifies both model parameter estimation and query processing. The matrix $\boldsymbol{U}_{i,k}$ can be estimated through eigendecomposition of the correlation matrix of the descriptors belonging to the k-th component of $\mathcal{I}_i$. The parameter $\pi_{i,k}$ can also be estimated from the number of descriptors belonging to each component. These are described in detail in Section 2.3. In query processing, local descriptors are assigned to one of the K components, so that the approximated log-likelihood value is computed only once for each of them. Additionally, it is worth noting that the computation of the score (12) is computationally efficient since it almost consists of basic arithmetic operations in contrast to the original score (3) or FV.

As discussed in the next section, applying (12) as it is on raw SIFT descriptors produces suboptimal results. Hence, as proposed by [12, 10] for FV, PCA is applied to the local descriptors to reduce their dimensionality. We apply PCA to perform dimensionality reduction and whitening on each of the clustered local descriptors as follows:

$$\boldsymbol{x}'_k = \widetilde{\boldsymbol{\Lambda}}_k^{-\frac{1}{2}}\widetilde{\boldsymbol{U}}_k^\top(\boldsymbol{x}-\boldsymbol{\mu}_k), \tag{13}$$

where $\widetilde{\boldsymbol{\Lambda}}_k$ is the $D'\times D'$ diagonal matrix whose diagonal elements are the D' largest eigenvalues of $\mathrm{E}[(\boldsymbol{x}-\boldsymbol{\mu}_k)(\boldsymbol{x}-\boldsymbol{\mu}_k)^\top]$, and $\widetilde{\boldsymbol{U}}_k$ is the $D\times D'$ matrix composed of the corresponding eigenvectors. The resulting $\boldsymbol{x}'_k$ is the D' dimensional vector. Then, by substituting $\boldsymbol{x}'_{n,k}$ for $(\boldsymbol{x}_n-\boldsymbol{\mu}_k)$ in (12), the matching score (12) can be rewritten as follows:

$$S(\mathcal{I}^{(\mathrm{Q})},\mathcal{I}_i) = \sum_{k=1}^{K}\left(N_k^{(\mathrm{Q})}\log\pi_{i,k} + \frac{1}{2\sigma_k^2}\sum_{n:z_{n,k}=1}\|\boldsymbol{U}_{i,k}^\top\boldsymbol{x}'_n\|^2\right). \tag{14}$$

Note that $\boldsymbol{U}_{i,k}$ is the $D'\times H$ matrix in this equation. In the case of FV, it is suggested that PCA dimensionality reduction leads to better estimation for diagonal GMMs because of its decorrelation and noise reduction effects [10]. However, in our case, whitening combined with dimensionality reduction may be expected to enlarge the distance between image subspaces.

2.3 Image Retrieval Method for Mixture of Subspaces Image Representation

Here we summarize the image retrieval method utilizing the proposed matching score and mixture of subspaces image representation.

Preparation

It is necessary to estimate several parameters using a distinct set of images (learning set) from images to be stored in the database.

1. Apply the K-means algorithm on local descriptors of the learning set to obtain $\boldsymbol{\mu}_k$ $(k=1,2,\ldots,K)$.
2. Perform PCA on the clustered data to obtain $\widetilde{\boldsymbol{\Lambda}}_k$ and $\widetilde{\boldsymbol{U}}_k$ $(k=1,2,\ldots,K)$.

3. Compute σ_k^2 as the average of the values obtained by (6) for each image in the learning set.

Storing Images in the Database

The storage procedure for image $\mathcal{I}_i$ is as follows.

1. Extract local descriptors from $\mathcal{I}_i$ and determine their cluster assignments.

2. Compute $\pi_{i,k}$ $(k = 1, 2, \ldots, K)$. These are determined so that

$$\pi_{i,k} = \max(N_{i,k}, 0.1) / \sum_{k=1}^{K} \max(N_{i,k}, 0.1). \quad (15)$$

   The constant 0.1 is used to avoid $\log 0$.

3. Compute $\boldsymbol{U}_{i,k}$ $(k = 1, 2, \ldots, K)$. The columns of $\boldsymbol{U}_{i,k}$ are given as the eigenvectors corresponding to the H largest eigenvalues of the following $D' \times D'$ matrix $\boldsymbol{C}_{i,k}$:

$$\boldsymbol{C}_{i,k} = \frac{1}{N_{i,k}} \sum_{n:z_{i,n,k}=1} \boldsymbol{x}'_{i,n,k} \boldsymbol{x}'^{\top}_{i,n,k}, \quad (16)$$

   where $N_{i,k}$ denotes the number of descriptors in X_i assigned to the k-th cluster. If the eigenvalues are too small or $N_{i,k} = 0$, the corresponding columns should be $\boldsymbol{0}$.

4. Store the parameters $\pi_{i,k}$ and $\boldsymbol{U}_{i,k}$ $(k = 1, 2, \ldots, K)$ in the database.

Processing a Query

Given a query $\mathcal{I}^{(\mathrm{Q})}$, the search for the most similar images in the database is performed as follows.

1. Extract local descriptors from $\mathcal{I}^{(\mathrm{Q})}$ and determine their cluster assignments.

2. Compute the matching scores (14) for all database images. Then, output the order of relevance (decreasing order of the scores) of the images.

3 Evaluation of The Proposed Image Representation

In this section, we evaluate the performance of the image retrieval method using the proposed mixture of subspaces image representation. The results are compared with those obtained by the state-of-the-art FV method [10].

3.1 Datasets and Experimental Procedure

Datasets

In all the experiments, the following four public datasets were employed.

1. PASCAL VOC: This dataset consists of 42467 images collected from PASCAL VOC datasets [17]. We collected all the images from VOC2007 to VOC2012 and removed any duplicates. This was used for constructing the learning set for learning the parameters (K-means clustering, GMM estimation, etc.).

2. INRIA Holidays [7, 18]: This dataset contains 1491 images of 500 different scenes; 991 images were used to construct a database, and the remaining 500 images were used for queries to evaluate the performance of image retrieval. The images were resized so that the maximal length of the longest side was equal to or less than 1024 pixels.

3. University of Kentucky Recognition Benchmark Images (UKB) [19, 20]: This dataset consists of 2550 objects, each having four images. These images were used as an additional dataset to evaluate the performance of image retrieval.

4. Flickr1M [21]: This dataset consists of over one million images that can be downloaded from Flickr. At the time of this writing, however, a portion of the images could not be retrieved; therefore, we employed 980 thousand randomly selected images from all available images. These images were merged with the Holidays images to construct datasets for large-scale retrieval experiments.

Local Descriptor Extraction and Preprocessing

Local image descriptors are extracted from the images in the above datasets following the setup of [10]. The Hessian-affine detector [22] and the SIFT descriptor [1] are employed to extract and describe local image features. We used the software available in [23]. The sampled local descriptors from the VOC dataset were used for estimating the parameters of the proposed and FV image representations. The methods for estimating parameters in the proposed image representation are described in Section 2.3. Here we used the standard K-means algorithm. For the FV image representation, PCA was first applied to reduce descriptor dimensionality to D'. Then, this D' dimensional data was used for estimating the diagonal-covariance GMM with K mixture components as described in [10]. We only use the terms with respect to the mean. In addition, power normalization

with $\alpha = 0.5$ and the L2-normalization were also applied to the FVs as described in [10, 12].

Evaluation Criteria

The performance of the proposed representation was evaluated in terms of image retrieval accuracy against the number of variables per database image in the evaluation datasets. For the Holidays dataset, we measured accuracy using the mean average precision (mAP). The average number of relevant images, including the query itself ranked in the top four search results, was used on the UKB dataset. We denote the latter measure as "KS". The number of floating-point variables of the proposed representation is $K(HD' + 1)$ per image, while that of FV representation is simply KD', because we only use the terms with respect to mean, as described in [10].

3.2 Effects of the Constraints and Dimensionality Reduction

Effects of the Constraints and Approximations on the Models

First, we confirm the validity of our approach using the likelihood functions of mixture of PPCA models as matching scores for image retrieval. We compare the retrieval accuracies obtained by the following three models with those obtained by using the FV representation:

- Fully image-wise PPCA mixture: A mixture of PPCA model was fitted by the EM algorithm for each database image and its log-likelihood function was used for scoring. Since such image-wise EM learning took computation time, we applied the learning only once for each image instead of trying different initial values.
- Constrained PPCA mixture: First, a single PPCA mixture model was fitted for local descriptor distribution sampled from the learning dataset. It was used for determining the component assignments (responsibilities of the K mixture components) of local descriptors. For each database image, the parameters $\pi_{i,k}$ were estimated as the ratios of the sum of the responsibilities, and then K PPCA models were fitted for each component on the condition that their means were fixed at that of the above single model: $\boldsymbol{\mu}_{i,k} = \boldsymbol{\mu}_k$.
- Mixture of subspaces: This model corresponds to the score of (12). The above single PPCA model was replaced with the K-means clustering, and each local descriptor was assigned to only one of the clusters. The PPCA models were simplified through the approximation as described in Section 2.2.

Table 1: Effects of the constraints and approximations on the models. Each accuracy shows the mAP obtained for the Holidays dataset.

Model		accuracy
fully image-wise PPCA mixture	$K = 16, H = 2$	0.631
constrained PPCA mixture	$K = 16, H = 2$	0.650
mixture of subspaces	$K = 16, H = 2$	0.663
Fisher Vector	$K = 16$	0.626
Fisher Vector	$K = 128$	0.661

Table 2: Effects of dimensionality reduction and whitening. Each value shows the mAP obtained by the proposed method for the Holidays dataset.

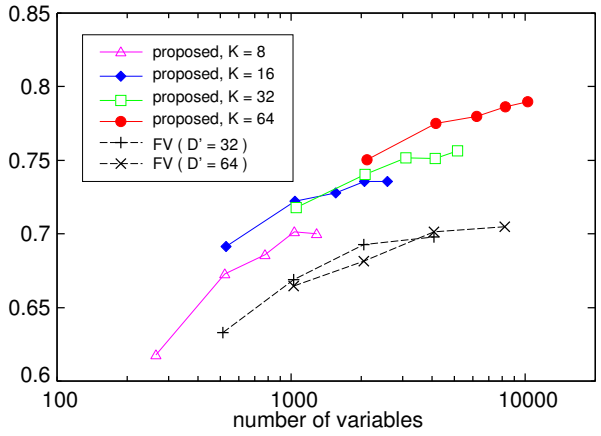
$K = 16, H = 2$	$D' = 16$	32	64	128
w/o whitening	0.654	0.669	0.666	0.663
with whitening	0.708	0.722	0.724	0.670

In this experiment, we did not apply dimensionality reduction to local descriptors in all cases. For the FV representation, we did not apply the power normalization either.

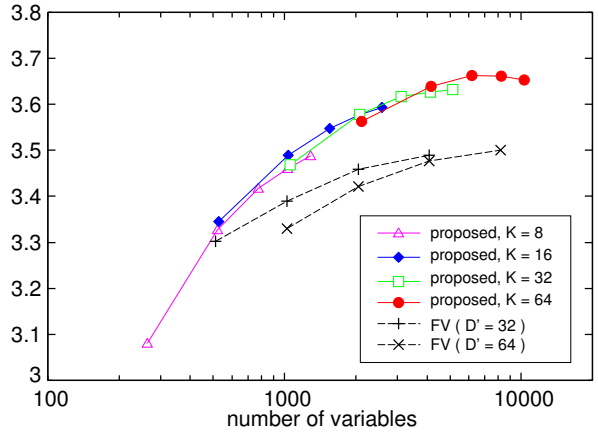
Table 1 indicates the mAP values for the Holidays dataset. The fully image-wise PPCA mixture model shows about the same accuracy as the FV with the same number of mixture components, while the other two PPCA-based models obtain higher accuracies. Specifically the proposed method attains comparable accuracy with the FV with much larger number of components. These results suggest the potential of our approach. Besides, they also imply that the fully image-wise PPCA mixture model might be redundant for describing local descriptor distribution of each individual image since there is little accuracy improvement compared with the FV. The introduced constraints and approximations are significant in terms of image retrieval accuracy as well as computational efficiency.

Dimensionality Reduction and Whitening

We investigate the influence of dimensionality reduction and whitening using the Holidays dataset. Table 2 shows the results for different settings. In the case of $D' = 128$, $D = 128$ dimensional local descriptors were not rotated and were used as is (without whitening) or were used after standardizing each element (with whitening). Although the results obtained are comparable to the FV representation without whitening (see Table3), they are clearly improved by combining whitening with dimensionality reduction. According to these results, we selected $D' = 32$ in the following experiments.



(a) mAP on the Holidays dataset



(b) KS on the UKB dataset

Figure 1: Image retrieval accuracy as a function of the number of variables for representing each database image.

3.3 Image Retrieval Experiments

Experiments on the Holidays and UKB datasets

We compare the image retrieval performance of the proposed method with the FV-based method. The results are presented in Fig. 1 and Table 3. The dimensionality D' was set to 32 or 64 for the FV representation. It can be observed that the proposed image representation outperforms the FV representation for both the Holidays and UKB datasets. In the proposed method, the two parameters K and H should be set in advance. These experimental results suggest that H can be fixed at a small number (e.g., 2 or 3), while a larger number is preferred for K.

Large-Scale Experiments

In order to evaluate the scalability of the proposed method, we conducted the experiments employing large-scale image databases. The databases were constructed by merging a fixed number (10000, 100000 or 980000) of randomly chosen images from the Flickr1M dataset with 991 database images of the Holidays dataset. The remaining 500 images of the Holidays dataset were used for queries.

Table 3: Comparison of the proposed method with the FV-based method. # var: the number of variables for representing each database image; mAP on the Holidays dataset; and KS on the UKB dataset.

	#var	K	H	Holidays	UKB
proposed ($D' = 32$)	264	8	1	0.618	3.08
	520	8	2	0.673	3.33
	776	8	3	0.686	3.42
	1552	16	3	0.728	3.55
	3104	32	3	0.752	3.52
	6208	64	3	0.780	3.66
	8256	64	4	0.786	3.66
	10304	64	5	0.790	3.65
FV ($D' = 32$)	512	16	-	0.633	3.30
	1024	32	-	0.669	3.39
	2048	64	-	0.693	3.46
	4096	128	-	0.698	3.49
FV ($D' = 64$)	1024	16	-	0.664	3.33
	2048	32	-	0.681	3.42
	4096	64	-	0.701	3.48
	8192	128	-	0.705	3.50

Fig. 2 compares the retrieval accuracies of the proposed method with those of the FV-based method as a function of the number of images in the database. The parameter H of the proposed method was set to 3. Accuracy degradation of the proposed method shows the same trend as that of the FV-based method. Accordingly, we can confirm the superiority of the proposed image representation to the FV representation in terms of accuracy and dimension. In this experiment, for instance, the FV representation needs 8192 variables ($D' = 64$ and $K = 128$), while the proposed representation requires only 1552 variables ($D' = 32, K = 16$ and $H = 3$) to attain higher accuracy.

4 Coding the Mixture of Subspaces Image Representation

In the previous section, we confirmed the efficiency of the proposed image retrieval method primarily in terms of retrieval accuracy. However, for large-scale image retrieval, it is desirable to further reduce the memory costs while retaining the information of each database image, which

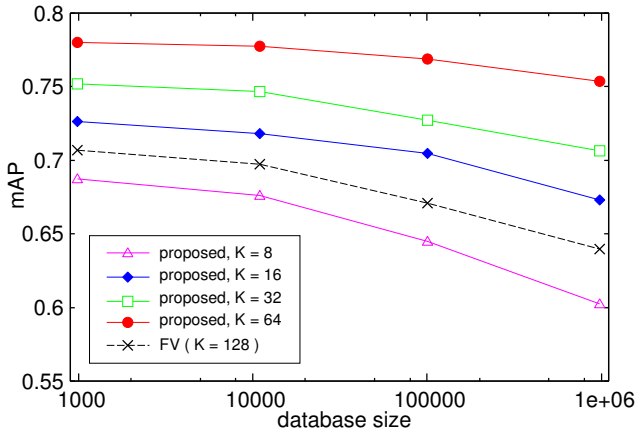


Figure 2: Image retrieval accuracy as a function of the database size.

consists of the $D' \times H$ matrices $\boldsymbol{U}_{i,k}$ and the parameters $\pi_{i,k}$ $(k = 1, 2, \ldots, K)$. Hence, we investigate how to encode the information into compact code (e.g., less than several hundred bytes per database image).

4.1 Encoding Subspaces

The columns of matrix $\boldsymbol{U}_{i,k}$ span the H-dimensional subspace of the local descriptors that belong to the k-th cluster. Suppose that this subspace can be approximated using H basis vectors $\boldsymbol{v}_{k,\ell_1}, \boldsymbol{v}_{k,\ell_2}, \ldots, \boldsymbol{v}_{k,\ell_H}$ chosen from the *basis dictionary*:

$$V_k = \{\boldsymbol{v}_{k,1}, \boldsymbol{v}_{k,2}, \ldots, \boldsymbol{v}_{k,L}\} \tag{17}$$

The basis dictionary is independently constructed from the database. It is assumed that V_k is overcomplete, and each basis vector has unit length: $\|\boldsymbol{v}_{k,\ell}\| = 1$ $(\ell = 1, 2, \ldots, L)$. See Section 4.2 for a description of the basis dictionary construction. Then, we can store only the set of H indices $I_H = \{\ell_1, \ell_2, \ldots, \ell_H\}$ on behalf of the $D' \times H$ floating-point numbers. One method to obtain such an index set is to minimize the following objective function with respect to any I_H:

$$\sum_{n=1}^{N} \left\| \boldsymbol{x}'_n - \sum_{\ell \in I_H} y_{n,\ell} \boldsymbol{v}_\ell \right\|^2 \tag{18}$$

where $y_{n,\ell}$ denotes the coefficient such that $y_{n,\ell} = 0$ for $\ell \notin I_H$. We omit the subscripts i and k and reassign n so that N denotes the number of local descriptors corresponding to $N_{i,k}$ in Section 2. It should be noted that, provided that $N = 1$, this problem becomes a type of sparse coding problem. However, if $N > 1$, it is somewhat different because in this case it is necessary to use the identical subset of the dictionary for all $\boldsymbol{x}'_n$, i.e., the location of the non-zero coefficients must be the same.

In the same as in the case of conventional sparse coding problem, the optimization of (18) is difficult in terms of computational complexity. By considering the well-known matching pursuit algorithm [24], we propose to use the following greedy algorithm that selects the basis vectors successively. Here I_h denotes the set of indices chosen through the h-th iteration, and $\boldsymbol{r}_{n,h}$ denotes the residual of the approximation of $\boldsymbol{x}'_n$ using the h chosen basis vectors.

1. Initialize I_0 and $\boldsymbol{r}_{n,0}$ as $I_0 = \emptyset$ and $\boldsymbol{r}_{n,0} = \boldsymbol{x}'_n$ $(n = 1, 2, \ldots, N)$, respectively.

2. Iterate the following procedure for $h = 1, 2, \ldots, H$:

   (a) Find the index of the optimal basis ℓ_h^* from $\{1, 2, \ldots, L\} - I_{h-1}$:

$$\begin{aligned} \ell_h^* &= \operatorname*{argmin}_{\ell \notin I_{h-1}} \sum_{n=1}^{N} \min_{y \in R} \|\boldsymbol{r}_{n,h-1} - y\boldsymbol{v}_\ell\|^2 \\ &= \operatorname*{argmin}_{\ell \notin I_{h-1}} \sum_{n=1}^{N} \|\boldsymbol{r}_{n,h-1} - (\boldsymbol{v}_\ell^\top \boldsymbol{r}_{n,h-1})\boldsymbol{v}_\ell\|^2 \\ &= \operatorname*{argmax}_{\ell \notin I_{h-1}} \sum_{n=1}^{N} (\boldsymbol{v}_\ell^\top \boldsymbol{r}_{n,h-1})^2. \end{aligned} \tag{19}$$

   (b) Compute I_h and $\boldsymbol{r}_{n,h}$:

$$I_h = I_{h-1} \cup \ell_h^* \tag{20}$$

$$\boldsymbol{r}_{n,h} = \boldsymbol{r}_{n,h-1} - (\boldsymbol{v}_{\ell_h^*}^\top \boldsymbol{r}_{n,h-1})\boldsymbol{v}_{\ell_h^*}. \tag{21}$$

Thus, the H-dimensional subspace of $\{\boldsymbol{x}'_n\}$ can be approximated by the subspace spanned by the chosen basis vectors $\boldsymbol{v}_{\ell_1^*}, \boldsymbol{v}_{\ell_2^*}, \ldots, \boldsymbol{v}_{\ell_H^*}$. Provided that the chosen basis vectors are almost orthogonal to each other, we can replace $\boldsymbol{U}_{i,k}$ with the matrix $\widehat{\boldsymbol{U}}_{i,k} = (\boldsymbol{v}_{\ell_1^*}\ \boldsymbol{v}_{\ell_2^*} \cdots \boldsymbol{v}_{\ell_H^*})$. The columns of $\widehat{\boldsymbol{U}}_{i,k}$ are indexed by integers; therefore, we can encode $\boldsymbol{U}_{i,k}$ compactly.

4.2 Learning the Basis Dictionary

To construct the dictionary of basis vectors V_k, we apply a type of K-subspace clustering method [25, 26, 27] provided that the dimensionality of each subspace is set to 1. However, conventional methods find the nearest subspace for each individual input vector rather than for a set of vectors. Hence, the obtained basis vectors may not necessarily be optimal for our objectives. Thus, we examine the following K-subspace clustering variant. The matrix $\boldsymbol{C}_i$ denotes the correlation matrix of the local descriptors (assigned to a cluster) of the i-th image in the learning set (the subscript k is omitted, cf. (16)).

1. Initialize the dictionary $V = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_L\}$.

2. Find the cluster index $\ell_i^* \in \{1, 2, \ldots, L\}$ for each of the images as follows:

$$\ell_i^* = \operatorname*{argmax}_{\ell} \boldsymbol{v}_\ell^\top \boldsymbol{C}_i \boldsymbol{v}_\ell \tag{22}$$

3. Update the dictionary so that each vector $\boldsymbol{v}_\ell$ is equal to the eigenvector corresponding to the largest eigenvalue of the following matrix ($\ell = 1, 2, \ldots, L$):

$$\widetilde{\boldsymbol{C}}_\ell = \sum_{i:\ell_i^*=\ell} \boldsymbol{C}_i \tag{23}$$

4. Repeat 2) and 3) until the termination condition is met.

If each $\boldsymbol{C}_i$ is obtained from only one descriptor, the algorithm coincides with the K-subspace clustering algorithm for one-dimensional subspace [26]. We apply this algorithm to the distinct learning dataset from the database to construct K basis dictionaries $V_1, V_2, \ldots, V_K$.

4.3 Encoding the Mixture of Subspaces Image Representation

In the proposed image representation, each database image is represented by $\boldsymbol{U}_{i,k}$ and $\pi_{i,k}$ ($k = 1, 2, \ldots, K$). The matrix $\boldsymbol{U}_{i,k}$ is approximated by $\widehat{\boldsymbol{U}}_{i,k}$ as described in the previous section. Each column of $\widehat{\boldsymbol{U}}_{i,k}$ can be represented as an integer from 0 to L (0 for the case of $N_{i,k} = 0$). Hence, the length of the code for $\widehat{\boldsymbol{U}}_{i,k}$ becomes $H \log_2 (L + 1) = HB_V$ bits. On the other hand, $\pi_{i,k}$ is approximated by $\widehat{\pi}_{i,k}$, which is computed from the quantized number of local descriptors $\widehat{N}_{i,k}$ using (15), where

$$\widehat{N}_{i,k} = \left\lfloor (2^{B_N} - 1) N_{i,k} / \max_k N_{i,k} + 0.5 \right\rfloor \tag{24}$$

is a B_N bit integer. In other words, we encode $N_{i,k}$ rather than $\pi_{i,k}$. Consequently, the total code length for one database image becomes $K(HB_V + B_N)$ bits.

Thus, the matching score (14) is approximated as follows:

$$\widehat{S}(\mathcal{I}^{(\mathrm{Q})}, \mathcal{I}_i) = \sum_{k=1}^{K} \left(N_k^{(\mathrm{Q})} \log \widehat{\pi}_{i,k} + \frac{1}{2\sigma_k^2} \sum_{n:z_{n,k}=1} \| \widehat{\boldsymbol{U}}_{i,k}^\top \boldsymbol{x}'_n \|^2 \right). \tag{25}$$

It is worth noting that the term $\| \widehat{\boldsymbol{U}}_{i,k}^\top \boldsymbol{x}'_n \|^2$ is the sum of H values chosen from the L-sorts of values $(\boldsymbol{v}_1^\top \boldsymbol{x}'_n)^2, (\boldsymbol{v}_2^\top \boldsymbol{x}'_n)^2, \ldots, (\boldsymbol{v}_L^\top \boldsymbol{x}'_n)^2$. Hence, the number of computations required to obtain the dot product for each vector $\boldsymbol{x}'_n$ is reduced from $H \times$ (number of database images) to L. Therefore, the proposed encoding method reduces both the memory cost and computational cost for query processing.

Here we summarize the image search method for the encoded mixture of subspaces image representation.

Preparation

In addition to the three steps described in Section 2.3, the following preparation is required.

4. Construct the basis dictionaries $V_1, V_2, \ldots, V_K$ using a distinct set of images from the images that will be stored in the database.

Storing Images in the Database

The storage procedure for image $\mathcal{I}_i$ is as follows.

1. Extract local descriptors from $\mathcal{I}_i$ and determine their cluster assignments.
2. Compute $\widehat{N}_{i,k}$ by (24) ($k = 1, 2, \ldots, K$).
3. Compute the indices for $\widehat{\boldsymbol{U}}_{i,k}$ ($k = 1, 2, \ldots, K$).
4. Store the code corresponding to $\widehat{N}_{i,k}$ and $\widehat{\boldsymbol{U}}_{i,k}$ ($k = 1, 2, \ldots, K$) in the database.

Processing a Query

Given a query $\mathcal{I}^{(\mathrm{Q})}$, the search for the most similar images in the database is performed as follows.

1. Extract local descriptors from $\mathcal{I}^{(\mathrm{Q})}$ and determine their cluster assignments.
2. Compute the matching scores (25) for all database images. Then, output the order of relevance of the images.

5 Experiments on the Proposed Coding Method

We examine the performance of the image retrieval method using the encoded mixture of subspaces image representation and compare the results with those obtained by the asymmetric distance computation (ADC) approach for FVs using the product quantization method [5, 10]; hereafter we refer to this method as the FV+ADC method.

5.1 Datasets and Experimental Procedure

We employed the same datasets and local descriptor extraction methods described in Section 3. The accuracy evaluation measure is also identical to the previous experiments (mAP and KS for the Holidays and the UKB datasets, respectively). The dimensionality D' was also set to 32.

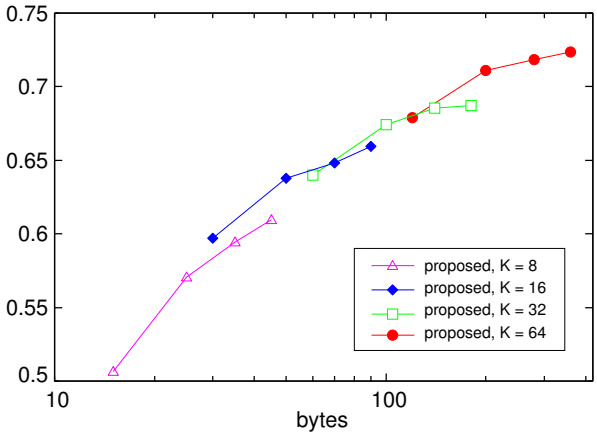
The learning of the basis dictionary was performed using the VOC dataset. For learning, we sampled only the data with correlation matrices $\boldsymbol{C}_i$ of equal or higher rank than $D'/2$. The number of basis vectors L for each dictionary was 1023. The learning was terminated after 10 iterations and was repeated 5 times with different initial conditions, keeping the best partition that gives the largest value of the following objective function.

$$\sum_{\ell=1}^{L} \sum_{i:\ell_i^*=\ell} \boldsymbol{v}_\ell^\top \boldsymbol{C}_i \boldsymbol{v}_\ell \tag{26}$$

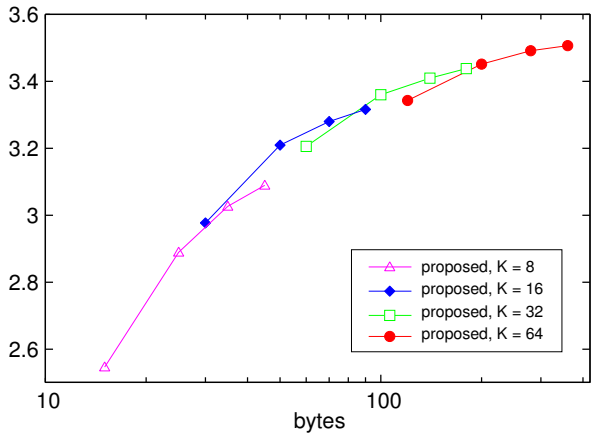
We set L to 1023 so that $B_V = 10$. Although we also tested the case of $B_V = 8$ ($L = 255$), there was marginal difference between these two cases. Therefore we only show results for $B_V = 10$. We have also chosen B_N to be 5. In the preliminary experiment, if $B_N \geq 4$, we observed only a slight decline of accuracy compared to the results without quantization.

For the FV+ADC method, we adopted two parameter settings, ADC 16×8 and 256×10, according to [10]. The number of GMM components are $K = 64$ and 256, respectively. The KD' ($D' = 64$) dimensional Fisher vectors were transformed to 96 and 2048 dimensional vectors by PCA and subsequent random orthogonal projection, respectively. Then these vectors were divided into 16 and 256 subvectors, and each subvector was quantized in 8 and 10 bits, respectively. Parameters of these processes were optimized using the VOC dataset.

In order to evaluate the scalability of the proposed coding method, we also conducted experiments using the same large-scale datasets as described in Section 3.1. Besides the retrieval accuracy we have measured the CPU time required for query processing. These experiments have been performed on a single processor core of a PC with an Intel Core i7 3.4GHz processor and 32GB memory. It should be noted that the query processing was performed by exhaustive search in these experiments, that is, matching scores were computed exhaustively for every image in the database. We did not adopt the non-exhaustive search method for the ADC approach (IVFADC) [5, 10] or any other ANN search techniques for both of the FV+ADC method and the proposed method.



(a) mAP on the Holidays dataset



(b) KS on the UKB dataset

Figure 3: Image retrieval accuracy using the encoded mixture of subspaces image representation.

5.2 Results

Experiments on the Holidays and UKB datasets

Fig. 3 and Table 4 show the experimental results. By comparing the accuracy values in Table 4 with those shown in Table 3, we can confirm that the proposed method can encode the mixture of subspaces image representation into less than several hundred bytes without significantly degrading accuracy. When $K = 64$ and $H = 3$, for instance, the accuracies reduce from 0.780 (Holidays) and 3.66 (UKB) to 0.718 and 3.49, respectively, by encoding the 6208-dimensional image representation into a 280-bytes code. We also suggest that H can be fixed at 2 or 3, as was the case in the previous experiments. Compared to the results obtained by the FV+ADC method, the proposed method outperforms under the condition of longer code length, though this is not the case for shorter code length.

Table 4: Comparison of the proposed method with the FV+ADC method. mAP on the Holidays dataset and KS on the UKB dataset.

		byte	K	H	Holidays	UKB
		15	8	1	0.506	2.55
proposed		25	8	2	0.571	2.89
$D' = 32$		35	8	3	0.595	3.03
$B_V = 10$		70	16	3	0.649	3.28
$B_N = 5$		140	32	3	0.683	3.41
		280	64	3	0.718	3.49
		360	64	4	0.724	3.51
FV+ADC	16×8	16	64	-	0.614	3.18
	256×10	320	256	-	0.675	3.44

Large-Scale Experiments

We evaluate the retrieval accuracy and the retrieval time of the proposed method using the encoded representation on large-scale databases. Fig. 4 compares the retrieval accuracies with those obtained by the FV+ADC method. From the figure, it is observed again that the proposed method is comparable to the state-of-the-art FV+ADC method when each database image is allowed to have several hundreds bytes code. For instance, the accuracies of 320 bytes code obtained by FV+ADC 256×10 are attained by 140 bytes in the case of the proposed method ($K = 32$ and $H = 3$).

On the other hand, Table 5 shows the average of the retrieval times for each of 500 queries from the Holidays dataset. For the proposed method, we focus on two cases: $(K, H) = (32, 3)$ and $(64, 3)$. The former is chosen because it attains similar accuracies to FV+ADC 256×10 with shorter code length, while the latter is chosen because it attains similar code length with higher accuracies (cf. Table 4). In Table 5, the time for computing local descriptors is excluded since it is identical for both methods. The values t_1 correspond to the CPU times required for computation that are independent of the database size. For the proposed method, such computation consists of the processes from the cluster assignment for the local descriptors till the calculation of the values $\sum_{n:z_{n,k}=1}(\boldsymbol{v}_\ell^\top \boldsymbol{x}'_n)^2$ for $k = 1, 2, \ldots, K$ and $\ell = 1, 2, \ldots, L$. For the FV+ADC method, it consists of the processes from the calculation of FV till the distance computation among the subvectors and their prototypes. The values t_2 are the CPU times that are dependent on the database size. In these experiments, they correspond to the processing time for exhaustively computing the matching scores for every image in the database. Hence they are linear with respect to the database size. As one can see from the table, both t_1 and t_2 of the proposed method are smaller than those of the FV+ADC method.

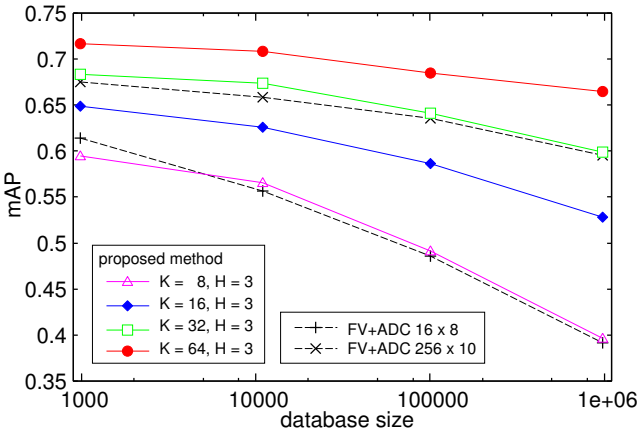


Figure 4: Image retrieval accuracy using the encoded mixture of subspaces image representation as a function of the database size.

Table 5: Retrieval time. t_1 and t_2: processing times (sec) independent and dependent of database size.

DBsize	method	t_1	t_2	$t_1 + t_2$
10991	proposed $K = 32, H = 3$	0.23	0.0017	0.23
	$K = 64, H = 3$	0.25	0.0041	0.26
	FV+ADC 256×10	0.97	0.0055	0.97
100991	proposed $K = 32, H = 3$	0.23	0.015	0.24
	$K = 64, H = 3$	0.25	0.035	0.29
	FV+ADC 256×10	0.97	0.049	1.0
980991	proposed $K = 32, H = 3$	0.23	0.15	0.37
	$K = 64, H = 3$	0.25	0.34	0.59
	FV+ADC 256×10	0.96	0.47	1.4

In particular, t_1 of the FV+ADC method is large since the computation of FV takes high computational cost. In the proposed method, in contrast, almost all computation for query processing consists of basic arithmetic operations. Therefore, the proposed method shows competitive performance compared with the FV+ADC method in terms of retrieval time as well as retrieval accuracy. In these experiments, we did not apply any ANN search technique for reducing the number of candidate images for which to compute the matching scores. Jégou et al. demonstrated that their IVFADC method can scale up to 100 million images [10]. Similar techniques are expected to be effective for the proposed method since our image feature representation has similar structure to their ADC method.

6 Conclusion

We introduced the mixture of subspaces image representation, which obtains both high accuracy and low mem-

ory cost in large-scale image retrieval. This representation outperforms the state-of-the-art FV-based approach in both plain and encoded representation. These results imply the advantage of our approach in which the distribution of local descriptors is modeled for each database image, and the likelihood function of each model is used for matching a query to the database images. However, whether this approach is effective for other problems, such as image classification and object recognition, remains an open question. In our preliminary investigation, a simple image-to-image nearest neighbor classification applying our approach did not produce competitive classification accuracy with the state-of-the-art such as the method using linear support vector machine with FV representation [12]. This observation coincides with the results reported in the case of descriptor-by-descriptor matching based method [28]. One of our future problems is to investigate how to leverage the image classification accuracy of our approach. As suggested by several researchers [29, 30], it might be necessary to introduce some kernelization technique or to develop an feature mapping method so that we can utilize powerful discriminative learning methods such as support vector machine.

References

[1] D. G. Lowe. Distinctive image features from scale-invariant key points. *Int'l. J. Computer Vision*, 60(2):91–110, 2004.

[2] A. Gionis and R. Motwani P. Indyk. Similarity search in high dimensions via hashing. In *Proceedings of the 25 th VLDB Conference*, 1999.

[3] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Communications of the ACM*, 51(1):117–122, 2008.

[4] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *Advances in Neural Information Processing Systems*, pages 1753–1760, 2008.

[5] H. Jégou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 33(1):117–128, 2011.

[6] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. of IEEE Int'l Conf. on Computer Vision (ICCV)*, volume 2, pages 1470–1477, 2003.

[7] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Hamming embedding and weak geometry consistency for large scale image search. In *Proc. of European Conference on Computer Vision (ECCV)*, October 2008.

[8] H. Jégou, M. Douze, and C. Schmid. Improving bag-of-features for large scale image search. *Int'l. J. Computer Vision*, 87(3):316–336, 2010.

[9] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[10] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid. Aggregating local image descriptors into compact codes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 34(9):1704–1716, 2012.

[11] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[12] F. Perronnin, J. Sanchezz, and T. Mensink. Improving the Fisher kernel for large-scale image classification. In *Proc. of European Conference on Computer Vision (ECCV)*, pages 143–156, 2010.

[13] D. Picard and P.H. Gosselin. Improving image similarity with vectors of locally aggregated tensors. In *Proc. IEEE Int'l Conf. Image Processing*, September 2011.

[14] R. Negrel, D. Picard, and P.H. Gosselin. Compact tensor based image representation for similarity search. In *Proc. IEEE Int'l Conf. Image Processing*, September 2012.

[15] M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 21:611–622, 1999.

[16] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[17] The PASCAL Visual Object Classes homepage. `http://pascallin.ecs.soton.ac.uk/challenges/VOC/`.

[18] INRIA Holidays dataset. `http://lear.inrialpes.fr/people/jegou/data.php`.

[19] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2161–2168, June 2006.

[20] Recognition Benchmark Images. `http://www.vis.uky.edu/~stewe/ukbench/`.

[21] Dataset Flickr1M. http://www.multimedia-computing.de/wiki/Flickr1M.

[22] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *Int'l. J. Computer Vision*, 60(1):63–86, 2004.

[23] Affine covariant features. http://www.robots.ox.ac.uk/~vgg/research/affine/.

[24] S. G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, December 1993.

[25] N. Kambhatla and T. K. Leen. Dimension reduction by local principal component analysis. *Neural Computation*, 9(7):1493–1516, 1997.

[26] J. Ho, M.-H. Yang, J. Lim, K.-C. Lee, and D. Kriegman. Clustering appearances of objects under varying illumination conditions. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 11–18, 2003.

[27] R. Vidal. Subspace clustering. *IEEE Signal Processing Magazine*, 28:52–68, 2011.

[28] Oren Boiman, Eli Shechtman, and Michal Irani. In defense of nearest-neighbor based image classification. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[29] T. Tuytelaars, M. Fritz, K. Saenko, and T. Darrell. The nbnn kernel. In *Proc. of IEEE Int'l Conf. on Computer Vision (ICCV)*, 2011.

[30] M. Jain, R. Benmokhtar, P. Gros, and H. Jégou. Hamming embedding similarity-based image classification. In *Proc. of ACM Int'l Conf on Multimedia Retrieval (ICMR)*, 2012.