Discriminative Codeword Selection for Image Representation

Lijun Zhang[1]
zljzju@zju.edu.cn

Chun Chen[1]
chenc@zju.edu.cn

Jiajun Bu[1]
bjj@zju.edu.cn

Zhengguang Chen[1]
cerror@zju.edu.cn

Shulong Tan[1]
laos1984@zju.edu.cn

Xiaofei He[2]
xiaofeihe@cad.zju.edu.cn

[1]Zhejiang Key Laboratory of Service Robot, College of Computer Science, Zhejiang University, Hangzhou, China
[2]State Key Lab of CAD&CG, College of Computer Science, Zhejiang University, Hangzhou, China

ABSTRACT

Bag of features (BoF) representation has attracted an increasing amount of attention in large scale image processing systems. BoF representation treats images as loose collections of local invariant descriptors extracted from them. The visual codebook is generally constructed by using an unsupervised algorithm such as K-means to quantize the local descriptors into clusters. Images are then represented by the frequency histograms of the codewords contained in them. To build a compact and discriminative codebook, codeword selection has become an indispensable tool. However, most of the existing codeword selection algorithms are supervised and the human labeling may be very expensive. In this paper, we consider the problem of unsupervised codeword selection, and propose a novel algorithm called Discriminative Codeword Selection (DCS). Motivated from recent studies on discriminative clustering, the central idea of our proposed algorithm is to select those codewords so that the cluster structure of the image database can be best respected. Specifically, a multi-output linear function is fitted to model the relationship between the data matrix after codeword selection and the indicator matrix. The most discriminative codewords are thus defined as those leading to minimal fitting error. Experiments on image retrieval and clustering have demonstrated the effectiveness of the proposed method.

Categories and Subject Descriptors

I.4.10 [**Image Processing and Computer Vision**]: Image Representation—*Statistical*; I.5.2 [**Pattern Recognition**]: Design Methodology—*Feature evaluation and selection*

MM'10, October 25–29, 2010, Firenze, Italy.

General Terms

Algorithms, Performance, Theory

Keywords

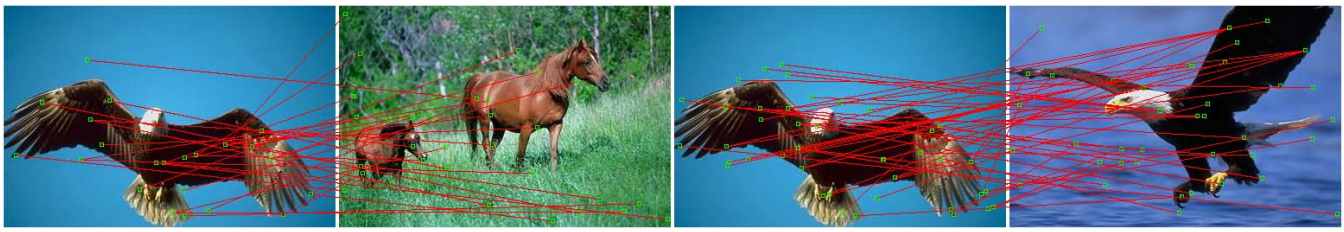
Bag of features, Codeword selection, Feature selection, Image clustering, Image retrieval

1. INTRODUCTION

With the continuous development of digital cameras, storage devices and computer networks, large scale image libraries are available in various application areas. The demands for managing image databases of ever-growing size lead to a great amount of research into Content Based Image Retrieval (CBIR) [5, 7, 32, 38]. In CBIR, images are usually represented by the low level visual features (e.g., color, texture and shape) extracted from them, and relevant images are retrieved based on the similarity of their visual features. CBIR is attractive since it provides one way to access the image database without manual annotation. However, as the visual features are usually high-dimensional and non-sparse, traditional CBIR systems suffer from the scalability problem. Efficient indexing and retrieval schemes remain the key factors for making CBIR a real-world technique.

Motivated by the success of text information retrieval, many existing CBIR systems rely on the bag of features (BoF) representation. The basic idea of BoF is to treat each image as a loose collection of local invariant descriptors (e.g., SIFT [23]) extracted from keypoints [18]. And a visual codebook is constructed by quantizing these local descriptors into clusters using an unsupervised algorithm such as K-means. The cluster centers are called codewords, analogous to the words in text documents. By mapping the descriptors in an image to the codewords, an image is then represented by the frequency histogram over the codebook [16]. In this way, the existing methods for text retrieval, such as inverted indexing, can be naturally applied to CBIR. Recent studies [17, 26, 27, 31, 36, 41] have shown that the BoF model is promising in both performance and scalability for image clustering and retrieval, object recognition, and video event detection.

The quality of the codebook is essential for the BoF-based systems [39, 43]. A small codebook may be lack of discriminative power, because dissimilar descriptors may be mapped to the same codeword. On the other hand, a large code-



(a) Before codeword selection.

(b) After codeword selection by the method proposed in this paper.

Figure 1: An illustration of the role of codeword selection. There are three images selected from the Corel image database: the first and fourth images belong to the *Eagle* category and the third image is the same as the first one. The second image belongs to the *Horse* category. SIFT descriptors are extracted from each image and are indicated by the green squares on the images. Initially, a codebook of size 1000 is constructed by using the K-means clustering algorithm. (a) Using the original codebook, we connect descriptors mapped to the same codeword with red lines. Although the first two images belong to different categories, there are a large number of connected lines between them. (b) After codeword selection by the method proposed in this paper, the number of connected lines between the first two images decreases greatly, whereas there is still a large number of connected lines between the last two images.

book may cause the problem that similar descriptors are mapped to different codewords. Notice that the dimension of the frequency histograms in BoF representation is equal to the size of the codebook. A large number of codewords not only requires more storage and computation resources, but also degrades the performance of many machine learning algorithms, due to the *curse of dimensionality* [13]. To resolve these problems, codeword selection has become an indispensable tool for building a compact and discriminative visual codebook.

Fig. 1 shows an illustrative example from our experimental evaluation. In this example, there are three images selected from the Corel image database: two from the *Eagle* category and one from the *Horse* category. SIFT descriptors are extracted for each image, and their positions are indicated by the green squares drawn on each image. Initially, a codebook of size 1000 is constructed for the BoF representation, and the SIFT descriptors mapped to the same codeword are connected with the red lines. As can be seen from Fig. 1(a), although the first two images belong to different categories, there are a large number of common codewords (connected by red lines) shared by the two images. This is probably due to the interference of the background. Note that, the numbers of extracted SIFT descriptors for the first (also the third), second, and fourth images are 245, 344, and 337, respectively. For the sake of clarity, we only show those SIFT descriptors corresponding to common codewords. Fig. 1(b) shows the results after codeword selection by the method proposed in this paper. As can be seen, a majority of the codewords shared by the first two images are removed, and for the last two images selected from the *Eagle* category, there are still many codewords shared by them. Thus, the new codebook after codeword selection becomes more compact and has more discriminating power.

However, most of the existing codeword selection algorithms [11,18,19,24,29,37] are supervised, which largely limits their applicability in a variety of applications. With the growth of the number of images, providing label information to guide the selection of discriminative codewords becomes infeasible in both time and cost-wise. In this paper, we consider the problem of unsupervised codeword selection for image representation. Inspired by the discriminative clustering framework [1], a novel unsupervised algorithm named Discriminative Codeword Selection (DCS) is proposed. DCS fits a linear function to model the relationship between the data matrix after codeword selection and the indicator matrix. The resulting *fitting error* can be written in closed form, and is dependent on the selected codewords and the indicator matrix. The most discriminative codewords are thus defined as those which lead to minimal *fitting error*. An efficient sequential approach is developed to solve the optimization problem of DCS.

The outline of the paper is as follows. In Section 2, we review the related work in codeword selection and feature selection. Our proposed Discriminative Codeword Selection (DCS) is introduced in Section 3. In Section 4, we develop a computational scheme to solve the optimization problem. Experiments are presented in Section 5. Finally, we provide some concluding remarks and suggestions for future work in Section 6.

Notation. Small letters (e.g. α) are used to denote scalars. Lower-case bold letters (e.g. $\mathbf{w}$) are used to denote column vectors and $\|\cdot\|$ is used to denote the ℓ_2-norm of a vector. Capital letters (e.g. A) are used to denote ma-

trices. We use $\mathrm{Tr}(\cdot)$ to denote the trace of a matrix, and $\|\cdot\|_F$ to denote the Frobenius norm of a matrix. $\mathrm{Diag}(\cdot)$ denotes a diagonal matrix formed from its vector argument, and $\mathrm{diag}(\cdot)$ denotes a column vector consisting of the diagonal elements of its matrix argument. Let $\succeq$ denote the associated generalized inequality of the positive semidefinite cone: $A \succeq B$ means $A - B$ is a positive semidefinite matrix. Script capital letters (e.g. $\mathcal{C}$) are used to denote ordinary sets.

2. RELATED WORK

In this section, we give a brief review of the existing codeword selection algorithms. Since many codeword selection algorithms are based on the feature selection techniques, we begin with a discussion of feature selection.

2.1 Feature Selection

In real applications, dimensionality reduction techniques [9,20,21,30,34] are widely used to deal with the *curse of dimensionality* [13]. Among various methods, feature selection reduces the dimensionality by choosing a subset of revelent features for compact representation [12]. Two types of feature selection techniques have been studied: supervised and unsupervised.

The typical approach for supervised feature selection is to evaluate the correlation between features and labels to determine their relevance. Pearson correlation, Fisher score, Kolmogorov-Smirnov test and Information Gain [10] are several popular methods. More advanced supervised techniques leverage some supervised learning models to select the most useful features. Linear regression based feature selection [35] and Support Vector Machine (SVM) based feature selection [28] have received a lot of attention in recent years. For example, in the Enhanced Biologically Inspired Model [15], SVM and AdaBoost are combined to select the effective features.

Due to the lack of labels, unsupervised feature selection is much harder. Existing unsupervised feature selection techniques can be classified into two categories. The first category exploits the geometrical structure of the data space to guide the selection [3, 14, 25]. The typical algorithms in this category include maximum variance, unsupervised feature selection for PCA [3] and Laplacian score [14]. Maximum variance selects features with the largest variances and unsupervised feature selection for PCA selects a subset of features that can best reconstruct other features. Different from these two methods, Laplacian score [14] selects features that best preserve the local geometrical structure. The second category of unsupervised feature selection techniques aims to maximize some clustering performance [2,6,40]. For example, $Q - \alpha$ [40] measures the cluster coherence by analyzing the spectral properties of the affinity matrix. A remarkable property of this algorithm is that it always yields sparse solutions.

2.2 Codeword Selection

The goal of codeword selection is to remove the redundancy and noise in the codebook, which is usually constructed by using an clustering algorithm. Since each codeword corresponds to one feature in the frequency histogram, feature selection techniques can be used for codeword selection.

In [18], three feature selection methods: mutual information (MI), odds ratio (OR) and linear SVM weights (LSVM) are used to select the most informative codewords. The criterion of information gain (IG) is used in [29] to select the codewords that are most informative about specific location. As more images can be utilized, the retrieval performance of city-scale location recognition is significantly improved. An entropy-based minimum description length (MDL) criterion is proposed in [19] for simultaneous classification and codeword selection.

In [37], a boosting feature selection approach is proposed to select the most discriminative codewords from a multi-resolution codebook. The key idea is to associate each weak classifier with a codeword, and the selection of codeword can be achieved by the selection of the weak classifier. Codeword selection is formulated as a multi-subset search problem in [11], and a novel region selection algorithm is proposed to identify region types that are frequently found in a particular class of scenes but rarely exist in other classes, and also consistently occur together in the same class of scenes. The work in [24] introduces one online codeword selection algorithm based on the dual-gradient descent approach. Side information in the form of pairwise constraints (*must-link* and *must-not link*) is required for this algorithm. A subset of codewords is selected such that the distance computed using them satisfies the given pairwise constraints. The work in [44] considers finding the Descriptive Visual Words (DVWs) and Descriptive Visual Phrases (DVPs) for each image category.

3. DISCRIMINATIVE CODEWORD SELECTION

3.1 Problem Formulation

Let $\mathcal{I} = \{\mathcal{I}_1, \cdots, \mathcal{I}_m\}$ be the given set of m images, which is represented over a visual codebook $\mathcal{C} = \{\mathbf{c}_1, \cdots, \mathbf{c}_n\}$. The goal of codeword selection is to identify a subset of codewords $\mathcal{D} = \{\mathbf{d}_1, \ldots, \mathbf{d}_k\} \subset \mathcal{C}$ that are most informative for describing the image set $\mathcal{I}$.

By mapping the descriptors in an image to the codewords, an image can be represented by the frequency histogram over the codebook $\mathcal{C}$. Then, image $\mathcal{I}_i$ is represented by a vector $\mathbf{x}_i \in \mathbb{R}^n$. Let $F = [\mathbf{x}_1^T; \cdots; \mathbf{x}_m^T] \in \mathbb{R}^{m \times n}$ be the data matrix consisting of all the images, where the i-th row is $\mathbf{x}_i^T$. We denote the j-th column (feature) of F by $\mathbf{f}_j$, and define the feature set $\mathcal{F} = \{\mathbf{f}_1, \ldots, \mathbf{f}_n\}$. Notice that, there exists a one-to-one correspondence between the codebook $\mathcal{C}$ and the feature set $\mathcal{F}$. Thus, the problem of codeword selection can be cast as the problem of selecting the most informative feature subset $\mathcal{H} = \{\mathbf{h}_1, \ldots, \mathbf{h}_k\} \subset \mathcal{F}$.

3.2 The Objective

In this paper, we propose to perform codeword selection under the discriminative clustering framework [1]. The goal is to select the codeword subset $\mathcal{D}$ (or equivalently the feature subset $\mathcal{H}$), such that the performance of discriminative clustering is the best.

Suppose the given images belong to r clusters, we use an indicator matrix $Y \in \{0,1\}^{m \times r}$ to denote the clustering result, where $Y_{ij} = 1$ if $\mathbf{x}_i$ is assigned to the j-th cluster, and $Y_{ij} = 0$ otherwise. Let $H = [\mathbf{h}_1, \ldots, \mathbf{h}_k] \in \mathbb{R}^{m \times k}$ denote the new data matrix containing the selected features only.

We consider fitting a multi-output linear function $f(H) = HW + \mathbf{1}_m\mathbf{b}^T$ to model the relationship between H and Y. In this linear function, $\mathbf{1}_m$ is a m-dimensional vector of all ones, $W \in \mathbb{R}^{k\times r}$ is the coefficient matrix, and $\mathbf{b} \in \mathbb{R}^r$ is the intercept. Following ridge regression [13], fitting this function can be mathematically formulated as

$$\min_{W,\mathbf{b}} \|Y - HW - \mathbf{1}_m\mathbf{b}^T\|_F^2 + \alpha\|W\|_F^2 \tag{1}$$

where $\|\cdot\|_F$ denotes the matrix Frobenius norm, and $\alpha \geq 0$ is the trade-off parameter for the regularizer $\|W\|_F^2$.

Taking the first order partial derivatives of Eq. (1) with respective to W, $\mathbf{b}$ and requiring them to be zero, we get the optimal W^* and $\mathbf{b}^*$:

$$W^* = (H^T\Pi H + \alpha I)^{-1}H^T\Pi Y \tag{2}$$

$$\mathbf{b}^* = \frac{1}{m}\big(Y^T - (W^*)^T H^T\big)\mathbf{1}_m \tag{3}$$

where I is the identity matrix and $\Pi = I - \frac{1}{m}\mathbf{1}_m\mathbf{1}_m^T$ is the centering matrix. To simplify the presentation, we assume that the data has zero mean, so that we have

$$\Pi H = H \tag{4}$$

Substituting the values of W^* and b^* into Eq. (1), we obtain the *fitting error* of the estimated linear function [1]:

$$\begin{aligned}
&J(Y,H)\\
=&\|Y - HW^* - \mathbf{1}_m(\mathbf{b}^*)^T\|_F^2 + \alpha\|W^*\|_F^2\\
=&\left\|Y - HW^* - \frac{\mathbf{1}_m\mathbf{1}_m^T}{m}(Y - HW^*)\right\|_F^2 + \alpha\|W^*\|_F^2\\
=&\|\Pi(Y - HW^*)\|_F^2 + \alpha\|W^*\|_F^2\\
=&\|\Pi\big(I - H(H^TH + \alpha I)^{-1}H^T\big)Y\|_F^2\\
&+ \alpha\|(H^TH + \alpha I)^{-1}H^TY\|_F^2\\
=&\operatorname{Tr}\Big(Y^T\big(\Pi - H(H^TH + \alpha I)^{-1}H^T\big)^2Y\Big)\\
&+ \alpha\operatorname{Tr}\big(Y^TH(H^TH + \alpha I)^{-2}H^TY\big)\\
=&\operatorname{Tr}\Big(Y^T\big(\Pi - H(H^TH + \alpha I)^{-1}H^T\big)Y\Big)
\end{aligned} \tag{5}$$

In the above derivation, we have used the fact that the centering matrix is idempotent, that is, $\Pi = \Pi^k$ for $k = 1, 2, \cdots$. Following the Woodbury-Morrison formula [33], Eq. (5) can be simplified as [1]:

$$\begin{aligned}
&\operatorname{Tr}\Big(Y^T\big(\Pi - H(H^TH + \alpha I)^{-1}H^T\big)Y\Big)\\
=&\operatorname{Tr}\Big(Y^T\Pi\big(I - H(H^TH + \alpha I)^{-1}H^T\big)\Pi Y\Big)\\
=&\operatorname{Tr}\big(Y^T\Pi(I + \frac{1}{\alpha}HH^T)^{-1}\Pi Y\big)\\
=&\alpha\operatorname{Tr}\big(Y^T\Pi(\alpha I + HH^T)^{-1}\Pi Y\big)
\end{aligned} \tag{6}$$

As can be seen, the fitting error $J(Y,H)$ contains Y and H as the variables. Then, it is natural to require that a good indicator matrix Y and the sub-matrix H lead to minimal $J(Y,H)$. In other words, we are looking for a feature subset $\mathcal{H}$, such that if the data is represented by these features, the performance of discriminative clustering is the best.

In the following, we give a mathematical formulation of our codeword selection problem. The constraint that Y is a $m \times r$ indicator matrix is equivalent to the following two constraints:

$$Y \in \{0,1\}^{m\times r},\ Y\mathbf{1}_r = \mathbf{1}_m \tag{7}$$

By introducing a n-dimensional vector $\boldsymbol{\lambda} = [\lambda_1, \cdots, \lambda_n]^T \in \{0,1\}^n$, where λ_i indicates whether or not feature $\mathbf{f}_i$ is chosen, we have

$$H^TH = \sum_{i=1}^{k}\mathbf{h}_i\mathbf{h}_i^T = \sum_{i=1}^{n}\lambda_i\mathbf{f}_i\mathbf{f}_i^T \tag{8}$$

To ensure that k features are selected, the following constraints should be added

$$\mathbf{1}_n^T\boldsymbol{\lambda} = k \tag{9}$$

Then, our codeword selection problem is formally stated below:

Definition 1. Discriminative Codeword Selection (DCS):

$$\begin{aligned}
\min_{Y,\boldsymbol{\lambda}}\ & \operatorname{Tr}\big(Y^T\Pi(\textstyle\sum_{i=1}^n\lambda_i\mathbf{f}_i\mathbf{f}_i^T + \alpha I)^{-1}\Pi Y\big)\\
\text{s.t.}\ & Y \in \{0,1\}^{m\times r},\ Y\mathbf{1}_r = \mathbf{1}_m\\
& \boldsymbol{\lambda} \in \{0,1\}^n,\ \mathbf{1}_n^T\boldsymbol{\lambda} = k
\end{aligned} \tag{10}$$

4. OPTIMIZATION

The problem (10) is difficult to solve due to its combinatorial nature. In this section, we develop a sequential algorithm to find a sub-optimal solution.

Let $(Y^*, \boldsymbol{\lambda}^*)$ be the optimal solution of the problem (10). Initially, we solve the standard discriminative clustering problem [1] with all the features selected. The resulting indicator matrix E can be used as a good estimation of Y^*. Then, by fixing $Y = E$, we solve the problem (10) to find the k most discriminative features.

4.1 Estimation of the Optimal Indicator Matrix

Our goal in this step is to find a good estimation E of the optimal indicator matrix Y^*, which can be used to guide the search of the most discriminative features. Without any prior knowledge, one natural choice is to solve the original discriminative clustering problem. Setting $\boldsymbol{\lambda} = \mathbf{1}_n$, the problem (10) becomes

$$\begin{aligned}
\min_{Y}\ & \operatorname{Tr}\big(Y^T\Pi(\textstyle\sum_{i=1}^n\mathbf{f}_i\mathbf{f}_i^T + \alpha I)^{-1}\Pi Y\big)\\
\text{s.t.}\ & Y \in \{0,1\}^{m\times r},\ Y\mathbf{1}_r = \mathbf{1}_m
\end{aligned} \tag{11}$$

In the following, we adopt the optimization procedure proposed in [1] to solve the above problem. Instead of computing Y, we introduce the variable $M = YY^T$. Using the fact that $\operatorname{Tr}(AB) = \operatorname{Tr}(BA)$, the objective function in the problem (11) becomes:

$$\operatorname{Tr}\left(\Pi(\sum_{i=1}^{n}\mathbf{f}_i\mathbf{f}_i^T + \alpha I)^{-1}\Pi M\right) \tag{12}$$

Following [1], we replace the constraint that M is the product of a $m \times r$ indicator matrix and its transpose with the following constraints:

$$\operatorname{diag}(M) = \mathbf{1}_m,\ M \succeq \frac{1}{r}\mathbf{1}_m\mathbf{1}_m^T,\ M \geq 0 \tag{13}$$

Define $A = \Pi(\sum_{i=1}^n\mathbf{f}_i\mathbf{f}_i^T + \alpha I)^{-1}\Pi$. We have the following optimization problem:

$$\begin{aligned}
\min_{M}\ & \operatorname{Tr}(AM)\\
\text{s.t.}\ & \operatorname{diag}(M) = \mathbf{1}_m,\ M \succeq \tfrac{1}{r}\mathbf{1}_m\mathbf{1}_m^T,\ M \geq 0
\end{aligned} \tag{14}$$

The above problem is a Semidefinite Program (SDP), and can be solved by general purpose interior-point methods [4]. However, directly solving the problem (14) has the complexity of $O(m^7)$, which is too slow for large scale data set. In [1], Bach and Harchaoui have proposed a more efficient approach by solving the following partial dual problem of (14):

$$\begin{array}{ll} \max\limits_{\mathbf{a},\mathbf{b},c,D} & \min\limits_{M} \mathrm{Tr}\left(\big(A + f(\mathbf{a},\mathbf{b},c,D)\big)M\right) - g(\mathbf{a},\mathbf{b},c,D) \\ \text{s.t.} & M \succeq 0,\ \mathrm{Tr}(M) = m \\ & f(\mathbf{a},\mathbf{b},c,D) = \mathrm{Diag}(\mathbf{a}) + \frac{\mathbf{b}\mathbf{b}^T}{2c} - D \\ & g(\mathbf{a},\mathbf{b},c,D) = \mathbf{a}^T\mathbf{1}_m + \mathbf{b}^T\mathbf{1}_m + \frac{cr}{2} \\ & c \geq 0,\ D \geq 0 \end{array} \tag{15}$$

where the variables $\mathbf{a} \in \mathbb{R}^m$, $(\mathbf{b} \in \mathbb{R}^m, c \in \mathbb{R}_+)$ and $D \in \mathbb{R}_+^{m\times m}$ are the dual variables of the constraints $\mathrm{diag}(M) = \mathbf{1}_m$, $M \succeq \frac{1}{r}\mathbf{1}_m\mathbf{1}_m^T$ and $M \geq 0$. The problem (15) can be solved more efficiently due to the fact that $\min\limits_{M} \mathrm{Tr}\left(\big(A + f(\mathbf{a},\mathbf{b},c,D)\big)M\right)$ can be solved simply through an eigenvalue decomposition.

Denote the optimal solution of (15) by M^*. The discrete indicator matrix E are recovered as follows:

1. Computing the first r eigenvectors of M^*, and forming a matrix Z by stacking the eigenvectors in columns;

2. Rescaling the rows of Z to unit norms and then perform K-means to obtain E.

For details, please refer to [1].

4.2 Selecting the Most Discriminative Features

After solving the discriminative clustering problem, we obtain the indicator matrix E. Substituting $Y = E$ into the problem (10), we get the following problem:

$$\begin{array}{ll} \min\limits_{\boldsymbol{\lambda}} & \mathrm{Tr}\left(E^T\Pi(\sum_{i=1}^n \lambda_i\mathbf{f}_i\mathbf{f}_i^T + \alpha I)^{-1}\Pi E\right) \\ \text{s.t.} & \boldsymbol{\lambda} \in \{0,1\}^n,\ \mathbf{1}_n^T\boldsymbol{\lambda} = k \end{array} \tag{16}$$

where the value of λ_i indicates whether or not feature $\mathbf{f}_i$ is chosen as the most discriminative one. This problem is still difficult to solve due to the integer constraint $\boldsymbol{\lambda} \in \{0,1\}^n$.

In the following, we introduce an efficient sequential approach to find the k most informative features. For conciseness, we firstly update E by centering its columns:

$$E \leftarrow \Pi E \tag{17}$$

Suppose a set of t features $\mathcal{H}_t = \{\mathbf{h}_1, \cdots, \mathbf{h}_t\} \subseteq \mathcal{F}$ have been selected as the t most discriminative ones, and define $H_t = [\mathbf{h}_1, \cdots, \mathbf{h}_t]$. The $(t+1)$-th feature $\mathbf{h}_{t+1}$ can be found by solving the following problem:

$$\begin{array}{ll} \min\limits_{\mathbf{f}} & \mathrm{Tr}\left(E^T(H_tH_t^T + \mathbf{f}\mathbf{f}^T + \alpha I)^{-1}E\right) \\ \text{s.t.} & \mathbf{f} \in \mathcal{F} \setminus \mathcal{H}_t \end{array} \tag{18}$$

The most expensive calculation in (18) is the matrix inverse $(H_tH_t^T + \mathbf{f}\mathbf{f}^T + \alpha I)^{-1}$, which need be computed for each $\mathbf{f} \in \mathcal{F} \setminus \mathcal{H}_t$. We use the Woodbury-Morrison formula [33] to avoid directly inverting a matrix. Let $P = (H_tH_t^T + \alpha I)^{-1}$, we have

$$\begin{aligned} &(H_tH_t^T + \mathbf{f}\mathbf{f}^T + \alpha I)^{-1} \\ =&(H_tH_t^T + \alpha I)^{-1} \\ &- \frac{(H_tH_t^T + \alpha I)^{-1}\mathbf{f}\mathbf{f}^T(H_tH_t^T + \alpha I)^{-1}}{1 + \mathbf{f}^T(H_tH_t^T + \alpha I)^{-1}\mathbf{f}} \\ =&P - \frac{P\mathbf{f}\mathbf{f}^TP}{1 + \mathbf{f}^TP\mathbf{f}} \end{aligned} \tag{19}$$

Then, the objective function of (18) can be rewritten as

$$\begin{aligned} &\mathrm{Tr}\left(E^T(H_tH_t^T + \mathbf{f}\mathbf{f}^T + \alpha I)^{-1}E\right) \\ =&\mathrm{Tr}\left(E^T\left(P - \frac{P\mathbf{f}\mathbf{f}^TP}{1 + \mathbf{f}^TP\mathbf{f}}\right)E\right) \\ =&\mathrm{Tr}(E^TPE) - \frac{\mathrm{Tr}(E^TP\mathbf{f}\mathbf{f}^TPE)}{1 + \mathbf{f}^TP\mathbf{f}} \\ =&\mathrm{Tr}(E^TPE) - \frac{\mathbf{f}^TPEE^TP\mathbf{f}}{1 + \mathbf{f}^TP\mathbf{f}} \\ =&\mathrm{Tr}(E^TPE) - \frac{\|E^TP\mathbf{f}\|^2}{1 + \mathbf{f}^TP\mathbf{f}} \end{aligned} \tag{20}$$

Notice that $\mathrm{Tr}(E^THE)$ is a constant when selecting the $(t+1)$-th feature. The optimization problem (18) can be simplified as

$$\begin{array}{ll} \max\limits_{\mathbf{f}} & \|E^TP\mathbf{f}\|^2/(1 + \mathbf{f}^TP\mathbf{f}) \\ \text{s.t.} & \mathbf{f} \in \mathcal{F} \setminus \mathcal{H}_t \end{array} \tag{21}$$

After we have obtained the $(t+1)$-th point $\mathbf{h}_{t+1}$ by solving the problem (21), the matrix P can be updated as

$$P \leftarrow (H_tH_t^T + \mathbf{h}_{t+1}\mathbf{h}_{t+1}^T + \alpha I)^{-1} \tag{22}$$

where the matrix inverse can be computed according to (19).

The above process is repeated until we have selected k features. In the beginning, there are no features selected. Therefore, we set $P = (\alpha I)^{-1} = \frac{1}{\alpha}I$.

5. EXPERIMENTAL RESULTS

In this section, we investigate the use of our proposed codeword selection algorithm for image retrieval and clustering.

5.1 Experimental Setting

Two image databases are used in our experiments. The first one is a subset of 4970 images from the Corel image database. This image subset contains 50 categories and the images are evenly divided among them. We denote this data set by Corel50. The second data set consists of the 10 largest categories, except the BACKGROUND_Google category, in the Caltech-101 object database [8]. This subset contains 3044 images, and is referred to as Caltech10. Fig. 2 shows some sample images from the Corel50 and Caltech10 image data sets.

The SIFT[1] descriptors [23] are extracted from each image. Each descriptor is represented using a 128-dimensional vector. We adopt the fast K-means [27] provided by the Visual Geometry Group[2] to generate the codewords for each image

[1]An implementation can be downloaded from `http://www.vlfeat.org/~vedaldi/code/sift.html`.

[2]An implementation can be downloaded from `http://www.robots.ox.ac.uk/~vgg/software/fastcluster/`.

(a) Samples images from Corel50

(b) Samples images from Caltech10

Figure 2: Sample images from the Corel50 and Caltech10 image data sets.

database. The number of SIFT descriptor extracted from the Corel50 data set is 1,755,935 and 1000 codewords are generated. By assigning the descriptors to the closest codewords, each image in Corel50 is represented by one 1000-dimensional frequency histogram according to the count of each codeword. For the Caltech10 data set, the number of SIFT descriptor is 555,292 and 500 codewords are generated. Thus, each image in Caltech10 is represented by one 500-dimensional frequency histogram.

In the following, several experiments were performed to show the effectiveness of our proposed DCS for unsupervised codeword selection. These experiments include image retrieval and image clustering. The following three codeword selection algorithms are compared:

- **Discriminative Codeword Selection (DCS)**[3]. The unsupervised codeword selection algorithm introduced in this paper.
- Codeword selection based on the $\boldsymbol{Q-\alpha}$ algorithm [40]. $Q-\alpha$ is a unsupervised feature algorithm which selects features to maximize the cluster coherence.
- Codeword selection based on the **Unsupervised Feature Selection using Feature Similarity (FSFS)** [25]. FSFS[4] uses feature similarity for redundancy reduction.

We also provided the results of the **Baseline** method, which uses the original codebook without codeword selection. We compare our proposed approach with $Q-\alpha$ since both of these two approaches aim at discovering the cluster structure of the image database. We compare with FSFS since it has been shown that FSFS is superior to many existing unsupervised feature selection methods such as correlation coefficients and sometimes even better than supervised feature selection methods such as Relief-F [25].

5.2 Image Retrieval

We perform image retrieval experiments on the Corel50 image database. *Precision* is used to evaluate the effectiveness of different codeword selection algorithms. The precision at top N is defined as the ratio of the relevant images presented to the user in the top N ranked images. Each image in the Corel50 database is used as a query image, and the other images are ranked according to the their Euclidean distances to the query image. For Baseline, the Euclidean distances are computed using the original 1000-dimensional frequency histogram. For $Q-\alpha$, FSFS and DCS, a given number ($k = 100, 200, \cdots, 900$) codewords are selected. Then, the part of the original frequency histogram that corresponds to the selected codewords, is used to describe each image. Thus, after codeword selection, the calculation of Euclidean distances will be much faster. The final precision rate is computed by averaging the results over the 4970 queries.

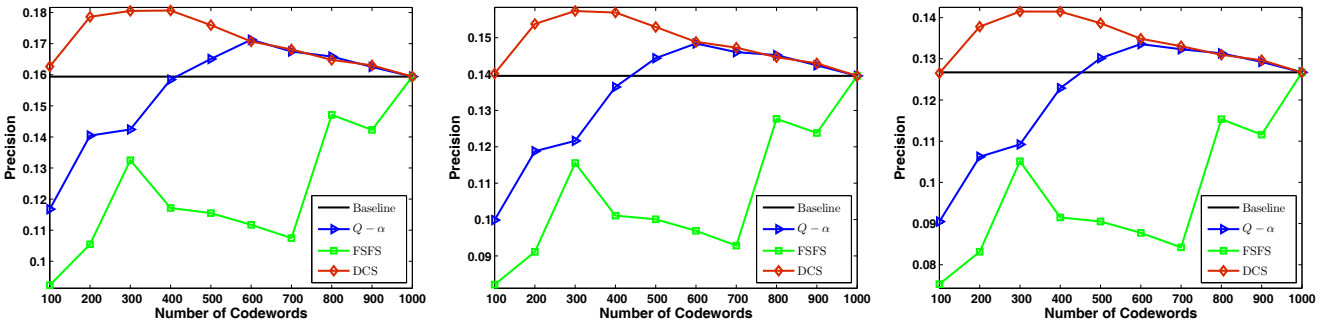
Fig. 3 shows the average precision (at top 20, 40 and 60) versus the number of the selected codewords. As can be seen, our DCS algorithm significantly outperforms the other algorithms in most cases. DCS is very effective in selecting those discriminative visual codewords. With only 100 codewords (selected by DCS), the retrieval performance is almost the same as using all the 1000 codewords. $Q-\alpha$ performs the second best. The accuracy of $Q-\alpha$ is similar to that of DCS when the number of the selected codewords is more than 600. When the number of the selected codewords is more than 400, the accuracy of $Q-\alpha$ is better than Baseline. However, when the number of codewords is smaller than 600, its performance decreases drastically as the number of codewords reduces.

The advantage of DSC and $Q-\alpha$ compared with Baseline validates that codeword selection not only reduces the computational cost, but also has the ability to improve the performance. The performance of FSFS is worse than the Baseline in this experiment. This is probably because FSFS can only remove the redundant codewords, and fails to remove the noisy ones. One common property of $Q-\alpha$ and DCS is that they both aim to maximize the performance of clustering. Thus, the clustering guided codeword selection is more effective for image retrieval. Since DCS is optimized for the discriminative clustering criterion, DCS can select those codewords with higher discriminative power and has higher retrieval performance.

In general, it is appropriate to present 20 images on a screen. Putting more images on a screen may affect the quality of the presented images. Therefore, the precision at top 20 is especially important. Table 1 shows the average precision at top 20 for the 50 categories. $Q-\alpha$, FSFS and DCS are applied to selecting 300 codewords in this table. Considering only the three codewords selection methods, our

[3]The implementation is based on the code for discriminative clustering (`http://www.di.ens.fr/~fbach/diffrac/index.htm`).

[4]An implementation can be downloaded from `http://www.facweb.iitkgp.ernet.in/~pabitra/paper.html`.



(a) Precision at top 20 ranked images. (b) Precision at top 40 ranked images. (c) Precision at top 60 ranked images.

Figure 3: The results of image retrieval on the Corel50 image database. The figures show the average precision versus the number of the selected codewords.

Table 1: Precision (%) at top 20 returns of the four algorithms. For $Q-\alpha$, FSFS and DCS, 300 codewords are selected. The highest precision achieved by the three codeword selection algorithms is in bold for each category.

Category	Baseline	$Q-\alpha$	FSFS	DCS	Category	Baseline	$Q-\alpha$	FSFS	DCS
Antelope	1.80	2.65	3.25	**4.30**	Doll	51.25	46.35	43.55	**46.95**
Antique	7.60	7.00	7.05	**11.10**	drink	22.80	13.20	**16.50**	16.10
aquarelle	8.05	5.90	6.05	**9.55**	Eagle	17.35	13.05	13.65	**14.40**
Balloon	4.75	4.95	3.75	**5.80**	Easter Egg	42.60	**47.00**	24.95	41.70
Beach	3.10	**4.05**	3.65	3.80	elephant	1.30	2.65	3.40	**6.35**
bead	29.50	**47.70**	31.40	33.25	Firework	39.60	25.35	25.70	**49.85**
Bird	4.70	**3.95**	3.40	3.85	Fitness	51.95	34.25	35.10	**47.40**
Bobsled	4.35	4.05	3.30	**5.30**	Flag	36.05	30.45	39.65	**42.80**
Bonsai	6.45	12.85	5.50	**13.10**	flower	8.20	**15.50**	6.95	12.60
Building	5.00	3.75	3.20	**6.15**	Forest	4.50	7.85	3.75	**9.30**
Bus	8.45	11.65	11.15	**20.45**	Fox	1.00	1.10	1.45	**2.85**
Butterfly	2.95	3.50	2.50	**4.40**	Fruit	17.15	11.55	12.25	**20.50**
cactus	1.95	**3.00**	1.85	2.50	Fungus	3.30	**5.30**	3.95	3.75
Canvas	2.40	5.05	2.75	**6.60**	Goat	1.35	2.05	1.05	**2.85**
Cards	71.30	53.95	**74.20**	64.85	Gun	32.05	18.25	26.70	**32.90**
Castle	0.55	2.05	0.85	**2.10**	Horse	6.85	4.40	6.75	**11.35**
Cat	5.90	6.60	9.05	**17.05**	Indoor decorate	10.55	3.30	8.85	**18.30**
Cave	4.75	3.85	4.70	**6.10**	Jewelry	25.80	10.95	13.40	**21.50**
Cell	20.75	16.75	11.55	**23.10**	KungFu	57.20	50.05	**56.05**	50.70
cougr	1.20	**2.50**	**2.50**	2.35	Leopard	3.55	2.60	1.55	**7.55**
Couples	0.70	1.55	0.30	**1.95**	LightHouse	4.00	**5.10**	2.75	**5.10**
Cuisine	6.30	3.00	6.45	**7.40**	Lion	2.45	3.80	3.45	**7.95**
Dinosaur	60.65	54.15	45.65	**68.50**	Lizard	12.80	11.00	10.45	**14.50**
Dish	29.75	**32.25**	19.80	30.45	Marble	11.15	**16.90**	8.95	11.90
Dog	1.35	2.00	2.05	**3.30**	Mask	33.20	27.00	21.85	**40.80**

DCS performs the best on 37 categories, $Q-\alpha$ performs the best on 11 categories, and FSFS performs the best on 4 categories.

5.3 Image Clustering

In this section, we show the experimental results of image clustering. Firstly, we introduce the evaluation metrics used in the experiments.

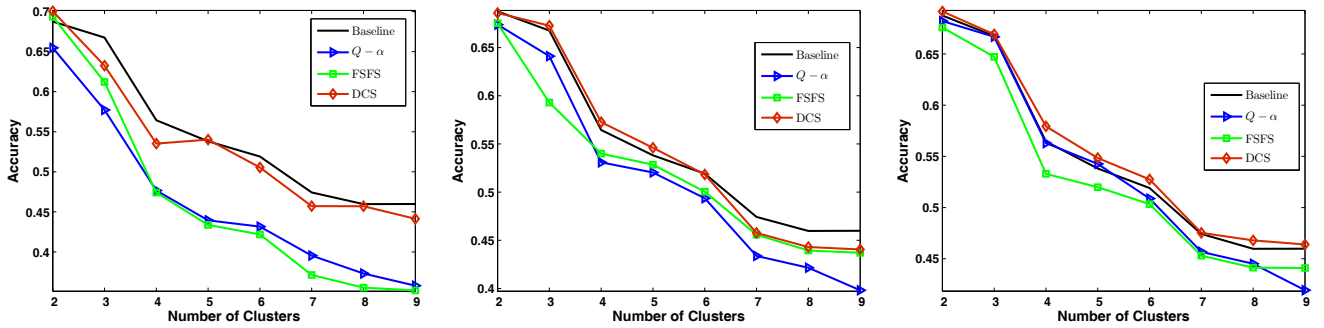
5.3.1 Evaluation Metric

Two metrics, the accuracy (AC) and the normalized mutual information ($\overline{MI}$), are used to measure the clustering performance [42]. Given an image x_i, let p_i and q_i be the obtained cluster label and the label provided by the database, respectively. The AC is defined as follows:

$$AC = \frac{\sum_{i=1}^{m} \delta(q_i, \text{map}(p_i))}{m}, \tag{23}$$

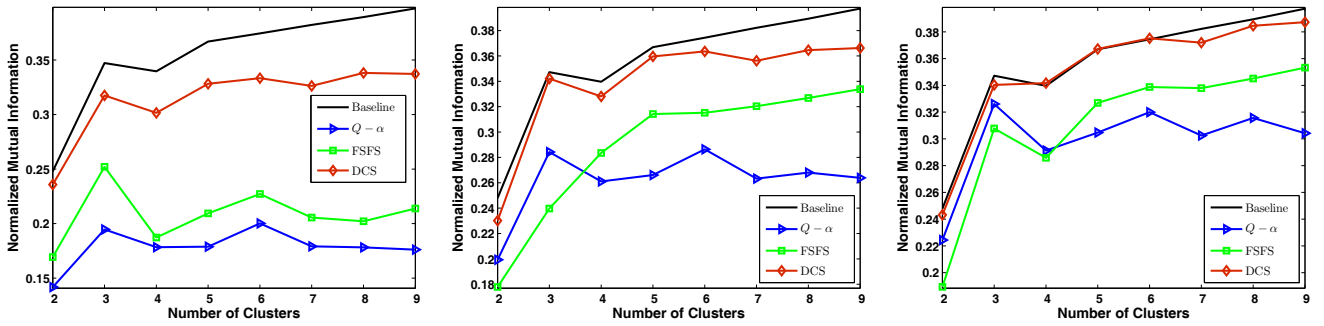
where m is the total number of images, $\delta(x, y)$ is the delta function that equals one if $x = y$ and equals zero otherwise, and $\text{map}(p_i)$ is the permutation mapping function that map each cluster label p_i to the equivalent label from the database. The best mapping can be found by using the Kuhn-Munkres algorithm [22].

Let C denote the set of clusters provided by the database and C' obtained from the clustering algorithm. Their mu-



(a) Accuracy with 100 codewords selected. (b) Accuracy with 200 codewords selected. (c) Accuracy with 300 codewords selected.

Figure 4: Clustering performance measured in terms of accuracy on the Caltech10 image database. The figures show the average accuracy versus the number of clusters.



(a) Normalized mutual information with 100 codewords selected. (b) Normalized mutual information with 200 codewords selected. (c) Normalized mutual information with 300 codewords selected.

Figure 5: Clustering performance measured in terms of normalized mutual information on the Caltech10 image database. The figures show the average normalized mutual information versus the number of clusters.

Table 2: Clustering performance on the Caltech10 image database. For $Q-\alpha$, FSFS and DCS, 200 codewords are used. The highest accuracy (normalized mutual information) achieved by the three codeword selection algorithms is in bold for each number of clusters.

Number of Clusters	Accuracy (%)				Normalized Mutual Information (%)			
	Baseline	$Q-\alpha$	FSFS	DCS	Baseline	$Q-\alpha$	FSFS	DCS
2	68.76	67.38	67.52	**68.59**	24.79	19.93	17.79	**23.01**
3	66.76	64.11	59.29	**67.27**	34.72	28.42	23.97	**34.21**
4	56.45	53.09	54.00	**57.26**	33.97	26.11	28.35	**32.81**
5	53.81	52.04	52.84	**54.60**	36.69	26.61	31.42	**35.96**
6	51.91	49.36	50.04	**51.83**	37.44	28.64	31.52	**36.36**
7	47.43	43.38	45.58	**45.76**	38.22	26.33	32.04	**35.62**
8	45.97	42.15	43.94	**44.30**	38.94	26.81	32.69	**36.46**
9	45.98	39.81	43.70	**44.05**	39.73	26.39	33.38	**36.62**
Avg.	54.63	51.41	52.11	**54.21**	35.56	26.15	28.89	**33.88**

tual information metric $MI(C, C')$ is defined as following:

$$MI(C, C') = \sum_{c_i \in C,\, c'_j \in C'} p(c_i, c'_j).\log_2 \frac{p(c_i, c'_j)}{p(c_i).p(c'_j)}, \quad (24)$$

where $p(c_i)$ and $p(c'_j)$ are the probabilities that an image arbitrarily selected from the database belongs to the clusters c_i and c'_j, respectively, and $p(c_i, c'_j)$ is the joint probability that the arbitrarily selected image belongs to the cluster c_i as well as c'_j at the same time. In our experiments, we use the normalized mutual information $\overline{MI}$ as follows:

$$\overline{MI} = \frac{MI(C, C')}{\max(H(C), H(C'))} \quad (25)$$

where $H(C)$ and $H(C')$ are the entropies of C and C', respectively. It is easy to check that $\overline{MI}$ takes values between 0 and 1.

5.3.2 *Clustering Results*

The image clustering experiments are performed on the Caltech10 image database. For the Baseline algorithm, we

cluster the images using the original 500-dimensional frequency histogram. For $Q-\alpha$, FSFS and DCS, a given number ($k = 100, 200, 300$) codewords are selected. After codeword selection, each image is represented by the part of the original frequency histogram that corresponds to the selected codewords. And the clustering experiments are conducted with this new representation. In the experiments, K-means is used as the clustering algorithm. Because the procedure for solving K-means can only find the local optimum, we ran K-means 10 times with different random starting points and the best result in terms of the objective function of K-means was recorded.

The evaluations were conducted with different number of clusters c, ranging from 2 to 9. At each run of the test, c clusters are randomly selected from the whole database. For each given cluster number c, 10 test runs are conducted, and the average performance was computed over these 10 tests. Fig. 4 shows the average accuracy versus the number of the selected clusters. As can be seen, DCS outperforms the other two codeword selection algorithms in all the cases. With only 200 codewords selected, the accuracy achieved by DCS is better than or comparable to that of Baseline. In terms of accuracy, the performance of $Q-\alpha$ and FSFS is very close. The clustering performance measured by normalized mutual information is shown in Fig. 5. Our DCS still outperforms $Q-\alpha$ and FSFS, and the advantage becomes more obvious. Table 2 shows the detailed clustering results for each algorithm with 200 codewords selected. With all the 500 codewords, the baseline achieves 54.63% in terms of accuracy and 35.56% in terms of normalized mutual information on average. By using only 200 selected codewords, DCS can achieve 54.21% in terms of accuracy (4% relative improvement over FSFS) and 33.88% in terms of normalized mutual information (17.2% relative improvement over FSFS).

6. CONCLUSIONS AND FUTURE WORK

In this paper, a novel unsupervised codeword selection algorithm called Discriminative Codeword Selection (DCS) is proposed. DCS uses the performance of discriminative clustering, a recently proposed unsupervised clustering framework, to guide the selection of the most discriminative codewords. As a result, DCS can select those features with most discriminative power. Image retrieval and clustering experiments on two standard image databases show the effectiveness of our proposed approach.

Because the objective function of DCS contains the indicator matrix as a variable, DCS can be easily extend to incorporate the prior knowledge to the indicator matrix. We will investigate this in our future work. More advanced methods for solving the optimization problem will be studied too.

7. ACKNOWLEDGMENTS

This work was supported by China National Key Technology R&D Program (2008BAH26B00, 2007BAH11B06) and National Natural Science Foundation of China (60875044, 90920303).

8. REFERENCES

[1] F. Bach and Z. Harchaoui. Diffrac: a discriminative and flexible framework for clustering. In *Advances in Neural Information Processing Systems 20*, pages 49–56, 2008.

[2] C. Boutsidis, M. Mahoney, and P. Drineas. Unsupervised feature selection for the k-means clustering problem. In *Advances in Neural Information Processing Systems 22*, pages 153–161. 2009.

[3] C. Boutsidis, M. W. Mahoney, and P. Drineas. Unsupervised feature selection for principal components analysis. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 61–69, 2008.

[4] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[5] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40(2):1–60, 2008.

[6] J. G. Dy and C. E. Brodley. Feature selection for unsupervised learning. *Journal of Machine Learning Research*, 5:845–889, 2004.

[7] J. Fan, Y. Gao, H. Luo, D. A. Keim, and Z. Li. A novel approach to enable semantic and visual image summarization for exploratory image search. In *Proceeding of the 1st ACM international conference on Multimedia information retrieval*, pages 358–365, 2008.

[8] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *Proceedings of the 2004 Computer Vision and Pattern Recognition Workshop on Generative-Model Based Vision*, 2004.

[9] I. K. Fodor. A survey of dimension reduction techniques. Technical report, Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Jun. 2002.

[10] E. Gabrilovich and S. Markovitch. Text categorization with many redundant features: using aggressive feature selection to make svms competitive with c4.5. In *Proceedings of the 21st international conference on Machine learning*, pages 321–328, 2004.

[11] D. Gokalp and S. Aksoy. Scene classification using bag-of-regions representations. In *Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.

[12] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[13] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York, 2009.

[14] X. He, D. Cai, and P. Niyogi. Laplacian score for feature selection. In *Advances in Neural Information Processing Systems 18*, pages 507–514. 2006.

[15] Y. Huang, K. Huang, L. Wang, D. Tao, T. Tan, and X. Li. Enhanced biologically inspired model. In *Proceedings of the 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2008.

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

[17] Y.-G. Jiang, C.-W. Ngo, and J. Yang. Towards optimal bag-of-features for object categorization and

semantic video retrieval. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 494–501, 2007.

[18] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 604–610, 2005.

[19] S. Kim and I. S. Kweon. Simultaneous classification and visualword selection using entropy-based minimum description length. In *Proceedings of the 18th International Conference on Pattern Recognition*, pages 650–653, 2006.

[20] X. Li, S. Lin, S. Yan, and D. Xu. Discriminant locally linear embedding with high-order tensor data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 38(2):342–352, 2008.

[21] H. Liu and Z. Wu. Non-negative matrix factorization with constraints. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, pages 506–511, 2010.

[22] L. Lováz and M. D. Plummer. *Matching Theory*. North-Holland, 1986.

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

[24] P. K. Mallapragada, R. Jin, and A. K. Jain. Online visual vocabulary pruning using pairwise constraints. In *Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010.

[25] P. Mitra, C. A. Murthy, and S. K. Pal. Unsupervised feature selection using feature similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):301–312, 2002.

[26] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2161–2168, 2006.

[27] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.

[28] A. Rakotomamonjy. Variable selection using svm based criteria. *Journal of Machine Learning Research*, 3:1357–1370, 2003.

[29] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. In *Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.

[30] S. Si, D. Tao, and B. Geng. Bregman divergence-based regularization for transfer subspace learning. *IEEE Transactions on Knowledge and Data Engineering*, 22:929–942, 2010.

[31] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proceedings of the Ninth IEEE International Conference on Computer Vision*, pages 1470–1477, 2003.

[32] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, 2000.

[33] G. Strang. *Introduction to Linear Algebra, 3rd Edition*. Wellesley-Cambridge Press, 2003.

[34] D. Tao, X. Li, X. Wu, and S. J. Maybank. Geometric mean for subspace selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:260–274, 2009.

[35] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[36] F. Wang, Y.-G. Jiang, and C.-W. Ngo. Video event detection using motion relativity and visual relatedness. In *Proceeding of the 16th ACM international conference on Multimedia*, pages 239–248, 2008.

[37] L. Wang. Toward a discriminative codebook: Codeword selection across multi-resolution. In *Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.

[38] M. Wang, B. Liu, and X.-S. Hua. Accessible image search. In *Proceedings of the seventeen ACM international conference on Multimedia*, pages 291–300, 2009.

[39] J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. In *Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 1800–1807, 2005.

[40] L. Wolf and A. Shashua. Feature selection for unsupervised and supervised inference: The emergence of sparsity in a weight-based approach. *Journal of Machine Learning Research*, 6:1855–1887, 2005.

[41] Z. Wu, Q. Ke, M. Isard, and J. Sun. Bundling features for large scale partial-duplicate web image search. In *Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 25–32, 2009.

[42] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273, 2003.

[43] J. Yang, Y.-G. Jiang, A. G. Hauptmann, and C.-W. Ngo. Evaluating bag-of-visual-words representations in scene classification. In *Proceedings of the international workshop on multimedia information retrieval*, pages 197–206, 2007.

[44] S. Zhang, Q. Tian, G. Hua, Q. Huang, and S. Li. Descriptive visual words and visual phrases for image applications. In *Proceedings of the 17th ACM international conference on Multimedia*, pages 75–84, 2009.