Combining Orientational Pooling Features for Scene Recognition

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, 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. The Bag-of-Features Model

The BoF model is 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 [8] or HOG [2], 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. Feature encoding can also be sparse [13] or high-dimensional [9]. Spatial pooling consists of partitioning an image into a set of regions, aggregating feature-level statistics over these regions [18], and normalizing then concatenating the region descriptors as an image-level feature vector [16]. Image partition can be obtained by Spatial Pyramid Matching (SPM) [7]. Aggregation of descriptors within a region is often performed with a pooling strategy.

2. 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.

2.1. 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 angles 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 $\{\mathbf{f}_{t,1}, \mathbf{f}_{t,2}, \ldots, \mathbf{f}_{t,M_t}\}$. We aggregate the M_t features together to generate a descriptor $\mathbf{f}_t$ for region $\mathcal{R}_t$. The overall image feature is then obtained by concatenating the pooled feature vectors of all the regions.

2.2. Extracting 3D Orientations

We follow the data-driven algorithm [4] 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 [4] 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-planar, and each planar region is also annotated with an orientation unit vector $(x, y, z)^\top$, $z \geqslant 0$. We use the azimuth (θ) and polar (φ) angles 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 intersects with 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 that 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 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 neighbors supporting the patch $\mathbf{P}$ is planar. In practice, $K = 100$ and $\tau = \frac{K}{2}$ works very well. If the patch $\mathbf{P}$ is planar, the orientation is then estimated by averaging the orientations of its 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.

3. Experimental Results

We evaluate our algorithm on two scene datasets, the MIT Indoor-67 dataset [10] and the SUN-397 dataset [14].

The basic setting follows [11]. Images are resized so that the larger axis has 600 pixels. We use VLFeat [12] to extract RootSIFT descriptors [1]. The spatial stride and window size are $(8, 8)$ and $(16, 16)$, respectively. The 128-D descriptors are reduced into 64 dimensions by PCA. We train a GMM with 256 centers by collecting around 5 million descriptors for clustering. Fisher vectors [9] are extracted as the image-level feature. 2-layer SPM and OPM are used. We use LibLINEAR [3] as a scalable SVM implementation. The penalty parameter C is set to 10.

The comparison with previous algorithms is listed in Table 1. It is verified that OPM provides complementary information to SPM, which is useful for scene understanding. For more details, please refer to our CVPR paper [17].

Algorithm	Indoor-67	SUN-397
Xie *et.al.* [15]	57.83	–
Perronnin *et.al.* [9]	61.22	–
Kobayashi [6]	58.91	–
Juneja *et.al.* [5] (SPM+BoP)	63.10	–
Xiao *et.al.* [14]	–	38.0
Sanchez *et.al.* [11]	–	43.2
Ours	**63.48**	**45.91**

Table 1. Performance comparison with previous methods.

References

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

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

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

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

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

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

[7] 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.

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

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

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

[11] 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.

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

[13] 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.

[14] 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.

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

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

[17] L. Xie, J. Wang, B. Guo, B. Zhang, and Q. Tian. Orientational Pyramid Matching for Recognizing Indoor Scenes. *Computer Vision and Pattern Recognition*, 2014.

[18] Y. Zuo and B. Zhang. Robust Hierarchical Framework for Image Classification via Sparse Representation. *Tsinghua Science & Technology*, 2011.