Spatial Graph for Image Classification

Zifeng Wu, Yongzhen Huang, Liang Wang, and Tieniu Tan

National Lab. of Pattern Recognition, Institute of Automation,
Chinese Academy of Sciences, Beijing 100190, China
{zfwu,yzhuang,wangliang,tnt}@nlpr.ia.ac.cn

Abstract. Spatial information in images is considered to be of great importance in the process of object recognition. Recent studies show that human's classification accuracy might drop dramatically if the spatial information of an image is removed. The original bag-of-words (BoW) model is actually a system simulating such a classification process with incomplete information. To handle the spatial information, spatial pyramid matching (SPM) was proposed, which has become the most widely used scheme in the purpose of spatial modeling. Given an image, SPM divides it into a series of spatial blocks on several levels and concatenates the representations obtained separately within all the blocks. SPM greatly improves the performance since it embeds spatial information into BoW. However, SPM ignores the relationships between the spatial blocks. To address this problems, we propose a new scheme based on a spatial graph, whose nodes correspond to the spatial blocks in SPM, and edges correspond to the relationships between the blocks. Thorough experiments on several popular datasets verify the advantages of the proposed scheme.

1 Introduction

Image classification has become one of the most active topics in the recent literature. In particular, the bag-of-words model (BoW) [1] has shown its effectiveness and applicability in terms of scene and object classification. In BoW, the occurrences of visual words are counted within the local feature set of each image respectively to generate a histogram, which is treated as a representation of the original image. Afterwards, we can just match the representations to figure out the similarity of two images, and furthermore to tell if they are of the same category.

In the original BoW model, the spatial information of visual words is not taken into account, which conflicts with our intuition and experience. We can better perceive the real world with the spatial information. A recent psychological study on recognizing jumbled images [2] demonstrates the importance of (global) spatial information and calls for research efforts in spatial modeling. In [2], an original image is divided into small blocks, which are then shuffled up randomly to obtain a jumbled image. For reference, this process is illustrated in Figure 1. The spatial information of visual words is missing in a jumbled image. As a result, subjects' classification accuracy might drop from 80% to 20% [2],

K.M. Lee et al. (Eds.): ACCV 2012, Part I, LNCS 7724, pp. 716–729, 2013.

Fig. 1. An original image (left) and the corresponding jumbled image (right)

which shows the influence of spatial information in the classification process. The original BoW model, without any spatial information involved, simulates human's behavior in recognizing the jumbled image. In this way, we can hardly anticipate a good classification result.

Among all the efforts in spatial modeling, the spatial pyramid matching (SPM) scheme [3] is probably the most widely applied one. In SPM, an image is regularly divided into various blocks on several levels, as illustrated in Figure 2. The occurrences of visual words are then counted within these blocks respectively. Accordingly, we should match the representations from multiple corresponding blocks to find out if two images are of the same category. SPM can greatly improve the performance of BoW, and at the same time, it is easy to implement and of acceptable extra computational cost. As a result, SPM has already become an indispensable part in the BoW model.

In spite of the advantages, the blocks in SPM are treated independently. Two neighboring blocks are probably related considering that they are located close to each other, as illustrated in Figure 2. The spatial information of an image can be better reflected if the relation of spatial blocks are taken into account. However, the relation of blocks is completely ignored in SPM. To solve this problem, we propose a spatial modeling scheme based on a directed graph in this paper. In our scheme, blocks in SPM are represented by the nodes, and the relation of blocks which is missing in SPM is represented by the edges.

The main contribution of this paper is that we propose to embed the spatial information of an image into a spatial graph, by generating a series of histograms corresponding to nodes or edges of the graph. The proposed scheme is more flexible than SPM. Thorough experiments on 15 Scenes [3] and PASCAL VOC 2007 [4] show that this new scheme achieves better performance compared with SPM.

The remainder of this paper is organized as follows: Section 2 reviews the related work. Section 3 first introduces the original BoW model and its extension with SPM, and then proposes our scheme. Section 4 first explains the implementation details, and then reports and analyzes the experimental results. Finally, Section 5 concludes this paper.

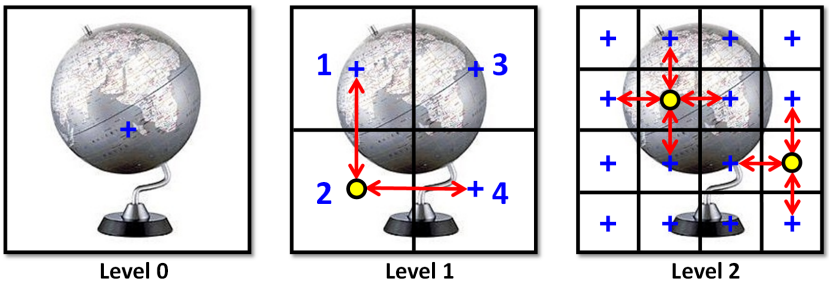

Fig. 2. SPM with 21 blocks on three levels: 1×1, 2×2 and 4×4. Middle: Bin 1 and Bin 2 are related to each other since they are neighbors, and so do Bin 2 and Bin 4. Right: More examples of relation between blocks.

2 Related Work

There is a great deal of work which takes the spatial information of visual words into account in the recent literature. They can be grouped into three major categories according to the adopted strategy for embedding spatial information.

The first is to embed spatial information into extended visual codes. Boureau et al. [5] embed local spatial information into macro-features which are extracted densely by concatenating small spatial neighboring local features. Morioka and Satoh [6] embed the relative spatial information of two visual words into a local pair-wise code. The pair-wise codes are obtained by clustering on pair-wise features extracted densely, each of which is a concatenation of two nearby local features. They further unify their work with the proximity distribution kernel [7] in [8], in order to combine the strengths of both, i.e., compactness and scale invariance. This kind of schemes focus on the local spatial information, but ignore the global spatial information.

The second is to express spatial information with an independent representation. The image-level representation of an image is the concatenation of a spatial section and an occurrence section obtained with the original bag-of-words (BoW) model [1]. Krapac et al. [9] propose to capture the spatial information of visual words with Fisher vectors. No matter how many dimensions a visual word owns (e.g., a visual word corresponds to a 129-dimensional vector in super-vector coding (SVC) [10]), the dimension of its spatial Fisher vector is the same. However, the performance is only comparable to the existing state-of-the-art schemes. The superiority of their scheme is thus about saving the memory and computational cost rather than improving the performance. Moreover, this superiority is true only if a high-dimensional coding scheme such as SVC is adopted.

The third is to pool spatially similar local features together to generate several representations and concatenate them, which is often referred to as spatial pooling. As a classic representative of the spatial pooling strategy, the spatial pyramid matching (SPM) scheme [3] is currently the most successful one, which is both

effective and easy to implement. There are also some extensions of SPM. Harada et al. [11] train a discriminative spatial pyramid by optimizing the weights of blocks. Wang et al. [12] adopt a shape-context-like division strategy with respect to 9 fixed reference points. Yang et al. [13] propose a co-occurrence kernel for image matching instead of the original kernel adopted in SPM. Their model acts better than SPM on their land-use dataset, but on other popular datasets such as 15 Scenes [3], it only achieves a modest improvement. Our scheme proposed in this paper is also an instance of the spatial pooling strategy. However, different from the above three studies, we focus on embedding spatial information into a directed graph.

To build up an integrated BoW model, coding is an indispensable part. Recently, many researchers make great efforts in developing better coding schemes. Generally, the existing coding schemes can be grouped into three categories, namely, *probabilistic schemes*, e.g., hard voting (HV) [1], soft voting [14] and supervector coding (SVC) [10], *reconstruction-based schemes*, e.g., sparse coding [15] and locality-constrained linear coding (LLC) [16], and saliency-based schemes, e.g., saliency coding [17,18]. Probabilistic schemes, cooperating with average pooling (or weighted average pooling for SVC [10]), and reconstruction-based schemes, cooperating with max pooling, often show different characteristics in various aspects. Saliency-based schemes usually show similar characteristics with reconstruction-based schemes. Accordingly, we conduct experiments with SVC, as a representative of probabilistic schemes, and LLC, as a representative of reconstruction-based schemes in this paper.

3 Methods

Methods in this section mainly refer to the pooling stage in the BoW model. In other words, with the output of the coding stage, the question is what we should do to generate the final image-level representation. In the following, we first introduce the original BoW model, and then its extension with SPM. Afterwards, we will propose our scheme.

3.1 BoW

Suppose that the codebook consists of K visual words, denoted by $\mathbf{c}_j$ respectively. For an image, local features are extracted either with a feature detector or just by dense sampling. We assign each of these features to a visual word and record the occurrences of each visual word. Thus, a K-bin histogram is obtained for each image.

Let X and Y denote two images, and $\mathbf{x}$ and $\mathbf{y}$ denote their normalized histograms respectively. Supposing that we extract M_X local features from X, denoted by $\mathbf{f}_i^X$ respectively, we can calculate $\mathbf{x}$ by:

$$\mathbf{x} = \mathbf{Z}^X_{k \times M_X} \cdot \mathbf{I}^X_{M_X} \tag{1}$$

wherein $\mathbf{Z}^X_{K\times M_X}$ is a matrix, each row of which (i.e., $\mathbf{z}^X_i$) corresponds to the coding output of the i-th feature, and $\mathbf{I}^X$ is a column vector whose entries are all one. In the case of HV, $\mathbf{z}_i$ is a vector with only one non-zero element, e.g., if $\mathbf{c}_j$ is the nearest code to $\mathbf{f}^X_i$, the j-th element of $\mathbf{z}_i$ will be one while the rest of its elements will be zero. Similarly, $\mathbf{y}$ is defined as:

$$\mathbf{y} = \mathbf{Z}^Y_{K\times M_Y} \cdot \mathbf{I}^Y_{M_Y}. \tag{2}$$

We can thus predict the similarity between the two images just by calculating the similarity between $\mathbf{x}$ and $\mathbf{y}$. Typically, it can be defined as the intersection kernel:

$$\kappa_{\mathrm{I}} = \min(\mathbf{x}, \mathbf{y})^{\mathrm{T}} \cdot \mathbf{I}_K. \tag{3}$$

Another common option is the linear kernel:

$$\kappa_{\mathrm{L}} = \mathbf{x}^{\mathrm{T}} \cdot \mathbf{y}. \tag{4}$$

3.2 BoW with SPM

No spatial information of visual words is considered in the original BoW model. To address this problem, SPM is proposed. The main idea of SPM is to match two images within a series of blocks on several levels. Those features matched on a high-resolution level will be excluded in matching on the following low-resolution levels.

The original definition of the SPM kernel is a little complicated [3], but it can be simply rewritten as the inner product of a weighting vector and the concatenation of every matching result within a separate block:

$$\begin{aligned}
\kappa'_{\mathrm{I}} &= \min(\mathbf{x}, \mathbf{y})^{\mathrm{T}} \cdot \mathbf{w} \\
\mathbf{x} &= [\mathbf{x}^{\mathrm{T}}_{0,1}, \mathbf{x}^{\mathrm{T}}_{1,1}, \ldots, \mathbf{x}^{\mathrm{T}}_{1,B(1)}, \ldots, \mathbf{x}^{\mathrm{T}}_{L,1}, \ldots, \mathbf{x}^{\mathrm{T}}_{L,B(L)}]^{\mathrm{T}} \\
\mathbf{y} &= [\mathbf{y}^{\mathrm{T}}_{0,1}, \mathbf{y}^{\mathrm{T}}_{1,1}, \ldots, \mathbf{y}^{\mathrm{T}}_{1,B(1)}, \ldots, \mathbf{y}^{\mathrm{T}}_{L,1}, \ldots, \mathbf{y}^{\mathrm{T}}_{L,B(L)}]^{\mathrm{T}} \\
\mathbf{w} &= [w_0, w_1, \ldots, w_1, \ldots, w_L, \ldots, w_L]^{\mathrm{T}}
\end{aligned} \tag{5}$$

wherein $\mathbf{x}_{l,b}$ and $\mathbf{y}_{l,b}$ denote the histograms of X and Y obtained within Bin b on Level l, L is the number of levels, $B(l)$ denotes a function returning the number of blocks on Level l, $\mathbf{w}$ denotes the weighting vector and w_l denotes the weight on Level l. $\mathbf{x}_{l,b}$ and $\mathbf{y}_{l,b}$ are the product of a coding matrix and a mask vector like:

$$\mathbf{x}_{l,b} = \mathbf{Z}_{K\times M_X} \cdot \mathbf{v}_{l,b}. \tag{6}$$

Different from the original BoW model, spatial information of local features is required in SPM. Suppose that $M_{\mathbf{X}}$ local features are extracted from X as $(\mathbf{f}^X_i, \mathbf{p}^X_i)$, wherein $\mathbf{p}^X_i$ denotes the location of $\mathbf{f}^X_i$ in the image. The i-th element of $\mathbf{v}_{l,b}$ can be defined as:

$$\mathbf{v}_{l,b}(i) = \begin{cases} 1 \text{ if } h(\mathbf{p}_i^X, l) = b \\ 0 \text{ else} \end{cases} \tag{7}$$

wherein $h(\mathbf{p}_i^X, l)$ is a function returning an index $\in \{1, 2, \ldots, B(l)\}$ denoting the block in which $\mathbf{p}_i^X$ lies on the specified Level l. Accordingly, the linear kernel with SPM is:

$$\kappa'_{\mathrm{L}} = \mathbf{x}^{\mathrm{T}} \cdot (\mathbf{y} \odot \mathbf{w}) \tag{8}$$

wherein $\odot$ denotes the element-wise multiplication.

3.3 Our Scheme

Our scheme is to match images with their spatial information embedded in a directed graph, so as to reflect the relation between neighboring blocks. The main idea is to represent an image with a series of histograms corresponding to the nodes and edges in a directed graph, as illustrated in Figure 3.

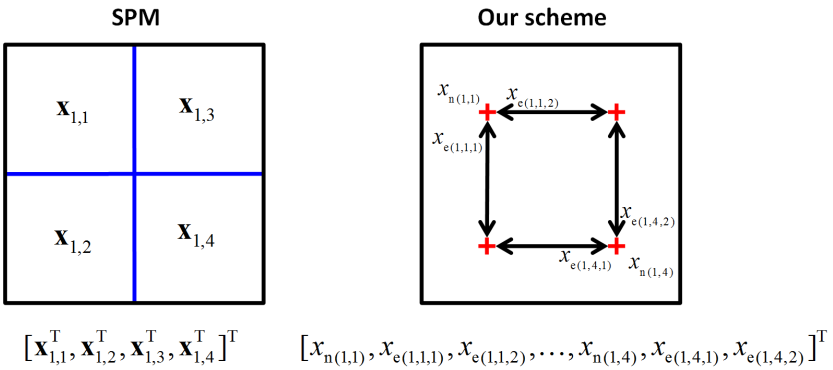

Fig. 3. A comparison between SPM and our schemes on Level 1. $x^j_{\mathrm{n}(l,n)}$: the j-th element of $\mathbf{x}_{\mathrm{node}(l,n)}$. $x^j_{\mathrm{e}(l,n,e)}$: the j-th element of $\mathbf{x}_{\mathrm{edge}(l,n,e)}$. See the text for details.

In our scheme, the image-level representations are defined as:

$$\begin{aligned} \mathbf{x} &= [\mathbf{x}_{0,1}^{\mathrm{T}}, \mathbf{x}_{1,1}^{\mathrm{T}}, \ldots, \mathbf{x}_{1,N(1)}^{\mathrm{T}}, \ldots, \mathbf{x}_{L,1}^{\mathrm{T}}, \ldots, \mathbf{x}_{L,N(L)}^{\mathrm{T}}]^{\mathrm{T}} \\ \mathbf{x}_{l,n} &= [\mathbf{x}_{\mathrm{node}(l,n)}^{\mathrm{T}}, \mathbf{x}_{\mathrm{edge}(l,n,1)}^{\mathrm{T}}, \ldots, \mathbf{x}_{\mathrm{edge}(l,n,E(l,n))}^{\mathrm{T}}]^{\mathrm{T}} \end{aligned} \tag{9}$$

wherein $\mathbf{x}_{\mathrm{node}(l,n)}$ denotes the histogram of X corresponding to Node n on Level l, $\mathbf{x}_{\mathrm{edge}(l,n,e)}$ denotes the histogram corresponding to the edge from Node n on Level l to its e-th neighbor, $N(l)$ denotes a function returning the number of nodes on Level l and $E(l,n)$ denotes a function returning the outdegree of

Node n on Level l. $\mathbf{x}_{\text{node}(l,n)}$ and $\mathbf{x}_{\text{edge}(l,n,e)}$ are each a product of the coding matrix and a mask vector:

$$\mathbf{x}_{\text{node}(l,n)} = \mathbf{Z}_{n\times m_X} \cdot \mathbf{u}_{l,n} \tag{10}$$

$$\mathbf{x}_{\text{edge}(l,n,e)} = \mathbf{Z}_{n\times m_X} \cdot \mathbf{u}_{l,n,e}. \tag{11}$$

The i-th element of $\mathbf{u}_{l,n}$ and $\mathbf{u}_{l,n,e}$ can be respectively defined as:

$$\mathbf{u}_{l,n}(i) = \begin{cases} 1 \text{ if } h_{\text{node}}(\mathbf{p}_i^X, l) = n \\ 0 \text{ else} \end{cases} \tag{12}$$

$$\mathbf{u}_{l,n,e}(i) = \begin{cases} 1 \text{ if } h_{\text{edge}}(\mathbf{p}_i^X, l, n) = e \\ 0 \text{ else} \end{cases} \tag{13}$$

wherein $h_{\text{node}}(\mathbf{p}_i^X, l)$ is an index $\in \{1, 2, \ldots, N(l)\}$ denoting the spatially nearest node to $\mathbf{p}_i^X$ on the specified Level l, and $h_{\text{edge}}(\mathbf{p}_i^X, l, n)$ is an index $\in \{1, 2, \ldots, E(l,n)\}$ denoting the nearest edge to $\mathbf{p}_i^X$ among all the edges originated from Node n on Level l.

The above explanations are presented supposing that the sum (average) pooling scheme is adopted. However, there will be no difficulty in extending the formulations for weighted average pooling and max pooling. We omit the details since the extension is straightforward.

It is worthy noting that we introduce the representations of edges ($\mathbf{x}_{\text{edge}(l,n,e)}$) to reflect the relation between neighboring nodes. From this point of view, $\mathbf{u}_{l,n,e}$ defined in Equation (13) is not appropriate. What we want is to reflect the relation between Node n and its neighbors, however, only the features belonging to Node n are involved. To deal with this problem, we introduce the soft assignment mechanism into this process. In this way, $\mathbf{u}_{l,n}$ and $\mathbf{u}_{l,n,e}$ turn into weighting vectors. We will discuss the details in Section 4.1.

4 Experimental Results

4.1 Implementation Details

To implement our scheme, there are two main aspects that we must handle with. The first one is how to build up a directed graph, and the second one is how to assign local features to the nodes and edges in these graphs.

To build up the graph, we must first locate the nodes denoting different blocks. In this paper, we simply set the center of each block as a node $\mathbf{p}_{(l,n)}$, i.e., the location of Node n on Level l. Afterwards, we assign an edge between two nodes on the same level if their corresponding blocks are neighbors, as illustrated in Figure 3.

Given a feature in an image, we should decide which node and edge it should be assigned to, as defined in Equations (12) and (13). As a common choice, we

can conduct the node assignment with respect to the spatial Euclidean distances between features and nodes. Thus, h_{node} in Equations (12) can be defined as:

$$\begin{aligned} h_{\text{node}}(\mathbf{p}_i^X, l) &= \underset{n=1,\ldots,N(l)}{\arg\min}\ d_{\text{node}}(\mathbf{p}_i^X, l, n) \\ d_{\text{node}}(\mathbf{p}_i^X, l, n) &= \left\|\mathbf{p}_i^X - \mathbf{p}_{(l,n)}\right\|_2. \end{aligned} \tag{14}$$

We define h_{edge} in Equation (13) as:

$$\begin{aligned} h_{\text{edge}}(\mathbf{p}_i^X, l, n) &= \underset{e=1,\ldots,E(l,n)}{\arg\max}\ d_{\text{edge}}(\mathbf{p}_i^X, l, n, e) \\ d_{\text{edge}}(\mathbf{p}_i^X, l, n, e) &= (\overrightarrow{P_{l,n}P_i^X} \cdot \overrightarrow{P_{l,n}P_{\text{neighbor}}^X}) \\ \overrightarrow{P_{l,n}P_i^X} &= \mathbf{p}_i^X - \mathbf{p}_{(l,n)} \\ \overrightarrow{P_{l,n}P_{\text{neighbor}}^X} &= \mathbf{p}_{(l,neighbor(l,n,e))} - \mathbf{p}_{(l,n)} \end{aligned} \tag{15}$$

wherein $neighbor(l, n, e)$ is a function returning the index $\in \{1, 2, \ldots, N(l)\}$ of the e-th neighbor of Node n on Level l. Here, we adopt the dot product as the distance between a feature and an edge. Compared with the spatial distance from a feature to a edge, the features which are close to a node will be smoothly assigned to the node's edges. As mentioned in Section 3.3, soft assignment is required for the motivation of our scheme. Fortunately, the treatment is in hand considering the distance, i.e., d_{node} in Equation (14), and the similarity, i.e., d_{edge} in Equation (15) have already been defined. We apply the Gaussian function for soft assignment:

$$d'_{\text{node}} = e^{-\lambda_{\text{n}} d_{\text{node}}^2} \tag{16}$$

$$d'_{\text{edge}} = \begin{cases} e^{-\lambda_{\text{e}}(d_{\text{edge}} - 0.5)^2} & \text{if } d_{\text{edge}} < 0.5 \\ 1 & \text{else} \end{cases} \tag{17}$$

wherein λ_{n} and λ_{e} are two parameters. And the elements of the weighting vectors in Equations (10) and (11) are obtained after normalization:

$$\mathbf{u}_{l,n}(i) = \frac{d'_{\text{node}}(\mathbf{p}_i^X, l, n)}{\sum\limits_{n=1,\ldots,N(l)} d'_{\text{node}}(\mathbf{p}_i^X, l, n)} \tag{18}$$

$$\mathbf{u}_{l,n,e}(i) = \frac{d'_{\text{edge}}(\mathbf{p}_i^X, l, n, e)}{\sum\limits_{e=1,\ldots,E(l,n)} d'_{\text{edge}}(\mathbf{p}_i^X, l, n, e)} \cdot \mathbf{u}_{l,n}(i). \tag{19}$$

Obviously,

$$\mathbf{u}_{l,n}(i) = \sum_{e=1,\ldots,E(l,n)} \mathbf{u}_{l,n,e}(i)$$

which means that

$$\mathbf{x}_{\text{node}(l,n)} = \sum_{e=1,\ldots,E(l,n)} \mathbf{x}_{\text{edge}(l,n,e)}.$$

In other words, $\mathbf{x}_{\text{node}(l,n)}$ and $\mathbf{x}_{\text{edge}(l,n,e)}$ $(e = 1, \ldots, E(l,n))$ are linearly correlated. Therefore, we can remove $\mathbf{x}_{\text{node}(l,n)}$ from the final representation without losing useful information. Notably, this is not always true as the strategy for generating the two weighting vectors varies.

4.2 Datasets and Experimental Settings

We evaluate our scheme on the 15 Scenes dataset [3] for scene classification, and the PASCAL VOC 2007 dataset [4] for object classification. In the 15 Scenes dataset, there are 4,485 images of natural scenes in total, belonging to 15 categories (e.g., bedroom, CALsuburb and industrial), each of which consists of 200 to 400 images. In the PASCAL VOC 2007 dataset, there are 9,963 images in total, belonging to 20 categories, e.g., bird, car and person. Images in VOC 2007 carry obvious variation in scale, illumination, viewpoint, pose, occlusion and so on. Generally speaking, the tendency of the resulting curves is similar on different VOC datasets, since they are of high overlap of the collected images (nearly 50% between VOC 2007 and VOC 2011). Most works in the recent literature report their results on VOC 2007 instead of the newer datasets because the labels on test images are fully released. For the sake of conveniences in evaluation and comparison with related work, we follow this policy.

For 15 Scenes, we follow the evaluation settings proposed in [3], i.e., randomly pick out 100 images from each category for training, and keep the remaining images for testing. We repeat the evaluation for 10 times and report the average classification accuracy and the standard deviation. For VOC 2007, we follow the official evaluation rules, i.e., train models on the *trainval* set, test on the *test* set, and report the mean average precision (mAP).

SIFT descriptors [19] are densely extracted every four pixels for all images on three scale, i.e., 16×16, 24×24 and 32×32 in pixels. The local features are L2-normalized as preprocessing. Codebooks are trained by the k-means clustering. Lib-linear SVMs [20] are trained as classifiers. For comparison, SPM [3] is performed on three levels, i.e., 1×1, 2×2 and 3×1. Accordingly, we build up graphs on three levels with the same setting. We do not follow the original SPM configuration in [3] for two reasons: first, the used configuration is of lower dimension and performs as well as or even better than the original one; second, there is no need to worry about the compatibility issues between the 3×1 level and the intersection kernel, since LLC and SVC are both designed for linear SVM. Cross-validation on training set shows that the optimal soft assignment parameters are relatively insensitive to the variation of code sizes and evaluation datasets. However, the optimal parameters tend to vary if the adopted coding scheme is different. Thus, we fix λ_n and λ_e for different kinds of coding schemes in our experiments. For SVC, $(\lambda_n, \lambda_e) = (32, 8)$, which are also appropriate for other probabilistic schemes. For LLC, $(\lambda_n, \lambda_e) = (16, 8)$, which are also appropriate for other reconstruction-based schemes.

It is worthy noting that we implement a general framework of the BoW model to ensure fair and comprehensive comparison. The results of BoW, SPM and our scheme reported in this paper are all obtained with this framework. As a result, there might be discrepancies between our results and those reported by the original authors.

4.3 Basic Results

To test the configurations of blocks and levels, we conduct a series of experiments with different combinations of levels. The detailed results are reported in Table 1. Only the results obtained with LLC on 15 Scenes are reported due to limited space, since the results for different coding schemes or datasets are basically the same. Results obtained with the representations on separate levels are also attached to show the contribution of different levels. Note that the last two rows are both on Level 2. Thus, the columns labeled by Pyramid in Row 3 denote the configuration: 1×1, 2×2 and 3×1, and those in Row 4 denote the configuration: 1×1, 2×2 and 4×4.

Table 1. Classification results obtained with LLC on 15 Scenes. The best results for different code sizes are shown in bold. Note that the last two rows are both on Level 2. See the text for details.

Code size:	$n = 16$		$n = 512$		$n = 8192$	
L	Single level	Pyramid	Single level	Pyramid	Single level	Pyramid
0 (1×1)	35.0 ± 0.6		64.5 ± 0.6		77.1 ± 0.7	
1 (2×2)	58.0 ± 0.6	58.7 ± 0.5	77.1 ± 0.6	77.4 ± 0.6	82.7 ± 0.4	83.0 ± 0.4
2 (3×1)	55.2 ± 0.5	61.2 ± 0.6	76.1 ± 0.5	$\mathbf{78.3 \pm 0.7}$	82.4 ± 0.4	$\mathbf{83.3 \pm 0.3}$
2 (4×4)	61.1 ± 0.5	$\mathbf{61.7 \pm 0.3}$	76.7 ± 0.3	77.7 ± 0.4	80.4 ± 0.2	82.5 ± 0.2

The improvement in performance shown in Table 1 agrees with our anticipation. When $L = 0$, our scheme becomes an analogue of the original BoW model, where no spatial information is involved. When $L > 0$, the performance improves as L increases, because finer spatial information is embedded. However, simply increasing the number of blocks does not always lead to better results. For example, the performance listed in Row 4 denoting the 4×4 level is inferior to the performance listed in Row 3 denoting the 3×1 level. The results demonstrate that our configuration is appropriate.

4.4 Comparison with SPM

We report the classification results of our scheme and SPM in Figure 4 for comparison on separate levels. The results of the original BoW model are also depicted for reference, and it again demonstrates the importance of spatial information. There are obvious gaps between SPM L1 and Graph L1, and between

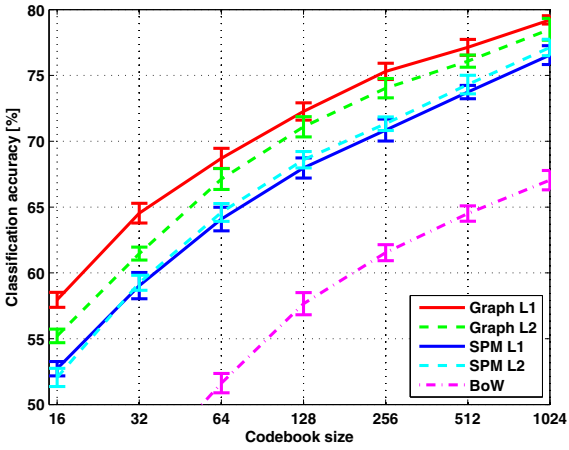

Fig. 4. Classification results obtained with representations on different levels separately on 15 Scenes. L1: 2×2. L2: 3×1.

Table 2. Classification results with LLC and different code sizes on VOC 2007

Code size	BoW	SPM	Ours
16	16.0	24.0	**27.0**
128	24.5	35.4	**38.2**
1024	35.8	45.7	**47.9**
8192	48.4	55.5	**56.3**

SPM L2 and Graph L2. Some researchers would argue that the dimension of the representations in our scheme is higher. But note that higher-dimensional representations do not always lead to better results, as demonstrated in Section 4.3, and that Graph L2 is always better than Graph L1 though they both involve representations of the same dimension ($4n$). The results in Figure 4 can thus verify the effectiveness of our scheme on different levels.

To investigate the performance of our scheme with the representations on three levels all involved, we test our method on two datasets, i.e., 15 Scenes [3] and PASCAL VOC 2007 [4]. On the 15 Scenes dataset, our result is 83.3%, as listed in Table 1. The result in the original SPM paper [3] is 81.4%, and the one in [12] is 81.6%. The classification results on VOC 2007 are given in Table 2. The results of the original BoW model are also attached for reference. Table 2 demonstrates the great contribution of spatial modeling, since both SPM and our scheme outperform BoW greatly. In addition, our scheme consistently outperforms SPM with different code sizes on different datasets. Note that results in Table 2 are all obtained with the same λ_n and λ_e. Therefore, it shows the insensitivity of the two parameters. We can draw a conclusion that our scheme cooperates fairly well with the representative of reconstruction-base schemes, i.e., LLC, considering

Table 3. Classification results with SVC and different code sizes on VOC 2007

Code size:	$n = 16$		$n = 64$		$n = 256$	
Category	SPM	Ours	SPM	Ours	SPM	Ours
aeroplane	63.9	**64.5**	70.5	**70.9**	73.2	**73.2**
bicycle	52.4	**53.7**	58.2	**60.3**	62.4	**63.3**
bird	38.6	**39.1**	39.9	**42.1**	50.4	**50.6**
boat	**64.7**	64.4	**69.3**	68.4	**70.8**	70.2
bottle	19.9	**20.1**	21.4	**22.8**	24.7	**26.0**
bus	54.1	**56.6**	61.8	**61.9**	**65.6**	65.0
car	70.6	**72.4**	74.7	**75.4**	77.0	**77.9**
cat	48.4	**49.5**	55.9	**57.9**	**59.9**	59.8
chair	47.9	**48.3**	50.0	**50.5**	**54.6**	54.2
cow	35.0	**36.6**	41.5	**41.5**	**44.4**	44.2
dinningtable	47.9	**50.4**	50.3	**51.8**	53.2	**53.2**
dog	**37.1**	36.3	36.6	**37.0**	44.1	**45.6**
horse	72.6	**73.7**	74.5	**76.2**	76.9	**78.1**
motorbike	57.8	**58.5**	62.8	**64.3**	66.8	**67.2**
person	77.0	**77.7**	80.3	**80.8**	83.0	**83.7**
pottedplant	20.6	**22.2**	24.6	**25.4**	28.1	**28.6**
sheep	**40.0**	39.8	44.6	**47.0**	47.1	**48.1**
sofa	47.6	**48.1**	52.1	**52.9**	55.1	**56.2**
train	68.1	**70.1**	74.3	**74.8**	77.4	**77.6**
tvmonitor	39.9	**43.2**	48.7	**50.4**	53.4	**55.1**
mean AP	50.2	**51.2**	54.6	**55.6**	58.4	**58.9**

the reported results in Table 2. Notably, Wang et al. [16] reports higher results with LLC, i.e., 59.3%. However, this result is not reproducible even with their own released source code, and our results are more comparable with those reported by Chatfield et al. in their extensive survey paper on coding schemes [21]. To further investigate the performance of our scheme when cooperating with the representative of probabilistic coding schemes, i.e., SVC, we list the category-wise classification results on VOC 2007 in Table 3. The results again show that our scheme consistently performs better than SPM with different codes sizes.

4.5 Efficiency Analysis

The extra computational cost of our method is brought in by assigning features to nodes and edges of the spatial graph. The overall computational complexity for image representation is less than $\mathcal{O}(K \cdot M + N_{\text{all}} \cdot M)$, wherein N_{all} denotes the total number of spatial regions. Usually, $N \ll K$. For example, in one of our experiments reported in Table 2, $K = 8192$, while $N_{\text{all}} = 13$. As a result, the additional cost of our method is ignorable.

4.6 Discussion

The original SPM scheme [3] grants different priority to different levels in order to balance their weights in the image-level representation. Harada et al. [11] even train the weights of different levels and blocks. Intuitively, such policy would boost the performance. However, we find empirically that re-weighting between levels gains limited improvements and brings in extra cost in practice. Using the same priority is a commonly-adopted policy in the recent literature, e.g., [21], a generally recognized survey on coding schemes.

Each node is a centroid of the local features extracted within a SPM block from the training image set, and each edge corresponds to a pair of neighboring blocks in SPM. The grid-like structure seems too rigid, and can be improved. We can make the nodes movable, the edges removable and the graph code-specific. Besides, supervised learning might generate discriminative spatial graphs and further boosts the performance.

A histogram on node reflects the occurrence of features in a block, and a histogram on edge reflects the occurrence of features which lie in one block and tend to shift into another. However, the histogram on a node is linearly correlated to those on its edges in our current implementation due to our assigning strategy. There might be a better strategy which preserves richer information. For example, the features definitely belonging to a block are assigned to the corresponding node, while the features tending to shift are assigned to histograms on edges.

5 Conclusion and Future Work

Among different strategies for spatial modeling, spatial pooling has been the most successful one. As a representative of spatial pooling schemes, SPM has become one standard part of an integrated BoW model due to its great simplicity and high performance. However, studies have shown that it is far from perfectly simulating human's behavior in perceiving spatial information. Two possible limitations of SPM include ignoring the relation of blocks. In this paper, we have proposed to capture the spatial information in images with a directed graph. Our scheme, which considers the relationship between spatial blocks, has shown its advantages in our experiments. In spite of the simplification in implementation, the proposed scheme has outperformed SPM with different kinds of coding schemes on several popular datasets.

After a period of achieving accomplishments in the feature space in terms of local feature detection, description and coding, it becomes more demanding for us to put more efforts in the work about the image space, i.e, capturing the spatial information contained in images. As one of the efforts towards this aim, the follow-up work of this paper is in two aspects: The first is to build up more flexible spatial graphs. The second is to find a better way to represent the relation of blocks so as to generate richer representation.

Acknowledgement. This work is supported by National Natural Science Foundation of China (61135003, 61203252), Tsinghua National Laboratory for Information Science and Technology Cross-discipline Foundation (Y2U1011MC1).

References

1. Csurka, G., Dance, C.R., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: ECCV (2004)
2. Parikh, D.: Recognizing jumbled images: the role of local and global information in image classification. In: ICCV (2011)
3. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In: CVPR (2006)
4. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC 2007) Results (2007)
5. Boureau, Y., Bach, F., LeCun, Y., Ponce, J.: Learning mid-level features for recognition. In: CVPR (2010)
6. Morioka, N., Satoh, S.: Building Compact Local Pairwise Codebook with Joint Feature Space Clustering. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 692–705. Springer, Heidelberg (2010)
7. Ling, H., Soatto, S.: Proximity distribution kernels for geometric context in category recognition. In: ICCV (2007)
8. Morioka, N., Satoh, S.: Compact correlation coding for visual object categorization. In: ICCV (2011)
9. Krapac, J., Verbeek, J., Jurie, F.: Modeling spatial layout with Fisher vectors for image categorization. In: ICCV (2011)
10. Zhou, X., Yu, K., Zhang, T., Huang, T.S.: Image Classification Using Super-Vector Coding of Local Image Descriptors. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 141–154. Springer, Heidelberg (2010)
11. Harada, T., Ushiku, Y., Yamashita, Y., Kuniyoshi, Y.: Discriminative spatial pyramid. In: CVPR (2011)
12. Wang, X., Bai, X., Liu, W., Latecki, L.J.: Feature context for image classification and object detection. In: CVPR (2011)
13. Yang, Y., Newsam, S.: Spatial pyramid co-occurrence for image classification. In: ICCV (2011)
14. van Gemert, J.C., Veenman, C.J., Smeulders, A.W.M., Geusebroek, J.M.: Visual word ambiguity. IEEE Transactions on Pattern Analysis and Machine Intelligence 32, 1271–1283 (2010)
15. Yang, J., Yu, K., Gong, Y., Huang, T.S.: Linear spatial pyramid matching using sparse coding for image classification. In: CVPR (2009)
16. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T.S., Gong, Y.: Locality-constrained linear coding for image classification. In: CVPR (2010)
17. Huang, Y., Huang, K., Yu, Y., Tan, T.: Salient coding for image classification. In: CVPR (2011)
18. Wu, Z., Huang, Y., Wang, L., Tan, T.: Group encoding of local features in image classification. In: ICPR (2012)
19. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 2(60), 91–110 (2004)
20. Fan, R., Chang, K., Hsieh, C., Wang, X., Lin, C.: Liblinear: a library for large linear classification. Journal of Machine Learning Research 9, 1871–1874 (2008)
21. Chatfield, K., Lempitsky, V., Vedaldi, A., Zisserman, A.: The devil is in the details: an evaluation of recent feature encoding methods. In: BMVC (2011)