Orientational Pyramid Matching for Recognizing Indoor Scenes

Lingxi Xie[1]* Jingdong Wang[2] Baining Guo[3] Bo Zhang[4] Qi Tian[5]
[1,4]LITS, TNList, Dept. of Computer Sci. and Tech., Tsinghua University, Beijing, China
[2,3]Microsoft Research Asia, Beijing, China
[5]Department of Computer Science, University of Texas at San Antonio, Texas, USA
[1]198808xc@gmail.com [2]jingdw@microsoft.com
[3]bainguo@microsoft.com [4]dcszb@mail.tsinghua.edu.cn [5]qitian@cs.utsa.edu

Abstract

Scene recognition is a basic task towards image understanding. Spatial Pyramid Matching (SPM) has been shown to be an efficient solution for spatial context modeling. In this paper, we introduce an alternative approach, Orientational Pyramid Matching (OPM), for orientational context modeling. Our approach is motivated by the observation that the 3D orientations of objects are a crucial factor to discriminate indoor scenes. The novelty lies in that OPM uses the 3D orientations to form the pyramid and produce the pooling regions, which is unlike SPM that uses the spatial positions to form the pyramid. Experimental results on challenging scene classification tasks show that OPM achieves the performance comparable with SPM and that OPM and SPM make complementary contributions so that their combination gives the state-of-the-art performance.

1. Introduction

Scene recognition is a fundamental task in computer vision. Conventional approaches, such as the Bag-of-Features (BoF) model [7], the Object Bank (OB) model [26], and the Bag-of-Parts (BoP) model [23], are shown capable of generating discriminative descriptors. Spatial Pyramid Matching (SPM) [25] together with the BoF model has achieved satisfactory performance in many recognition tasks. However, those approaches have limited ability to deal with the challenging indoor scene recognition problem as many indoor scenes are composed of almost the same set of objects with similar spatial layouts.

Let us take the scene images in Figure 1 (a) as the examples. The top two rows show the example images of two categories, *classroom* and *meeting room*. One can see that these two scenes contain almost the same objects: a

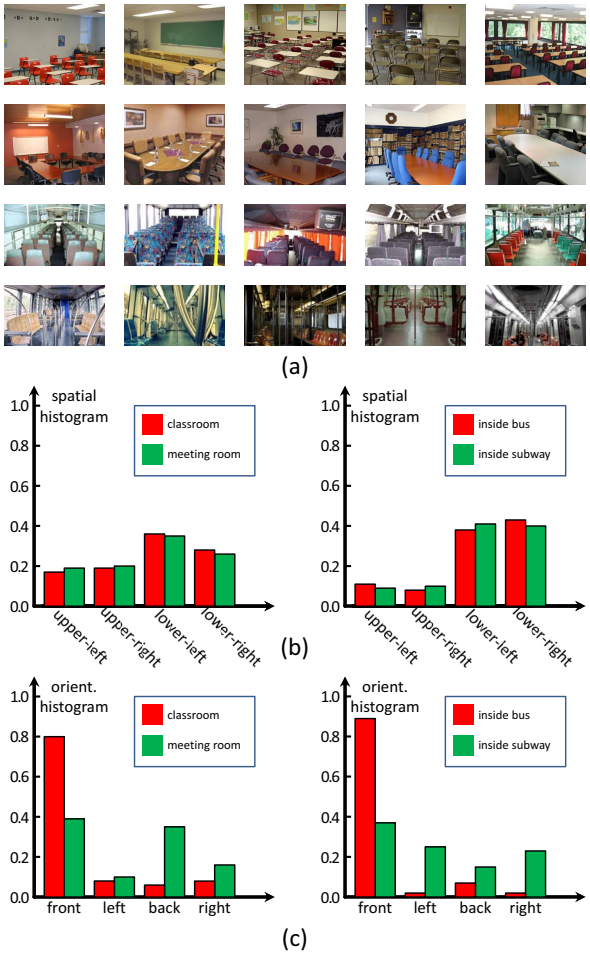

Figure 1. (a) Sample images for 4 scene categories from the MIT Indoor-67 dataset [32]: *classroom*, *meeting room*, *inside bus* and *inside subway* (from top to bottom). For both category pairs, the orientational features are more discriminative than the spatial features. (b) The spatial distributions of *chairs* in both confusing pairs are very similar. (c) The orientational distribution of chairs are much more dispersive.

*This work was done when Lingxi Xie was an intern at Microsoft Research.

1063-6919/14 $31.00 © 2014 IEEE
DOI 10.1109/CVPR.2014.477

large number of chairs, desks (tables), tidy walls, and daylight lamps. These objects are visually similar, resulting in the appearance features, *e.g.*, SPM with BoF, have limited disciminative ability. However, we notice that the 3D orientations of some objects, *e.g.*, chairs, in the rooms are different: the chairs in a *classroom* are often oriented to the same direction, towards the teacher's desk or the blackboard, whereas the chairs in a *meeting room* are oriented to different directions so that people in the room can face to others for talk convenience. Similar situations also occur in other scene categories, such as *subway carriages* and *bus carriages* (shown in the bottom rows), and others.

Figures 1 (b) and (c) show the quantitative illustration of the above analysis: the spatial and orientational histograms of chairs in the *classroom* and *meeting room* examples in the MIT Indoor-67 dataset [32]. We manually count the numbers of chairs with different spatial locations and 3D orientations in the two scene categories. The statistical results on 100 *classroom* and *meeting rooms* images are plotted as histograms. It can be seen that the divergence between the histograms from the orientations is much larger than the divergence between the histograms from the spatial locations, and thus orientation is a crucial cue to differentiate the *classroom* category from the *meeting room* category.

The above observation suggests that 3D orientations are helpful to discriminate confusing indoor scene categories. We follow the data-driven model [15] to estimate 3D orientations of image patches. and propose a so-called Orientational Pyramid Matching (OPM) model, which is the key novelty of this paper, to encode 3D orientations into the image-level feature vectors. OPM partitions the set of local features into a hierarchical set of pooling regions. Different from SPM that uses the positions of the local image patches, OPM uses the 3D orientations to index the patches and form the pyramid in the orientational space. Experimental results indicate that OPM achieves recognition performance comparable to SPM, and that OPM and SPM make complementary contributions to recognition task, therefore the integration of SPM and OPM achieves the state-of-the-art performance on challenging scene classification datasets.

The remainder of this paper is organized as follows. First, we give a survey of related works in Section 2. Then we introduce the state-of-the-art methods for scene recognition in Section 3. In Section 4, we illustrate our approach in two parts, *i.e.*, the Orientational Pyramid Matching (OPM) algorithm in Section 4.1, and the 3D orientation extraction in Section 4.2. After experimental results are shown in Section 5, we draw the conclusions in Section 6.

2. Related Works

Scene recognition is a fundamental task towards image understanding. Most often, we refer to *scene* as a place where an event or action happens, and it is possible for humans to recognize hundreds or even thousands of scene categories [42]. However, it is still challenging for the computer algorithms to discriminate even a small number of scene concepts. Many previous works are focused on representing scenes with statistics-based features, such as the global [30] and local [11] image descriptors. Beyond the Bag-of-Features (BoF) model [7], efforts are made to improve the description power of the features, such as incorporating spatial context in the scene representation [14][25], integrating multiple types of descriptors [4], quantizing local descriptors with less information loss [40][13], constructing mid-level concepts for better representation [45][44], constructing visual topic models [29], adopting kernel methods [41], discovering semantic regions for feature summarization [22] [43], and normalizing features to cooperate with various pooling strategies [46].

It is well known that geometric context, such as spatial layout, planar surfaces and orientational features, are more important in scene understanding than in object recognition tasks [3]. Based on basic geometric elements such as vanishing points [33], straight lines [19] and rectangles [2], various types of geometric features could be efficiently extracted, such as the occlusion boundaries [18], orthogonal planes [28], box layouts [16][17], and so on. This paper will study another type but not yet well studied geometric feature, 3D orientation, for scene recognition.

There are some works investigating 3D orientations. For example, in the areas of 3D image processing [35] or video action recognition [36], various techniques have been adopted to generalize the 2D descriptors to the corresponding 3D version. It is verified that 3D descriptors are much more descriptive since richer information has been encoded into the histograms [36]. Those 3D orientation based features are different from ours as their data are 3-dimensional while the 3D orientations described in this paper come from the 3D geometric information of a 2D image.

3. The State-of-the-Art

The BoF model and its variants are popular image representation methods for classification. They are composed of three basic stages: local descriptor extraction, feature encoding, and spatial pooling.

The local feature extraction stage usually extracts a set of local descriptors, *e.g.*, SIFT [27] or HOG [8], from the interest points or densely-sampled image patches of an image. The feature encoding module then assigns each descriptor to the closest entry in a visual vocabulary: a codebook learned offline by clustering a large set of descriptors with K-Means or Gaussian Mixture Model (GMM) algorithm. The descriptor assignment can also be soft [47] and the assignment weights can be determined according to the distances to the dictionary elements or learned using the Locality-constrained Linear Coding (LLC) algorithm [40].

Recently coding techniques includes Fisher Vector [31], the Vector of Locally Aggregated Descriptors (VLAD) [20], and the Super Vector [49]. The BoF model and other coding techniques have been compared in [34].

Spatial pooling consists of partitioning an image into a set of regions, aggregating feature-level statistics over these regions, and concatenating the region descriptors as an image-level feature vector. Image partition can be obtained by Spatial Pyramid Matching (SPM) [25] or some learning techniques [21]. Aggregation of descriptors within a region is often performed with pooling strategy [5], such as average-pooling, max-pooling, or Geometric ℓ_p-norm Pooling (GLP) [12]. Geometric or semantic attributes has also been proposed in the previous literatures [38][42][37] for visual representation.

In this paper, we propose to complement spatial pooling with orientational pooling, and introduce a novel algorithm, *i.e.*, Orientational Pyramid Matching (OPM).

4. Our Approach

In this section, we first introduce the proposed Orientational Pyramid Matching model, and then present the algorithm of estimating the 3D orientations for image patches.

4.1. Orientational Pyramid Matching

Given a set of patch descriptors that are extracted from interest points or densely-sampled regions, the goal is to summarize then into an image-level feature vector. Different from Spatial Pyramid Matching (SPM) in which each patch descriptor is associated with its spatial position, our approach augments the patch descriptor $\mathbf{f}$ with an additional 3D orientation denoted by the azimuth and polar angles $\mathbf{o} = (\theta, \varphi)^\top$. We denote the set of encoded local features as $\mathcal{S} = \{(\mathbf{f}_1, \mathbf{o}_1), (\mathbf{f}_2, \mathbf{o}_2), \ldots, (\mathbf{f}_M, \mathbf{o}_M)\}$.

The proposed Orientational Pyramid Matching (OPM) algorithm starts with partitioning the set $\mathcal{S}$ into subsets $\{\mathcal{S}_t\}$, $t = 1, 2, \ldots, T_{\mathrm{O}}$, where each subset consists of the patch descriptors that are close in the orientational angles rather than the spatial positions used in Spatial Pyramid Matching (SPM). The partition can be done in various ways, such as clustering the angles. In this paper, we follow the simple way similar to SPM and perform a regular partition scheme, *i.e.*, dividing the orientational space $\mathcal{U} = \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]^2$ into regular grids, which is shown to perform well in practice. Let L_{A} and L_{P} be the numbers of the pyramid layers along the azimuth and polar angles, respectively. The bin in the l-th layer along the azimuth (polar) angle is then of size $\frac{\pi}{2^{\min\{l, L_{\mathrm{A}}\}}} \times \frac{\pi}{2^{\min\{l, L_{\mathrm{P}}\}}}$, *i.e.*, the number of orientational pooling bins in the l-th layer is $2^{\min\{l, L_{\mathrm{A}}\}} \times 2^{\min\{l, L_{\mathrm{P}}\}}$.

Denote the set of partitions produced from orientational pyramid by $\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_{T_{\mathrm{O}}}$. Each region $\mathcal{R}_t$ contains a set of M_t patch descriptors $\{\mathbf{f}_{t,1}, \mathbf{f}_{t,2}, \ldots, \mathbf{f}_{t,M_t}\}$.

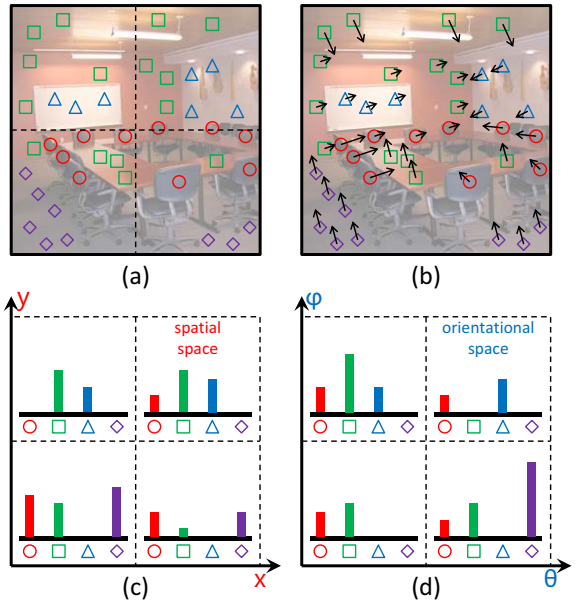

Figure 2. Comparison of the 2nd-layer bins in Spatial Pyramid Matching (SPM) and Orientational Pyramid Matching (OPM). The local features (quantized into 4 types) are grouped according to their spatial positions in (a), and according to their 3D orientations in (b). The SPM pooling results on the (x, y) space and OPM pooling results on the (θ, φ) space are plotted in (c) and (d), respectively. Since the spatial and orientational distributions of each kind of local features are not always similar, the feature distributions on the spatial and orientational spaces might be very different. See the purple diamonds (*floor*) for example.

We aggregate the M_t features together to generate a descriptor $\mathbf{f}_t$ for region $\mathcal{R}_t$. In this paper, we choose to use the max-pooling strategy for the LLC-based features, *i.e.*, $\mathbf{f}_t^{\mathrm{LLC}}[i] = \max_{m=1}^{M_t} \mathbf{f}_{t,m}[i]$, and sum-pooling for the Fisher Vectors, *i.e.*, $\mathbf{f}_t^{\mathrm{Fisher}}[i] = \sum_{m=1}^{M_t} \mathbf{f}_{t,m}[i]$. The overall image feature is then obtained by concatenating the pooled feature vectors of all the regions.

An illustration of Orientational Pyramid Matching and Spatial Pyramid Matching is given in Figure 2. The two schemes are very similar, where the only difference lies in the way of producing pooling regions: OPM uses orientational pooling and SPM uses spatial pooling. The two schemes share many common properties. The feature length is the same if $L_{\mathrm{X}} = L_{\mathrm{A}}$ and $L_{\mathrm{Y}} = L_{\mathrm{P}}$ or $L_{\mathrm{X}} = L_{\mathrm{P}}$ and $L_{\mathrm{Y}} = L_{\mathrm{A}}$, where L_{X} and L_{Y} being the layers along the x and y directions in SPM. The time complexities of computing the SPM and OPM features are the same as each patch feature in one layer is checked only once.

Most techniques extending SPM can also be adopted in extending OPM. For example, we can learn receptive fields [21] or pose pooling kernels [48] based on orientational

pooling. In this paper, we study the performance of Orientational Pyramid Matching with the LLC algorithm [40] and Fisher Vector [31] for encoding patch descriptors.

4.2. Extracting 3D Orientations

We follow the data-driven algorithm [15] for 3D orientation assignment. The KNN criterion is used to judge the planarity of a patch, and predict the 3D orientations of the planar patches.

We use the Bristol dataset in [15] for training and validating the model. Each image in the dataset is equipped with a set of manually labelled landmark points, and a set of regions defined by contouring some of the landmarks. Each region is labeled as planar or non-planer, and each planar region is also annotated with an orientation unit vector $(x, y, z)^\top$, *i.e.*, $x^2 + y^2 + z^2 = 1$. In practice, the scene images are taken with cameras which could be considered as a point light-collector. Therefore the z-component on the planner surface is non-negative, and all the orientation vectors fall onto a unit semi-sphere. This means the azimuth (θ) and polar (φ) angles are enough to represent the 3D orientation: $\theta = \arctan\left(\frac{z}{x}\right)$ and $\varphi = \arcsin(y)$.

We then extract local patches $\{\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_M\}$ densely from each training image. Each patch is assigned into one of the three categories, *i.e.*, planar (it falls completely within a planar region), non-planar (it falls completely within a non-planar region) and boundary (it falls on two or more regions). We denote these categories by C_1, C_2, and C_3, respectively, and define the orientation of the planar patch as the orientation of the corresponding region. In summary, each patch $\mathbf{P}_m$ is represented by the SIFT descriptor $\mathbf{f}_m$, the planar information $c_m \in \{C_1, C_2, C_3\}$ and the orientation (θ_m, φ_m). We collect 100000 patches (50000 planar, 30000 non-planar and 20000 boundary) for the KNN prediction.

Given a new patch $\mathbf{P}$ with the descriptor $\mathbf{f}$, the prediction process finds its K nearest neighbors in the feature space and checks if there are τ enough neighbors supporting the patch $\mathbf{P}$ is planar. In practise, $\tau = \frac{K}{2}$ works very well. If the patch $\mathbf{P}$ is planar, the orientation is then estimated by averaging the orientations of its τ planar neighbors. About half of the patches are classified to be not planar, and they are simply ignored, *i.e.*, not used in feature pooling.

The 3D orientation extraction process is evaluated with two measures: the planarity classification accuracy, and the orientation estimation accuracy. The planarity classification score (**c-score**) is the percentage of correctly classified samples (planar vs. not planar), and the orientation estimation score (**r-score**) is calculated using the cosine similarity between true and estimated orientations. Averaged scores over the densely sampled testing patches are reported. The results with different parameters, K, are listed in Table 1. We choose the best parameter $K = 100$ for prediction in the later experiments.

K	c-score	r-score
0	0.5000	0.5676
1	0.7694	0.6458
5	0.8653	0.6882
10	0.8872	0.6946
50	0.8896	0.6980
100	**0.8902**	**0.6990**
500	0.8899	0.6985
1000	0.8896	0.6968

Table 1. The classification score (c-score) and regression score (r-score) with respect to K, the number of nearest neighbors used for prediction, where $K = 0$ means random guess.

It is worth noting that the Bristol dataset only contains outdoor images. Using outdoor images to train the orientation prediction model for indoor images is questionable. However, experimental results show that the model learnt from Bristol is relatively reliable to estimate orientations, yet we also think the estimation will be better using the same database, if it is labeled with orientations.

5. Experimental Results

In this section, we first provide detailed analysis on the model and parameters with the MIT Indoor-67 dataset, then report our results on several challenging scene and generic classification databases.

5.1. The Dataset and Implementation Details

We have used two scene datasets to evaluate the performance of our algorithm.

The MIT Indoor-67 dataset [32] is the currently largest **indoor** scene recognition dataset, which contains 67 classes and 15620 images. Sample images in this dataset are shown in Figure 1. The SUN-397 dataset [42] is a much larger dataset containing both indoor and outdoor scene categories. There are 397 well sampled scene concepts and more than 100K images in the database. We follow the setting in the previous literatures to choose a fixed number of images for training the classifier, and test it on another fixed set of images to report the average classification accuracy. The numbers of images used for training and testing per category are $(80, 20)$ for the Indoor-67 dataset, and $(50, 50)$ for the SUN-397 dataset. The accuracy is averaged over 10 fixed training/testing splits.

The basic setting in our experiments follows [45] and [34]. Images are resized so that the larger axis has 600 pixels. We use the VLFeat [39] library to extract grayscale RootSIFT descriptors [1]. We extract two sets of SIFT descriptors with the spatial stride and window size equal to $(8, 8)$ and $(16, 16)$, respectively. The 128-D SIFT descriptors are reduced into 64 dimensions in the case of Fisher

$(L_X, L_Y) = (L_A, L_P)$	SPM	OPM	OPM+SPM
$(1,1)$	41.93	41.93	41.93
$(1,2)$	49.10	43.57	52.35
$(1,3)$	54.10	43.75	56.14
$(2,1)$	48.78	46.31	52.98
$(2,2)$	53.55	46.47	56.09
$(2,3)$	55.42	46.48	57.30
$(3,1)$	53.61	48.64	57.20
$(3,2)$	55.09	48.79	58.19
$(3,3)$	57.83	48.83	**59.57**

Table 2. MIT Indoor-67 classification accuracy (%) with different models and parameters. We have used the LLC algorithm in the encoding step.

$(L_X, L_Y) = (L_A, L_P)$	SPM	OPM	OPM+SPM
$(1,1)$	46.63	46.63	46.63
$(1,2)$	57.21	48.25	59.14
$(2,1)$	56.47	50.99	59.55
$(2,2)$	61.22	51.45	**63.48**

Table 3. MIT Indoor-67 classification accuracy (%) with different models and parameters. We have used the Fisher Vectors in the encoding step.

Algorithm	Accuracy
Quattoni *et.al.* [32]	26.0
Li *et.al.* [26]	37.6
Wang *et.al.* [40]	54.62
Xie *et.al.* [45]	57.83
Juneja *et.al.* [23] (BoP)	46.10
Juneja *et.al.* [23] (SPM+BoP)	56.66
Ours (OPM)	48.83
Ours (SPM+OPM)	**59.57**

Table 4. MIT Indoor-67 classification accuracy (%) of our algorithm and previous works without Fisher Vector encoding.

Algorithm	Accuracy
Perronnin *et.al.* [31]	61.22
Kobayashi [24]	58.91
Juneja *et.al.* [23] (BoP)	46.10
Juneja *et.al.* [23] (SPM+BoP)	63.10
Ours (OPM)	51.45
Ours (SPM+OPM)	**63.48**

Table 5. MIT Indoor-67 classification accuracy (%) of our algorithm and previous works with Fisher Vector encoding.

Vector encoding. For LLC encoding, we train a codebook with 8192 codewords using K-Means clustering, while for Fisher Vector encoding, we train a codebook with 256 centers using the Gaussian Mixture Model (GMM). The number of descriptors collected for clustering is around 5 million. For LLC encoding, we adopt the GPP [45] algorithm to enhance the local features with geometric visual phrases. Both SPM and OPM algorithms are used to capture the spatial layouts of the scene images. The number of layers of SPM is 3 for LLC encoding and 2 for Fisher Vector encoding. We use the LibLINEAR toolbox [9] as a scalable SVM implementation. The penalty parameter C is set to 10.

5.2. Model and Parameters

Here we enumerate different combinations of L_A and L_P, and report classification accuracies on the MIT Indoor-67 dataset with three different features, *i.e.*, SPM features, OPM features, and the concatenation of SPM and OPM features (denoted as OPM+SPM). Please note that we are always using $L_X = L_A$ and $L_Y = L_P$, which results in the same length of spatial and orientational features vectors, *i.e.*, we are using the same amount of information to model the spatial and orientational contexts.

The results are listed in Table 2 and 3. We can observe that the classification accuracy grows with the number of spatial and orientational pooling bins. To prevent the feature vectors from becoming too long, we select $L_X = L_Y = L_A = L_P = 3$ with LLC features, and $L_X = L_Y = L_A = L_P = 2$ with Fisher Vectors. One can observe that OPM produces inferior classification results to SPM. When we concatenate the feature vectors produced by SPM and OPM, we achieve higher accuracies than using SPM and OPM features alone. This gives us the evidence that features provided by SPM and OPM are complementary to each other.

5.3. Comparison with Previous Works

We compare our algorithm with some previous works on the MIT Indoor-67 dataset. Since Fisher Vector encoding [31] is verified very efficient in this dataset, we list the algorithms without and with Fisher Vector encoding in Table 4 and Table 5, respectively, and compare our algorithms (without and with Fisher Vector encoding) with them. We can see that in both cases, our algorithm achieves the state-of-the-art classification accuracy. The improvement over the second best without using Fisher vectors, as shown in Table 4, is around 2%, which is not trivial in the challenging indoor scene recognition task. As shown in Table 5, a Bag-of-Parts (BoP) model is proposed in the very recently-published approach [23], and the concatenation of the BoP and SPM features improves the performance of the SPM features from 60.77% to 63.10%. Our approach improves the performance of SPM features from 61.22% to 63.48%. In addition, the performance from the BoP features is only 46.10%, which is significantly lower than 51.45% reported by OPM features. This shows that OPM provides more

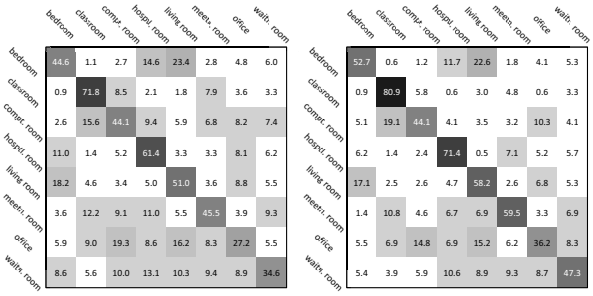

Figure 3. The difference between the confusion matrices without (left) and with (right) OPM features. We have used the LLC-based features for classification.

category A	category B	OPM≻	Comb≻
bookstore	*library*	+6.69%	+3.75%
auditorium	*concert hall*	+5.35%	+3.93%
computer room	*office*	+4.40%	+1.80%
jewellery shop	*lobby*	+2.90%	+1.06%
bakery	*buffet*	+2.88%	+2.18%

category A	category B	SPM≻	Comb≻
gameroom	*garage*	+4.78%	+1.95%
rest. kitchen	*studiomusic*	+4.64%	+2.33%
library	*clothingstore*	+4.62%	+3.40%
children room	*kindergarden*	+4.47%	+1.91%
laboratorywet	*kindergarden*	+4.47%	+2.48%

Table 6. Categorie pairs better distinguished by OPM (top) and SPM (bottom) features. For example, the confusion value from *bookstore* to *library* is reduced by 6.69% by using OPM features instead of SPM features, and 3.75% by using combined features instead of SPM features.

complementary information to SPM than the BoP model. In the future, we will investigate the combination of OPM with several advanced features such as BoP.

5.4. Empirical Analysis

We present empirical comparisons of the SPM and OPM features in scene recognition. Let us first look at 8 categories: *bedroom*, *classroom*, *computer room*, *hospital room*, *living room*, *meeting room*, *office* and *waiting room*, which are the case that SPM features cannot discriminate very well. The confusion matrices of the SPM features and of the combination of SPM and OPM features are shown in the left and right part of Figure 3, respectively. One can see that most of the off-diagonal elements of the right confusion matrix are smaller, *e.g.*, the confusion value from *computer room* to *hospital room* is reduced from 9.4% to 4.1%. In addition, it can also be seen that most of the diagonal elements of the right confusion matrix are larger, *e.g.*, the classification accuracy of *meeting room* increases to 45.5% from 59.5%. The above analysis suggests that OPM is ca-

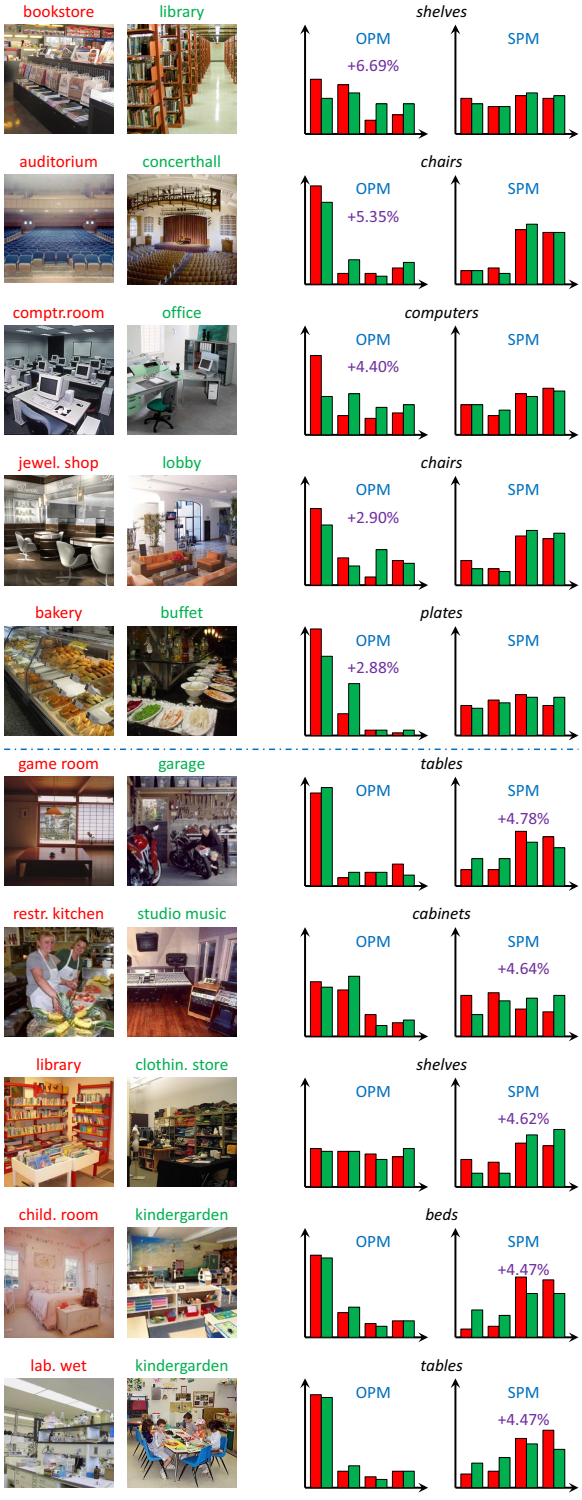

Figure 4. Category pairs with largest differences in confusion values (best viewed in color PDF). The 2nd-layer spatial and orientational distributions of a common object are plotted as well. The red and green histogram bins correspond to the category with the same color of title.

Algorithm	Accuracy
Xiao *et.al.* [42]	38.0
Sanchez *et.al.* [34]	43.2
Ours (SPM)	43.58
Ours (OPM)	34.61
Ours (SPM+OPM)	**45.91**

Table 7. SUN-397 classification accuracy (%) of our algorithm and previous works.

Algorithm	Accuracy
Chatfield *et.al.* [6]	77.78
Jia *et.al.* [21]	75.3
Ours (SPM)	80.73
Ours (OPM)	65.59
Ours (SPM+OPM)	**81.75**

Table 8. Caltech101 classification accuracy (%) of our algorithm and previous works.

pable of producing the complementary information that is ignored by SPM.

We also check those pairs of categories that are difficult to discriminate using single SPM or OPM features. Top and bottom parts of Table 6 show the pairs on which the performance improvements of OPM over SPM are the largest, and those improvements of SPM over OPM are the largest, respectively. To observe the spatial and orientational distributions of objects in these categories, we manually labeled about 20 objects in the images from the corresponding categories, and find the most frequent visual word (in a 8-entry codebook) in the labeled regions to represent the object. Then the spatial and orientational histograms are calculated automatically using the visual words. Results are shown in Figure 4. One can see that for the pairs on which OPM performs better the histograms from orientations are more discriminative, while for the pairs on which SPM performs better the histograms from locations are more discriminative. This indicates that the complementary information of OPM indeed comes from the orientational distribution. The better performance of their combination is because of the complementary benefits of SPM and OPM.

5.5. Large-scale and General Cases

To evaluate the scalability of our model, we test it on the SUN-397 dataset [42]. We report the classification performance with Fisher vectors in Table 8. One can observe again in this case, that OPM provides complementary orientational information, which helps our classifier beat the competitors using only spatial information. It is also worth noting that the SUN-397 dataset contains a number of outdoor scene concepts. We can therefore conclude that the OPM model designed for recognizing indoor scenes could be generalized to the outdoor scene categories.

We also report the results using OPM features on the Caltech101 dataset [10], a widely adopted database for generalized object categorization. It is a little surprising to observe that OPM, a model designed for indoor scenes, also provides useful information for classifying general objects (combined model obtains 1.02% accuracy gain). The improvement looks small, but is not easy for such datasets. We believe that OPM benefits from the orientational features extracted on the background regions, *e.g.*, a car often appears in an outdoor scene, while a TV is more likely to be put in a living room. This suggests that OPM could also provide auxiliary cues in generic classification.

6. Conclusions

In this paper, we discuss the use of 3D orientational features in scene classification tasks. We propose a novel Orientational Pyramid Matching (OPM) algorithm to capture the orientational contexts in the images, and combine the OPM features with SPM features to capture the complementary information for scene recognition. State-of-the-art classification performance is achieved on both MIT Indoor-67 and SUN-397 datasets. In the future, we will investigate the combination of OPM with many other approaches, and look forward to some more accurate orientation assignment algorithms to improve the OPM performance.

7. Acknowledgements

This work was supported by the National Basic Research Program (973 Program) of China (Grant Nos. 2013CB329403, 2012CB316301 and 2014CB347600), the National Natural Science Foundation of China (Grant Nos. 61128007, 61332007, 61273023 and 91120011), the Beijing Natural Science Foundation (Grant No. 4132046), and the Tsinghua University Initiative Scientific Research Program (Grant No. 20121088071). This work was also supported in part to Dr. Qi Tian by ARO grant W911NF-12-1-0057, Faculty Research Awards by NEC Laboratories of America, and 2012 UTSA START-R Research Award, respectively.

References

[1] R. Arandjelovic and A. Zisserman. Three Things Everyone Should Know to Improve Object Retrieval. *Computer Vision and Pattern Recognition*, 2012.

[2] D. H. Ballard. Generalizing the Hough Transform to Detect Arbitrary Shapes. *Pattern Recognition*, 1981.

[3] A. Bartoli and P. Sturm. Constrained Structure and Motion from Multiple Uncalibrated Views of a Piecewise Planar Scene. *International Journal of Computer Vision*, 2003.

[4] A. Bosch, A. Zisserman, and X. Muoz. Image Classification using Random Forests and Ferns. *International Conference on Computer Vision*, 2007.

[5] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning Mid-Level Features for Recognition. *Computer Vision and Pattern Recognition*, 2010.

[6] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The Devil is in the Details: An Evaluation of Recent Feature Encoding Methods. *British Machine Vision Conference*, 2011.

[7] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual Categorization with Bags of Keypoints. *Workshop on Statistical Learning in Computer Vision, ECCV*, 2004.

[8] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. *Computer Vision and Pattern Recognition*, 2005.

[9] R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin. LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research*, 2008.

[10] 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*, 2007.

[11] L. Fei-Fei and P. Perona. A Bayesian Hierarchical Model for Learning Natural Scene Categories. *Computer Vision and Pattern Recognition*, 2005.

[12] J. Feng, B. Ni, Q. Tian, and S. Yan. Geometric ℓ_p-norm Feature Pooling for Image Classification. *Computer Vision and Pattern Recognition*, 2011.

[13] S. Gao, I. W. Tsang, L.-T. Chia, and P. Zhao. Local features are not lonely – laplacian sparse coding for image classification. *Computer Vision and Pattern Recognition*, 2010.

[14] K. Grauman and T. Darrell. The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features. *IEEE International Conference on Computer Vision*, 2005.

[15] O. Haines and A. Calway. Detecting Planes and Estimating their Orientation from a Single Image. *British Machine Vision Conference*, 2012.

[16] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the spatial layout of cluttered rooms. *International Conference on Computer vision*, 2009.

[17] V. Hedau, D. Hoiem, and D. Forsyth. Thinking Inside the Box: Using Appearance Models and Context Based on Room Geometry. *European Conference on Computer Vision*, 2010.

[18] D. Hoiem, A. A. Efros, and M. Hebert. Recovering Surface Layout From an Image. *International Journal of Computer Vision*, 2007.

[19] J. Illingworth and J. Kittler. A Survey of the Hough Transform. *Computer Vision, Graphics, and Image Processing*, 1988.

[20] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating Local Descriptors Into a Compact Image Representation. *Computer Vision and Pattern Recognition*, 2010.

[21] Y. Jia, C. Huang, and T. Darrell. Beyond Spatial Pyramids: Receptive Field Learning for Pooled Image Features. *Computer Vision and Pattern Recognition*, 2012.

[22] Y. Jiang, J. Yuan, and G. Yu. Randomized Spatial Partition for Scene Recognition. *European Conference on Computer Vision*, 2012.

[23] M. Juneja, A. Vedaldi, C. Jawahar, and A. Zisserman. Blocks that Shout: Distinctive Parts for Scene Classification. *Computer Vision and Pattern Recognition*, 2013.

[24] T. Kobayashi. BoF meets HOG: Feature Extraction based on Histograms of Oriented pdf Gradients for Image Classification. *Computer Vision and Pattern Recognition*, 2013.

[25] S. Lazebnik, C. Schmid, and J. Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. *Computer Vision and Pattern Recognition*, 2006.

[26] L.-J. Li, H. Su, L. Fei-Fei, and E. P. Xing. Object Bank: A High-Level Image Representation for Scene Classification & Semantic Feature Sparsification. *Advances in Neural Information Processing Systems*, 2010.

[27] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal on Computer Vision*, 2004.

[28] B. Micusik, H. Wildenauer, and M. Vincze. Towards Detection of Orthogonal Planes in Monocular Images of Indoor Environments. *International Conference on Robotics and Automation*, 2008.

[29] Z. Niu, G. Hua, X. Gao, and Q. Tian. Context Aware Topic Model for Scene Recognition. *Computer Vision and Pattern Recognition*, 2012.

[30] A. Oliva and A. Torralba. Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. *International Journal of Computer Vision*, 2001.

[31] F. Perronnin, J. Sánchez, and T. Mensink. Improving the Fisher Kernel for Large-scale Image Classification. *European Conference on Computer Vision*, 2010.

[32] A. Quattoni and A. Torralba. Recognizing Indoor Scenes. *Computer Vision and Pattern Recognition*, 2009.

[33] C. Rother. A New Approach to Vanishing Point Detection in Architectural Environments. *Image and Vision Computing*, 2002.

[34] J. Sánchez, F. Perronnin, T. Mensink, and J. J. Verbeek. Image Classification with the Fisher Vector: Theory and Practice. *International Journal of Computer Vision*, 2013.

[35] M. Scherer, M. Walter, and T. Schreck. Histograms of Oriented Gradients for 3D Object Retrieval. *International Conferences in Central Europe on Computer Graphics, Visualization and Computer Vision*, 2010.

[36] P. Scovanner, S. Ali, and M. Shah. A 3-Dimensional SIFT Descriptor and Its Application to Action Recognition. *ACM Multimedia*, 2007.

[37] Y. Su and F. Jurie. Visual Word Disambiguation by Semantic Contexts. *International Conference on Computer Vision*, 2011.

[38] M. Ullah, S. N. P., and I. Laptev. Improving Bag-of-Features Action Recognition with Non-Local Cues. *British Machine Vision Conference*, 2010.

[39] A. Vedaldi and B. Fulkerson. VLFeat: An Open and Portable Library of Computer Vision Algorithms. *ACM Multimedia*, 2010.

[40] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained Linear Coding for Image Classification. *Computer Vision and Pattern Recognition*, 2010.

[41] P. Wang, J. Wang, G. Zeng, W. Xu, H. Zha, and S. Li. Supervised Kernel Descriptors for Visual Recognition. *Computer Vision and Pattern Recognition*, 2013.

[42] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. SUN Database: Large-scale Scene Recognition from Abbey to Zoo. *Computer Vision and Pattern Recognition*, 2010.

[43] L. Xie, Q. Tian, R. Hong, S. Yan, and B. Zhang. Hierarchical Part Matching for Fine-Grained Visual Categorization. *International Conference on Computer Vision*, 2013.

[44] L. Xie, Q. Tian, M. Wang, and B. Zhang. Spatial Pooling of Heterogeneous Features for Image Classification. *IEEE Transactions on Image Processing*, 2014.

[45] L. Xie, Q. Tian, and B. Zhang. Spatial Pooling of Heterogeneous Features for Image Applications. *ACM Multimedia*, 2012.

[46] L. Xie, Q. Tian, and B. Zhang. Feature Normalization for Part-based Image Classification. *International Conference on Image Processing*, 2013.

[47] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear Spatial Pyramid Matching using Sparse Coding for Image Classification. *Computer Vision and Pattern Recognition*, 2009.

[48] N. Zhang, R. Farrell, and T. Darrell. Pose Pooling Kernels for Subcategory Recognition. *Computer Vision and Pattern Recognition*, 2012.

[49] X. Zhou, K. Yu, T. Zhang, and T. S. Huang. Image Classification using Super-Vector Coding of Local Image Descriptors. *European Conference on Computer Vision*, 2010.