A 2-D Histogram Representation of Images for Pooling

Xinnan YU and Yu-Jin ZHANG

Department of Electronic Engineering, Tsinghua University, Beijing, 100084, China

ABSTRACT

Designing a suitable image representation is one of the most fundamental issues of computer vision. There are three steps in the popular Bag of Words based image representation: feature extraction, coding and pooling. In the final step, current methods make an $M \times K$ encoded feature matrix degraded to a K-dimensional vector (histogram), where M is the number of features, and K is the size of the codebook: information is lost dramatically here. In this paper, a novel pooling method, based on 2-D histogram representation, is proposed to retain more information from the encoded image features. This pooling method can be easily incorporated into state-of-the-art computer vision system frameworks. Experiments show that our approach improves current pooling methods, and can achieve satisfactory performance of image classification and image reranking even when using a small codebook and costless linear SVM.

Keywords: Bag of Words, pooling, image classification, image reranking, 2-D histogram

1. INTRODUCTION

Designing efficient image representation is one of the most fundamental issues of computer vision. One popular framework of image representing is called Bag of Words, which represents images as an orderless collection of several kinds of features. This method is first introduced as an analogy of Bag of Words model in text retrieval.[1]

There are three steps in Bag of Words based image representation. The first one is feature extraction: extracting local or global descriptors from grey level images. The descriptors are selected at interest point locations, or in a dense sampling manner. Local features such as SIFT[2] and HOG[3] are widely used in state-of-the-art computer vision systems. After this step, each image is represented by a set of raw feature descriptors:

$$\boldsymbol{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_M]^{\mathrm{T}} \in \boldsymbol{R}^{M \times D}, \tag{1}$$

where M is the number of feature descriptors, and D is the dimension of the feature descriptor.

Although raw feature descriptor contains the most information from the image, it is too large for efficient processing; also its discrimination ability is not satisfactory. The second step, which is called coding, transforms raw image features into more efficient representations. In a popular approach, feature descriptors are first clustered to build feature codebook (dictionary):

$$\boldsymbol{V} = [\boldsymbol{v}_1, ..., \boldsymbol{v}_K]^{\mathrm{T}} \in \boldsymbol{R}^{K \times D}, \tag{2}$$

where $\boldsymbol{v}_1, ..., \boldsymbol{v}_K$ are the codewords. Then each descriptor is quantized into one or several codewords to form matrix $\boldsymbol{A} \in \boldsymbol{R}^{M \times K}$, in which $\alpha_{i,j}$ is the "belongingness" of the i-th descriptor to the j-th codeword. In traditional Bag of Words method,[1] hard quantization is used, i.e.

$$\alpha_{i,j} = \begin{cases} 1, & \text{if} \quad j = \arg\min\limits_{k} ||\boldsymbol{x}_i - \boldsymbol{v}_k||_2^2 \\ 0, & \text{otherwise} \end{cases}, \tag{3}$$

Further author information:
Xinnan YU: E-mail: yuxinnan@gmail.com
Yu-Jin ZHANG: E-mail: zhang-yj@mail.tsinghua.edu.cn
This work was partially supported by National Nature Science Foundation (NNSF: 60872084).

Image Processing: Machine Vision Applications IV, edited by David Fofi, Philip R. Bingham, Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 7877, 787706

where $\boldsymbol{V} = [\boldsymbol{v}_1, ..., \boldsymbol{v}_K]^T$ is the set of cluster centers. Each row of $\boldsymbol{A}$ has only one non-zero value 1. An example of $\boldsymbol{A}$ in traditional Bag of Words model is:

$$\boldsymbol{A}_1 = \left\{ \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{matrix} \right\}. \tag{4}$$

Here five descriptors are quantized into four cluster centers. A more general approach is soft quantization,[4] which uses a linear combination of multiple visual words to approximate the raw descriptors.

$$\alpha_{i,j} = \frac{\exp(-\beta \left\| \boldsymbol{x}_i - \boldsymbol{v}_j \right\|_2^2)}{\sum_{k=1}^{K} \exp(-\beta \left\| \boldsymbol{x}_i - \boldsymbol{v}_k \right\|_2^2)}, \tag{5}$$

where β is the soft assignments factor. Another method is sparse coding,[5] in which $\boldsymbol{A} \in R^{M \times K}$ is defined by the following minimization problem:

$$\min_{\boldsymbol{A},\boldsymbol{V}} \sum_{i=1}^{M} ||\boldsymbol{x}_i - \boldsymbol{\alpha}_i \boldsymbol{V}||^2 + \lambda |\boldsymbol{\alpha}_i|, \quad s.t. \quad ||\boldsymbol{v}_k|| \leq 1, \quad \forall k = 1, 2, ..., K, \tag{6}$$

where λ is the sparse regularization, ensuring most values of $\boldsymbol{A}$ to be zeros. An example of $\boldsymbol{A}$ in sparse coding is:

$$\boldsymbol{A}_2 = \left\{ \begin{matrix} 0.3 & -0.1 & 0.2 & 0 \\ 0 & -0.2 & 0 & 0.5 \\ 0.4 & 0 & 0.3 & 0 \\ 0 & 0 & 0.1 & 0.5 \\ 0.5 & 0.4 & 0 & 0 \end{matrix} \right\}. \tag{7}$$

Here five descriptors are represented as sparse linear combinations of four codewords.

The third step is pooling: summarizing the encoded features across each image to form the final image representation. The objective of pooling is to achieve invariance to image transformation, more compact representation, and better robustness to noise and clutter.[6] Current pooling methods make $\boldsymbol{A} \in \boldsymbol{R}^{M \times K}$ degraded to a K-dimensional vector (histogram) $\boldsymbol{H} \in \boldsymbol{R}^K$. Two general configurations of pooling are average pooling and max pooling. In average pooling, $\boldsymbol{H} \in \boldsymbol{R}^K$ is the sum (or mean) values of columns of $\boldsymbol{A} \in \boldsymbol{R}^{M \times K}$; in max pooling, $\boldsymbol{H} \in \boldsymbol{R}^K$ is the max value of columns of $\boldsymbol{A} \in \boldsymbol{R}^{M \times K}$. Figure 1 and Figure 2 are example $\boldsymbol{H}$s by average pooling and max pooling for $\boldsymbol{A}_1$ in equation 4, respectively. And Figure 3 and Figure 4 are example $\boldsymbol{H}$s by average pooling and max pooling for $\boldsymbol{A}_2$ in equation 7, respectively. From the general pooling process above, we can know that information is lost dramatically here.*

The motivation of our work is to design methods which can retain more information in pooling, and thus improve the performance of computer vision systems. The intuition is that other than pooling $\boldsymbol{A} \in \boldsymbol{R}^{M \times K}$ into a K-dimensional vector $\boldsymbol{H} \in \boldsymbol{R}^K$, we can do a trade off between the efficiency and performance: to design methods to pool $\boldsymbol{A} \in \boldsymbol{R}^{M \times K}$ into a larger vector in order to retain more information. In this paper, a novel pooling method, 2-D histogram representation based pooling, is proposed to achieve the goal. 2-D histogram representation can be used to pool image features encoded by all kinds of image coding techniques such as hard quantization, soft quantization and sparse coding, and it can be easily incorporated into state-of-the-art computer vision system frameworks. Experiments show that our approach can improve current pooling methods, and can achieve satisfactory performance of image classification and image reranking even when using a small codebook and costless linear SVM.

*When hard quantization (traditional Bag of Words model) and average pooling is used, information is not lost in the pooling step, because $\boldsymbol{A} \in \boldsymbol{R}^{M \times K}$ can be fully reconstructed by $\boldsymbol{H} \in \boldsymbol{R}^K$ when the order of the encoded features is not important. However, hard quantization is a highly information-lost process. Section 2 will show that after a slight modification, our later proposed method is also applicable to hard quantization.

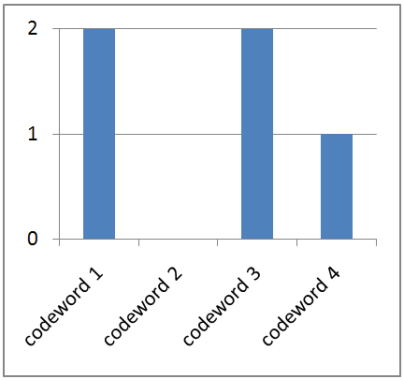

Figure 1. $\boldsymbol{H}$ by average pooling of $\boldsymbol{A}_1$ (traditional Bag of Words model). The distribution of descriptors is described as a 1-D histogram.

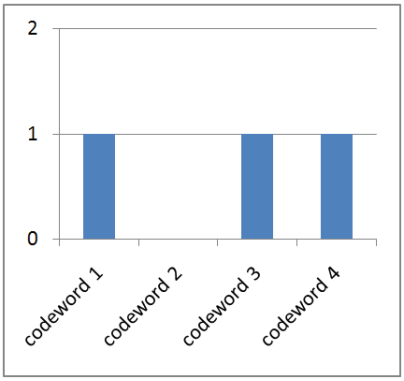

Figure 2. $\boldsymbol{H}$ by max pooling of $\boldsymbol{A}_1$ (traditional Bag of Words model). In contrast to average pooling, the acquired $\boldsymbol{H}$ is a binary "histogram".

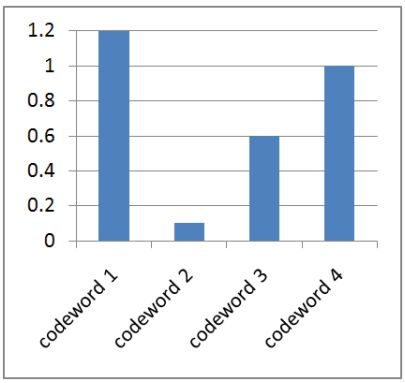

Figure 3. $\boldsymbol{H}$ by average pooling of $\boldsymbol{A}_2$ (sparse coding). $\boldsymbol{H}$ is the sum values of columns of $\boldsymbol{A}_2$.

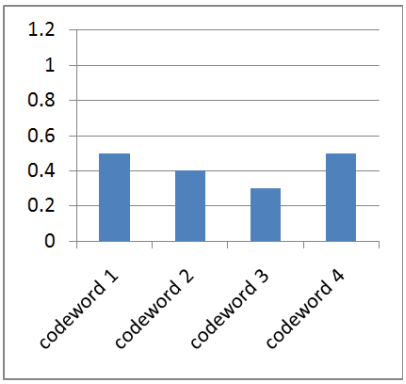

Figure 4. $\boldsymbol{H}$ by max pooling of $\boldsymbol{A}_2$ (sparse coding). $\boldsymbol{H}$ is the max values of columns of $\boldsymbol{A}_2$.

The rest of the paper is organized as follows. In section 2, related works are reviewed. The 2-D histogram representation algorithm and its analysis are presented in Section 3. Experiment results are evaluated in Section 4. Some discussions are provided in Section 5. Concluding remarks and future research direction are provided in Section 6.

2. RELATED WORKS

Besides the effort of designing feature descriptors and coding methods, pooling is relatively less studied. Several approaches have shown that a change in the strategy of pooling can lead to much better performance. Yang et al.[5] has shown that under sparse coding, max pooling demonstrates superior performance than average pooling. Y. Boureau et al.[7] has studied the performance of max pooling and average pooling under various experiment setups, and has provided a simple analysis of max and average pooling. In a more recent work of them,[6] detailed theoretical analysis of max pooling and average pooling are provided by modeling binary features as Bernoulli distribution and sparse codes as Exponential distribution. Although all the above works show that pooling can be very important in image representation, none of them try to design methods to get better pooling by extending the pooling vector $\boldsymbol{H}$.

The ultimate goal of improving pooling method is to get efficient and discriminative representation of images, and thus improve the discrimination of different classes of images. In other words, given some images, and each of them has been represented by sets of encoded features, we want to get better measurement of the similarity between the images. From this perspective, one research direction is to design better kernels between sets of features. Work in this category include but not limited to kernels on attributed pointsets,[8] which extends sum matching kernel[9] by spatial information; pyramid matching kernel[10] which matches sets of image features by

multi-resolution histograms (it can be described as a multi-resolution average pooling), Bhattacharyya kernel[11] and Fisher kernel,[12–16] which analysis the similarity by a estimated probability distribution of the features. Beyond Bag of Words model, these works have provided another view in image representation.†

Some related methods which fall into the category of Bag of Words model are EMK (efficient matching kernel),[17] MiniBOF,[18] spatial pyramid matching,[19] and spatial quadratic codebook.[20,21] In the first approach,[17] local features are first mapped to a low-dimensional space, and then average pooled across images. In the second approach,[18] the author builds a set of sparse projections of max-pooled Bag of Words vector in order to achieve index efficiency. Different from these two works, our method is to extend, not to compress, the feature vector, and 2-D histogram is used to address the problem directly in pooling step, not before or after pooling. In the third approach,[19] encoded image features are pooled on different spatial grads to incorporate spatial information in the final image representation. In the fourth and fifth approaches,[20,21] image features which are represented by Bag of Words model are paired based on spatial information, yielding higher-order features. Thus, a quadratic codebook, i.e. a 2-D histogram in spatial domain, can be used to describe the feature pairs. While also trying to retain more information in pooling, the information source in the proposed 2-D histogram is from the feature itself other than spatial positions of the feature. Therefore, 2-D histogram pooling can be considered as orthogonal to the last three approaches.

Our work is related to a recent work called super vector,[22] which also tries to design better pooling method by retaining more information from the encoded features. Different from their approach which is probability based, our work is a simple extension of histogram in traditional Bag of Word model which can achieve both efficiency and the simple elegance of Bag of Words model.

3. 2-D HISTOGRAM REPRESENTATION FOR POOLING

In this section, a general framework of 2-D histogram representation for pooling is stated and analyzed. Then detailed algorithms extended from traditional Bag of Words model and sparse coding are provided. The algorithms can be easily modified to fit other coding techniques.

3.1 Framework of 2-D Histogram Representation in Pooling

In contrast to current pooling methods, which make $\boldsymbol{A} \in \boldsymbol{R}^{M \times K}$ degraded to a K-dimensional vector $\boldsymbol{H} \in \boldsymbol{R}^{K}$, 2-D representation pools the encoded features into matrix $\tilde{\boldsymbol{H}} \in \boldsymbol{R}^{K \times K}$. A general framework of 2-D histogram representation can be described in Algorithm 1.

According to the algorithm, $\tilde{\boldsymbol{H}}$ is a weighted distribution of $\boldsymbol{A}$ according to two specific dimensions. That is why we call our approach 2-D histogram representation. f_1 and f_2 are functions to choose the specific dimensions of $\boldsymbol{\alpha}_i$, $i = 1, 2..., M$. A simple example for selecting f_1 and f_2 is to choose the first and second largest numbers in $\boldsymbol{\alpha}_i$, respectively. g determines the value added to the specific bins of the 2-D histogram. If average pooling is used, $g(\alpha_{i,x}, \alpha_{i,y})$ will be added one by one; if max pooling is used, only the largest $g(\alpha_{i,x}, \alpha_{i,y})$ will be used. An example of g is the mean function: $g(\alpha_{i,x}, \alpha_{i,y}) = (\alpha_{i,x} + \alpha_{i,y})/2$.

In general, computing f_1 and f_2 pays a computational complexity $O(K)$. The overall complexity of pooling in Algorithm 1 is $O(M \times K)$, which is the same with 1-D histogram pooling. The problem is that 2-D histogram is not space efficient: the additional dimension will increase the space complexity from $O(K)$ to $O(K^2)$ under the same coding configuration. The increased space complexity would make the post processes time consuming. However, in Section 3, experiments show that this scheme can be utilized in small codebooks without much sacrifice of the performance.

Specific algorithms will be stated in the following two subsections with the configurations of Bag of Words model and sparse coding.

†A review of image representation method other than Bag of Words model is beyond the topic of this article.

Algorithm 1 : Framework of 2-D Histogram Representation for Pooling

for each image **do**
  extracting image features $\boldsymbol{x}_1, \boldsymbol{x}_2, ...\boldsymbol{x}_M$
  coding to get $\boldsymbol{A} \in \boldsymbol{R}^{M \times K}$
  initialize $\tilde{\boldsymbol{H}} = \text{zeros}(K, K)$
  for $i = 1$ to M **do**
    find $\alpha_{i,x}$ and $\alpha_{i,y}$, s.t. $\alpha_{i,x} = f_1(\boldsymbol{\alpha}_i)$ and $\alpha_{i,y} = f_2(\boldsymbol{\alpha}_i)$
    if using average pooling **then**
      $\tilde{H}_{x,y} = \tilde{H}_{x,y} + g(\alpha_{i,x}, \alpha_{i,y})$
    else
      $\tilde{H}_{x,y} = \max(\tilde{H}_{x,y}, g(\alpha_{i,x}, \alpha_{i,y}))$
    end if
  end for
end for

3.2 Algorithm for Bag of Words Model

In the coding step of traditional Bag of Words model, each descriptor is assigned as a single cluster center. In order to obtain 2-D information from the encoded features, the coding step is slightly modified as shown in Algorithm 2. Each descriptor is assigned as a combination of two codewords. Simpler than soft quantization, there is no weights in the assignments, and a descriptor is assigned as only two codewords instead of all of them. Equation 8 shows an example of modified encoded feature matrix:

$$\tilde{\boldsymbol{A}}_1 = \left\{ \begin{matrix} 1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ -1 & 0 & 1 & 0 \end{matrix} \right\}. \tag{8}$$

Algorithm 2 Modified Coding Step in Bag of Words Model

for each image **do**
  extract image features $\boldsymbol{x}_1, \boldsymbol{x}_2, ...\boldsymbol{x}_M$
  initialize $\boldsymbol{A}$ =zeros (M, K)
  for $i = 1$ to M **do**
    find the closest cluster center $\boldsymbol{v}_{j_1}$ to $\boldsymbol{x}_i$
    $\alpha_{i,j_1} = 1$
    find the second closest cluster center $\boldsymbol{v}_{j_2}$ to $\boldsymbol{x}_i$
    $\alpha_{i,j_2} = -1$
  end for
end for

Algorithm 3 is 2-D representation in pooling for Bag of Words model. We simply design function g as a constant value 1. Figure 5 and Figure 6 are the acquired $\tilde{\boldsymbol{H}}$s of $\tilde{\boldsymbol{A}}_1$ according to Algorithm 3 with average pooling and max pooling, respectively. Compared to Figure 1 and Figure 2, 2-D histogram contains much more information, and the pooling results in traditional Bag-of-Words model can be fully reconstructed from 2-D histogram representation.

3.3 Algorithm for Sparse Coding

Algorithm 4 is a simple 2-D histogram pooling process for sparse coding. Because $\boldsymbol{A}$ from sparse coding is not a non-negative matrix, it is simply modified as its absolute value:$\boldsymbol{A} = \text{abs}(\boldsymbol{A})$. Equation 9 shows an example of

Algorithm 3 : Algorithm for Bag of Words Model

for each image **do**
  code according to modified Bag of Word method to get $\boldsymbol{A} \in \boldsymbol{R}^{M \times K}$
  initialize $\tilde{\boldsymbol{H}} = \text{zeros}(K, K)$
  for $i = 1$ to M **do**
    find $\alpha_{i,x}$ and $\alpha_{i,y}$, s.t. $\alpha_{i,x} = 1$ and $\alpha_{i,y} = -1$
    if using average pooling **then**
      $\tilde{H}_{x,y} = \tilde{H}_{x,y} + 1$
    else
      $\tilde{H}_{x,y} = 1$
    end if
  end for
end for

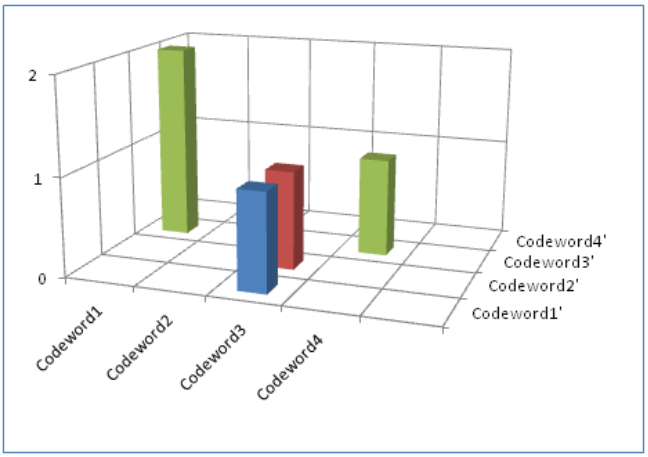

Figure 5. $\tilde{\boldsymbol{H}}$ by average pooling of $\tilde{\boldsymbol{A}}_1$ (Bag of Words)

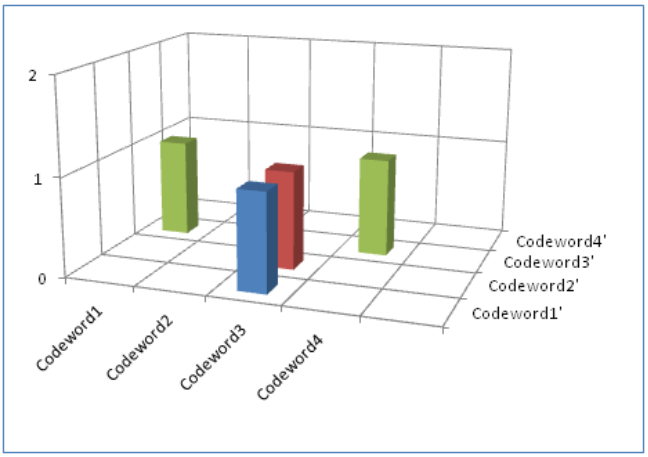

Figure 6. $\tilde{\boldsymbol{H}}$ by max pooling of $\tilde{\boldsymbol{A}}_1$ (Bag of Words)

absolute value of encoded feature matrix:

$$\tilde{\boldsymbol{A}}_2 = \begin{Bmatrix} 0.3 & 0.1 & 0.2 & 0 \\ 0 & 0.2 & 0 & 0.5 \\ 0.4 & 0 & 0.3 & 0 \\ 0 & 0 & 0.1 & 0.5 \\ 0.5 & 0.4 & 0 & 0 \end{Bmatrix}. \tag{9}$$

Figure 7 and Figure 8 are $\tilde{\boldsymbol{H}}$s of $\tilde{\boldsymbol{A}}_2$ according to Algorithm 4 with average pooling and max pooling, respectively. Compared to Figure 3 and Figure 4, 2-D histogram contains much more information.

4. EXPERIMENTS

In this section, our proposed method is tested in two applications, image classification and image reranking. For both classification and reranking, 15 scene categories[23] and Caltech101[24] are used as the test benchmarks. Results from 1-D histogram and 2-D histogram with a small codebook size (16 or 8) are evaluated. And results from 1-D histogram with a large codebook size (256 or 64) are provided for reference.

4.1 Image Classification

As spatial pyramid[19] is a successful method that has been widely used in scene and object classification, a 3-level spatial pyramid is used. For 15 scene categories dataset, 100 images in each class are used for training and the rest for testing. For Caltech101 dataset, 30 images in each class are used for training and the rest (with the limit of 50) for testing. SVM based on 1-vs-all rule is used. In sparse coding, no matter whether the method we use

Algorithm 4 : Algorithm for Sparse Coding

for each image **do**
  extract image features $\boldsymbol{x}_1, \boldsymbol{x}_2, ...\boldsymbol{x}_M$
  do sparse coding to get $\boldsymbol{A} \in \boldsymbol{R}^{M \times K}$
  $\boldsymbol{A} = \text{abs}(\boldsymbol{A})$
  initialize $\tilde{\boldsymbol{H}}$ =zeros(K, K)
  for $i = 1$ to M **do**
    find the largest value of $\boldsymbol{\alpha}_i$: $\alpha_{i,x}$
    find the second largest value of $\boldsymbol{\alpha}_i$: $\alpha_{i,y}$
    if using average pooling **then**
      $\tilde{H}_{x,y} = \tilde{H}_{x,y} + (\alpha_{i,x} + \alpha_{i,y})/2$
    else
      $\tilde{H}_{x,y} = \max(\tilde{H}_{x,y}, (\alpha_{i,x} + \alpha_{i,y})/2)$
    end if
  end for
end for

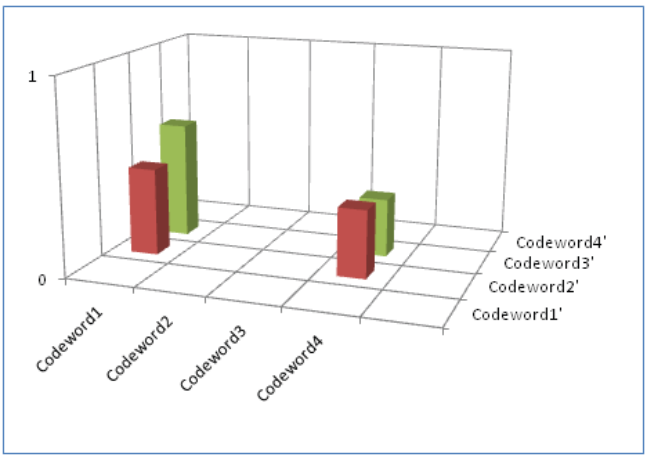

Figure 7. $\tilde{\boldsymbol{H}}$ by average pooling of $\tilde{\boldsymbol{A}}_2$ (sparse coding)

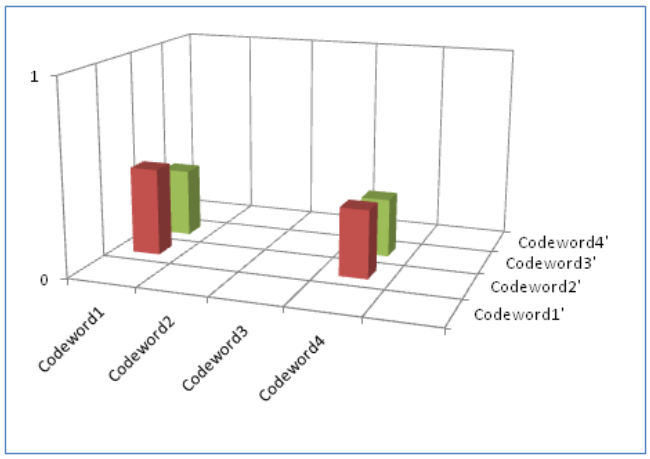

Figure 8. $\tilde{\boldsymbol{H}}$ by max pooling of $\tilde{\boldsymbol{A}}_2$ (sparse coding)

is 1-D histogram or 2-D histogram, $\boldsymbol{A}$ is modified as its absolute value. The result is the mean recognition rate for all the classes. For other experimental settings, we follow the work of S. Lazebnik et al.[19] and J. Yang et al.[5] Each experiment is performed 5 times with randomly selected training and testing images in order to get reliable results. The reported recognition rate is presented by the mean and standard deviation of the experiments.

Table 1 and Table 2 are the experiments results of 15 scene categories and Caltech101 datasets, respectively. We have tested our method in various experiment settings, i.e. average/max pooling, histogram intersection kernel (HIK)/linear kernel. From Table 1 and Table 2, with a codebook size of 16, the proposed 2-D histogram representation for pooling can improve the recognition rate by 4% to 19% for 15 scene categories dataset, and 5% to 20% for Caltech101 dateset. For some experiments settings (e.g. hard quantization, linear kernel,average pooling), the recognition rate of 2-D histogram with a small codebook size (16) is already comparable or even better than the recognition rate of 1-D histogram with a large codebook size (256). When using a linear kernel, the performance of 2-D histogram representation does not degrade much compared to HIK kernel. The reason behind this is that 2-D pooling, which leads to sparser representation of images, can improve the linear discrimination of the features. This property of 2-D histogram pooling is of great value, since it enable us to use costless linear SVM[25] to do the classification job, and further make the method potential to be utilized in large-scale data.

4.2 Image Reranking

In an image retrieval system, the input of a certain query will yield millions of images. The problem of how to rank the most relevant images on the top is called image reranking. Image reranking has received an increasing

Table 1. Classification Accuracy of 15 Scene Categories

Method	1-D histogram CodebookSize=16	2-D histogram CodebookSize=16	1-D histogram CodebookSize=256
hard quantization, linear kernel, average pooling	66.0±0.8	71.7±0.4	73.3±1.1
hard quantization, linear kernel, max pooling	54.9±1.0	71.6±1.0	78.5±1.6
hard quantization, HIK kernel, average pooling	72.9±0.6	78.8±1.1	81.5±0.5
hard quantization, HIK kernel, max pooling	53.2±0.4	72.1±0.9	78.3±1.0
sparse coding, linear kernel, average pooling	69.1±0.9	73.4±1.5	77.5±0.3
sparse coding, linear kernel, max pooling	66.7±3.4	75.5±2.8	80.9±1.5
sparse coding, HIK kernel, average pooling	71.3±1.4	77.6±1.1	81.6±1.5
sparse coding, HIK kernel, max pooling	68.1±2.2	77.1±1.9	82.0±0.8

Table 2. Classification Accuracy of Caltech101

Method	1-D histogram CodebookSize=16	2-D histogram CodebookSize=16	1-D histogram CodebookSize=256
hard quantization, linear kernel, average pooling	43.8±1.5	49.0±1.7	47.9±1.3
hard quantization, linear kernel, max pooling	35.5±0.9	56.2±1.9	64.7±1.3
hard quantization, HIK kernel, average pooling	54.5±1.2	59.9±1.6	65.3±2.3
hard quantization, HIK kernel, max pooling	35.7±2.1	56.7±2.2	65.1±1.1
sparse coding, linear kernel, average pooling	48.2±1.1	53.0±1.2	62.2±1.3
sparse coding, linear kernel, max pooling	49.3±2.1	58.1±0.7	69.1±2.2
sparse coding, HIK kernel, average pooling	52.5±1.4	60.0±2.0	69.3±1.0
sparse coding, HIK kernel, max pooling	53.0±2.4	60.1±1.7	71.3±1.3

attentions from scholars these years.[26,27] In this section, we apply our 2-D histogram representation for pooling with Bag of Words model under the framework of random walk based image reranking.[27]

In random-walk based reranking, for M images, a similarity matrix of images, $\boldsymbol{S} \in \boldsymbol{R}^{M\times M}$, is first built, where $s_{i,j}$ is the similarity value of the i-th image to the j-th image. Then the rank of images is defined as:

$$\boldsymbol{Rank} = d\boldsymbol{S} \times \boldsymbol{Rank} + (1-d)\boldsymbol{p}, \quad \boldsymbol{p} = \left[\frac{1}{M}\right]_{M\times 1}, \tag{10}$$

where d is the damping factor. As reported in the reference,[27] $d > 0.8$ is often chosen, and the setting of d have relatively minor impact on the results. In this experiment, d is set to be 0.8.

For the experiments, images are first transformed to 1-D or 2-D histogram by Bag of Words model and average pooling, with a spatial pyramid level of 3. Then similarity matrix is build by computing the HIK kernel of the histograms. Finally, random walk based reranking algorithm is used to get the rank of the images.

The test datasets are build based on 15 scene categories and Caltech101 datasets. For 15 scene categories dataset, we first randomly select 40 images from each categories. Then for each categories, we add 20 noise images which are randomly selected from other categories. For Caltech101 dataset, the original number of images and the number of added noise images are 30 and 15, respectively. Images in each categories are reranked using the method described above. The final result is the average precision-recall curve of all the classes.

The codebook size is set to be 8. Results of 1-D histogram with codebook size 64 are also provided for reference. The experiments are repeated 5 times with different randomly selected images to get reliable results. Figure 9 and Figure 10 are the mean precision-recall curves for all the categories of 15 scene categories dateset and Caltech101 dataset, respectively. From the results, with a small codebook size (8), the results of 2-D histogram pooling are better than 1-D histogram pooling.

5. DISCUSSIONS

From the experiments above, by applying 2-D histogram representation of images for pooling, the performance of image classification and image reranking can be improved especially when using a small codebook. The reason

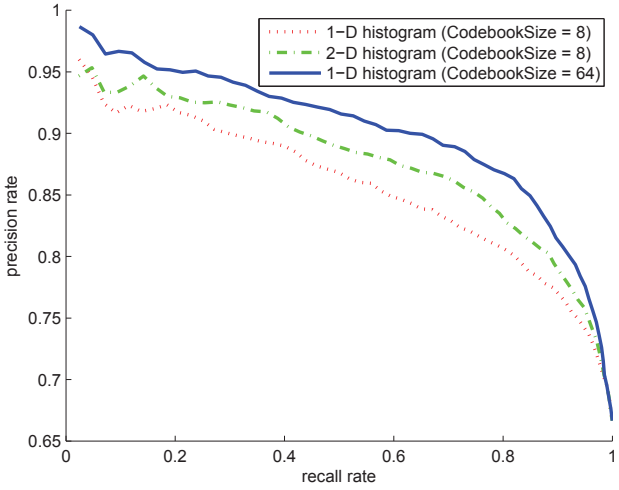

Figure 9. Reranking precision-recall curve for 15 scene categories dataset

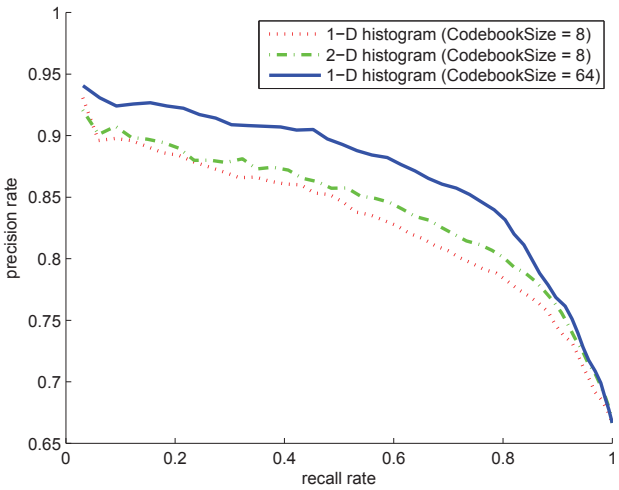

Figure 10. Reranking precision-recall curve for Caltech101 dataset

behind the improvement is that 2-D histogram, with the additional dimension, can extract more information from the encoded features $\boldsymbol{A} \in \boldsymbol{R}^{M \times K}$. Also, this sparser representation will lead to better linear discrimination of the features.

As mentioned in Section 3, 2-D histogram representation based pooling is not a space efficient method: it will transform an image to $\tilde{\boldsymbol{H}} \in \boldsymbol{R}^{K \times K}$ which has the same size with a 1-D representation $\boldsymbol{H} \in \boldsymbol{R}^{K^2}$. Also, in most of the time, the performance of the proposed method is not comparable to 1-D representation with a large codebook size. So what are the advantages of 2-D histogram based pooling? Firstly, it can be used to improve the performance when using small codebook. Small codebook is easy to be built, and can greatly save the time in coding step. Secondly, experiments show that 2-D histogram representation $\tilde{\boldsymbol{H}} \in \boldsymbol{R}^{K \times K}$ is a highly sparse matrix, which means we can speed up the processing of 2-D histogram representation by designing methods to compress the matrix. Here, the approaches in quadratic spatial codebook[20, 21] are of reference value.

To further improve the performance, 2-D histogram can be extended to high-dimensional histogram. For example, the largest, second largest and third largest values of $\boldsymbol{\alpha}_i$ can be extracted to build 3-D histogram. However, preliminary experiment shows that adding more dimension, which will obviously add more space complexity, cannot improve the performance coherently.

When using 2-D histogram representation, although more information from the encoded image features $\boldsymbol{A} \in \boldsymbol{R}^{M \times K}$ is extracted, only two values in each row of $\boldsymbol{A} \in \boldsymbol{R}^{M \times K}$ are utilized. A natural idea is to make the two values represent more information. For example, we can modify sparse coding into the following problem:

$$\min_{\boldsymbol{A},\boldsymbol{V}} \sum_{i=1}^{M} ||\boldsymbol{x}_i - \boldsymbol{\alpha}_i \boldsymbol{V}||^2, \quad s.t. \quad \exists! \quad \{x_i, y_i\}(x_i \neq y_i), \quad \alpha_{i,x_i} \neq 0, \quad \alpha_{i,y_i} \neq 0. \tag{11}$$

By replacing the sparse regulation factor $\lambda|\boldsymbol{\alpha}_i|$ in Equation 6 by a more strict condition, the two non-zero values in every $\boldsymbol{\alpha}_i$, $i = 1, 2, ..., M$ can be used to describe the descriptor better.

When sparse coding is used, $\boldsymbol{A} \in \boldsymbol{R}^{M \times K}$ has negative values. The absolute value of $\boldsymbol{A}$ used in pooling contains less information. To retain more information, one idea is to use NMF (non-negative matrix factorization) instead of sparse coding. Similar to sparse coding, in NMF, $\boldsymbol{A} \in \boldsymbol{R}^{M \times K}$ is defined by the following minimize problem:

$$\min_{\boldsymbol{A},\boldsymbol{V}} \sum_{i=1}^{M} ||\boldsymbol{x}_i - \boldsymbol{\alpha}_i \boldsymbol{V}||, \quad s.t. \quad \alpha_{i,j} \geq 0, \quad \forall i = 1, 2, ..., M, \quad j = 1, 2, ..., K. \tag{12}$$

Another solution is to utilize the negative values of $\boldsymbol{A}$ in the pooling step. For example, we can design f_1 and f_2 in Algorithm 1 to be the largest positive value and the smallest negative value of $\boldsymbol{a}_i$, respectively. However, preliminary experiments show it cannot improve the result.

6. CONCLUSION AND FUTURE WORK

In this paper, a novel pooling method, based on 2-D histogram representation, is proposed to retain more information from the encoded image features. 2-D histogram representation can be easily incorporated into state-of-the-art computer vision system frameworks. Experiments show that our approach improves current pooling methods, and can achieve satisfactory performance of image classification and image reranking even when using a small codebook and costless linear SVM.

The next step of our efforts is to design 2-D histogram pooling with more variations, and to test the method under more coding configurations. We will also do a theoretical evaluation of 2-D histogram representation to further understand it. Another direction is to research into compressing the sparse 2-D histogram matrix to make it space efficient, and applicable to large codebooks.

ACKNOWLEDGMENTS

We would like to thank Baodi LIU and Bin SHEN for beneficial discussions.

REFERENCES

[1] J. Sivic and A. Zisserman, "Video google: a text retrieval approach to object matching in videos," in *Proceedings of Ninth IEEE International Conference on Computer Vision*, **2**, pp. 1470–1477, 2003.

[2] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision* **60**, pp. 91–110, 2004.

[3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, **1**, pp. 886–893, 2005.

[4] J. van Gemert, C. Veenman, A. Smeulders, and J. Geusebroek, "Visual word ambiguity," *IEEE transactions on pattern analysis and machine intelligence* **32**, pp. 1271–1283, 2010.

[5] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1794–1801, 2009.

[6] Y. Boureau, J. Ponce, and Y. LeCun, "A theoretical analysis of feature pooling in vision algorithms," in *Proc. International Conference on Machine learning*, 2010.

[7] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2559–2566, 2010.

[8] M. Parsana, S. Bhattacharya, C. Bhattacharya, and K. R. Ramakrishnan, "Kernels on attributed pointsets with applications," in *Advances in Neural Information Processing Systems 20*, J. Platt, D. Koller, Y. Singer, and S. Roweis, eds., pp. 1129–1136, MIT Press, Cambridge, MA, 2008.

[9] D. Haussler, "Convolution kernels on discrete structures," in *Technical Report UCSC-CRL-99-10*, UC Santa Cruz, 1999.

[10] K. Grauman and T. Darrell, "The pyramid match kernel: discriminative classification with sets of image features," in *Proceedings of the Tenth IEEE International Conference on Computer Vision*, **2**, pp. 1458–1465, IEEE Computer Society, 2005.

[11] R. Kondor and T. Jebara, "A kernel between sets of vectors," in *Proceedings of International Conference on Machine Learning*, 2003.

[12] T. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," *Advances in Neural Information Processing Systems* , pp. 487–493, 1999.

[13] F. Perronnin and C. R. Dance, "Fisher kernels on visual vocabularies for image categorization," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[14] F. Perronnin, Y. Liu, J. Sanchez, and H. Poirier, "Large-scale image retrieval with compressed Fisher vectors," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3384–3391, IEEE, 2010.

[15] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the Fisher kernel for large-scale image classification," in *Proceedings of the European Conference on Computer Vision*, pp. 143–156, Springer, 2010.

[16] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[17] L. Bo and C. Sminchisescu, "Efficient Match Kernel between Sets of Features for Visual Recognition," in *Advances in Neural Information Processing Systems*, 2009.

[18] H. Jégou, M. Douze, and C. Schmid, "Packing bag-of-features," in *Proceedings of IEEE International Conference on Computer Vision*, 2009.

[19] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: spatial pyramid matching for recognizing natural scene categories," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, **2**, pp. 2169–2178, 2006.

[20] S. Lazebnik, C. Schmid, and J. Ponce, "A maximum entropy framework for part-based texture and object recognition," in *Proceedings of Tenth IEEE International Conference on Computer Vision*, **1**, pp. 832–838, IEEE, 2005.

[21] D. Liu, G. Hua, P. Viola, and T. Chen, "Integrated feature selection and higher-order spatial feature extraction for object categorization," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, 2008.

[22] X. Zhou, K. Yu, T. Zhang, and T. Huang, "Image classification using Super-vector coding of local image descriptors," in *Proceedings of the European Conference on Computer Vision, 2010*, pp. 141–154, Springer, 2010.

[23] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, **2**, pp. 524–531, 2005.

[24] 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," *Computer Vision and Image Understanding* **106**, pp. 59–70, 2007.

[25] S. Shalev-Shwartz, Y. Singer, and N. Srebro, "Pegasos: primal estimated sub-gradient solver for SVM," in *Proceedings of the 24th International Conference on Machine Learning*, pp. 807–814, ACM, 2007.

[26] R. Fergus, P. Perona, and A. Zisserman, "A visual category filter for google images," in *Proceedings of the 8th European Conference on Computer Vision, Prague, Czech Republic*, pp. 242–256, 2004.

[27] Y. Jing and S. Baluja, "Visualrank: applying pagerank to large-scale image search.," *IEEE Trans. Pattern Anal. Mach. Intell.* **30**, pp. 1877–1890, 2008.