Spatial-Bag-of-Features

Yang Cao[1]*, Changhu Wang[2], Zhiwei Li[2,3], Liqing Zhang[4], Lei Zhang[2]
[1,3,4]MOE-Microsoft Key Laboratory for Intelligent Computing and Intelligent Systems, Shanghai Jiao Tong University
[2]Microsoft Research Asia, Beijing, China
[1]cybersun.sjtu@gmail.com, [2]{chw, zli, leizhang}@microsoft.com, [4]zhang-lq@cs.sjtu.edu.cn

Abstract

In this paper, we study the problem of large scale image retrieval by developing a new class of bag-of-features to encode geometric information of objects within an image. Beyond existing orderless bag-of-features, local features of an image are first projected to different directions or points to generate a series of ordered bag-of-features, based on which different families of spatial bag-of-features are designed to capture the invariance of object translation, rotation, and scaling. Then the most representative features are selected based on a boosting-like method to generate a new bag-of-features-like vector representation of an image. The proposed retrieval framework works well in image retrieval task owing to the following three properties: 1) the encoding of geometric information of objects for capturing objects' spatial transformation, 2) the supervised feature selection and combination strategy for enhancing the discriminative power, and 3) the representation of bag-of-features for effective image matching and indexing for large scale image retrieval. Extensive experiments on 5000 Oxford building images and 1 million Panoramio images show the effectiveness and efficiency of the proposed features as well as the retrieval framework.

1. Introduction

In recent years, large-scale image retrieval is receiving increasingly significant attention owing to its great potential in both industry applications and research problems [12, 13]. Inspired by the success of Web search, most existing works represent images by bag-of-features (BOF) models and index histogram features of images by inverted files [2, 5, 9, 12, 13, 14]. Although this framework has demonstrated to be simple and efficient, it still suffers from the accuracy and scalability problems, which are very important to many computer vision problems [16, 5].

To improve retrieval accuracy, many approaches have been proposed, *e.g*. large vocabularies [9, 12], soft quantization [11], and query expansion [2]. A major limitation of such approaches is that they often ignore spatial information of local features, which has been observed very helpful in improving retrieval accuracy [12]. To overcome this limitation, several research attempts have been made to utilize spatial information to improve retrieval accuracy. Among these attempts, RANSAC-based image reranking achieved the state-of-the-art result in terms of retrieval accuracy [2, 12]. However, to rerank top image search results, it requires random access to raw features of these images, and inevitably increases the memory cost and slow down the search speed. "Bundling features" can address this problem to some extent by encoding local spatial information in stable regions in inverted index [14]. But the ranking process requires feature order information, which makes distance measure no longer an L1 or L2 distance, and thus cannot be further accelerated by indexing technologies such as locality sensitive hashing[1]. Spatial pyramid matching [6] and visual phrase [15] encode spatial information to inverted index from another way by either enforcing the spatial distributions of local features belonging to the same category to be globally coherent, or considering local adjacency of visual words. However, as the encoded spatial information is too weak, the search precision is not as good as that of the RANSAC-based approach.

As an effective technology proven by web search engines *e.g*. Google and large-scale image search [9, 12], inverted index has another important property. That is, it is essentially a representation of high dimensional sparse vectors, and provides an efficient mechanism for fast cosine similarity computation. This property makes it possible to further improve its efficiency and scalability by index compression or dimension reduction. For example, [5] proposed an efficient way to compress inverted files, and [16] proposed a framework to efficiently approximate $cosine$ similarity computation by conducting dimension reduction and leveraging residual error information.

Based on these studies, in this work we develop a new class of features for large-scale image retrieval to meet two design goals: 1) the new features should have the same format as current bag-of-features (i.e. histogram-like features), and 2) the new features are able to effectively encode spatial information. The first goal is to guarantee that the new features can be indexed by mature inverted index techniques. Therefore, if we achieve these goals, we are able to effectively and efficiently organize local features and their spatial relationships in one inverted index. As a result, the system can search images accurately and fast.

The basic idea could be supported by the spatial properties of local features of images. As shown in Fig. 1, scene

*This work was performed at Microsoft Research Asia.

978-1-4244-6983-3/10/$26.00 ©2010 IEEE

Figure 1. Illustrations of embedded spatial configurations in objects and natural scenes

images, *e.g*. buildings or sea, have horizontal and vertical relationships among local features, while objects, *e.g*. sun and flowers, have circle-like relationships. Therefore, projecting features onto certain lines or circles are able to capture basic geometric information in images. In this way, we obtain a kind of so-called *ordered bag-of-features*. This is a generalization of the spatial pyramid matching idea [6]. However, in terms of utilizing of spatial constraints, these features are too rigorous to handle typical transformations of objects, *i.e*., translation, rotation, scaling. Therefore, we further process the ordered bag-of-features to obtain a kind of so-called *spatial bag-of-features* by some operations for histogram features, *i.e*. calibration, equalization and decomposition.

To tune parameters and select the most effective features, a boosting-based method is introduced. Since the proposed spatial-bag-of-features are in the same format as the traditional bag-of-features, we adopt the inverted file technique to index images [13, 12]. Without increasing the memory cost of index, all information used in ranking is able to be packed in inverted files. This property guarantees the ranking could be accomplished instantly. Extensive experiments on benchmark datasets show the effectiveness and efficiency of the proposed spatial bag-of-features as well as the retrieval framework.

2. Ordered Bag-of-Features

In this section, we introduce two families of ordered bag-of-features which can weakly capture some geometric information of images. These representations are the foundations of the spatial bag-of-features which will be introduced in the next section. It should be noted that *features* refers to quantified local descriptors in this work.

2.1. Motivations

Our target is to design bag-of-features-like representations for images to 1) encode objects' geometric information, and 2) enable efficient retrieval. On the one hand, the orderless bag-of-features totally ignore geometric relationships of local descriptors. On the other hand, the two-dimension spatial information of local descriptors of an image is difficult to be directly encoded into the bag-of-features models. To address this problem, we propose to project the local descriptors which reside on a two-dimensional space to a one-dimensional space. The projected features could weakly capture the geometric information of objects, while they are still a kind of bag-of-features.

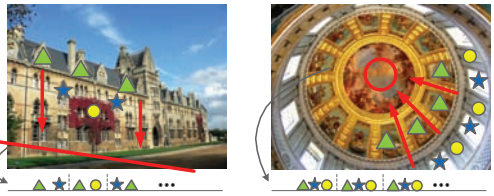

Figure 2. Illustration of ordered bag-of-features generated by linear and circular projections. Markers represent quantified local features. From (a) linear projection or (b) circular projection, certain geometric information could be preserved to some extent.

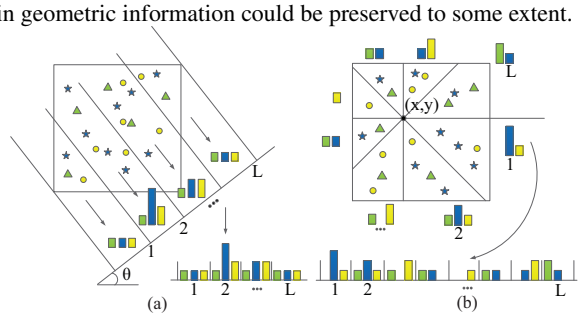

Figure 3. Toy examples of constructing linear and circular ordered bag-of-features. Stars, triangles, and circles represent three kinds of features. (a) Linear projection: all features are projected onto a line with angle θ and resolution $L = 4$, and then the features within each spatial bin are counted. (b) Circular projection: we locate the center at (x, y) and evenly divide the space into $L = 8$ sectors, and then count features within each sector.

Two projection strategies, i.e. the linear projection and the circular projection, are used to generate the ordered bag-of-features according the following reasons:

1) Line and circle are two basic elements to represent an object. Both natural objects, *e.g*. mountains, sun and flowers, and man-made objects, *e.g*. buildings, windows and chairs, could be simply sketched out using lines and circles.

2) The ordered bag-of-features based on these two kinds of projections could reflect some basic geometric information of objects. As shown in Fig. 2, the linear projection captures the intrinsic order of features in one direction; the circular projection preserves the feature alignment in a polar coordinate system.

3) The ordered bag-of-features have the same representations as the traditional bag-of-features, which could serve as the foundations of spatial bag-of-features introduced in the next section.

2.2. Linear ordered bag-of-features

Locality is one of the fundamental spatial information to depict the configuration of an image. As shown in Fig. 3(a), we project features in a two-dimensional space (i.e. the image plane) onto a line with an arbitrary angle, by which the locality of each feature is transformed to a one-dimensional coordinate along the line. Inspired by "subdivide and disorder" techniques [4, 6], we divide this line into equal seg-

ments[1]. Each segment is considered as a *bin*, and a histogram statistics (or say *sub-histogram*) is leveraged to represent the features inside this bin. All L bins are connected to be a long *histogram*, which is named as *linear ordered bag-of-features*. This projection has two degrees of freedom, i.e. the angle θ, which represents the specific orientation we want to preserve, and the number of bins L, which control the resolution of dividing the line. Based on this method, a long histogram with L connected sub-histograms could be generated for each image, which encodes the rough locality information along the direction of θ.

By enumerating different angles and resolution levels, we get a family of linear ordered bag-of-features. Obviously, the traditional orderless bag-of-features is a special case of this representation with L being 1. The spatial pyramid matching (SPM) [6] could be considered as a combination of a set of this kind of features, *i.e.* with vertical ($\theta = 90°$) and horizontal ($\theta = 0°$) projections under some resolution levels. Moreover, this representation could also capture other slantwise directions that SPM cannot handle.

2.3. Circular ordered bag-of-features

In order to capture the geometric information of object sketched by more complex curves, and tolerate object rotation variance[2], circular projection is used to design a new family of ordered bag-of-features. As shown in Figure 3(b), after locating a center, it evenly divides the two-dimensional space into sectors with the same radian. Similar to linear projection, each sector is considered as a *bin* and a *sub-histogram* is used to represent the features in the sector. This projection has two parameters, i.e. the center (x, y) and the number of bins L. By this mean, the locality relationship in the polar coordinate system with its focus at (x, y) could be captured from a circular projection, and the locality precision in this polar coordinate system is determined by L. In the same manner as linear projection, different centers and resolutions are enumerated to deal with multiple situations.

2.4. Image matching using ordered bag-of-features

To simplify the notations, we use Θ to represent the parameters of linear projection $\{L, \theta\}$ and circular projection $\{L, (x, y)\}$. For any *histogram* H^Θ generated by either a linear or a circular projection with parameter Θ and resolution L, it is concatenated by L sub-histograms:

$$H^\Theta = [h^{1,\Theta}, h^{2,\Theta}, \cdots, h^{L,\Theta}] \quad (1)$$

where $h^{i,\Theta}$ is the sub-histogram in the i-th bin of the projection parameterized by Θ. Let P and Q be two images that need to be compared. Their similarity under this feature is defined as:

$$< H_P^\Theta, H_Q^\Theta >= \sum_{i=1}^{L} Sim(h_P^{i,\Theta}, h_Q^{i,\Theta}) \quad (2)$$

[1]The start and the end of this line are the projective points of the left-most and right-most corners (edges) of the original image.

[2]See Section 3.2 for details of encoding rotation variance based on circular ordered bag-of-features.

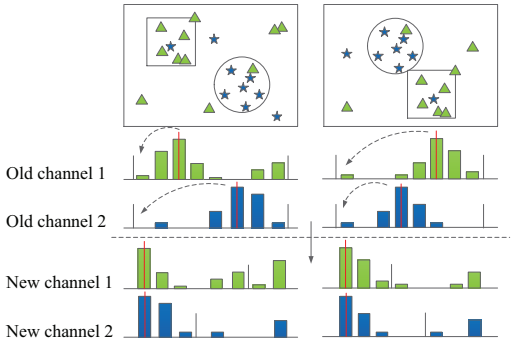

Figure 4. Illustration of the mismatching of two images using linear ordered bag-of-features caused by translation. Squares and circles in the two images represent two objects. Before calibration, histograms on each feature channel are quite dissimilar. After histogram calibration, the spatial bag-of-features of two images become much more similar.

where $Sim(\cdot, \cdot)$ could be any histogram similarity measure, e.g. *cosine* similarity or histogram intersection. Since the histograms are very sparse, these operations are very efficient. The computational complexity is linear to the number of features [6]. It is noted that we only measure the similarity between histograms generated from an identical projection. By enumerating multiple projections, a family of histogram representations could be obtained for each image.

3. Spatial Bag-of-Features

Although the ordered bag-of-features could encode basic spatial information of local descriptors of an image, they are too rigorous to tolerate different spatial variations of objects. For example, as shown in Fig. 4, 5 and 6, they will fail to match two images with object translation, rotation, or scaling, *etc*. Therefore, based on the two families of ordered bag-of-features, we propose three variant features which are designed to tolerate the variances on features caused by object translation, rotation, and scaling, respectively. Moreover, we introduce a strategy to avoid heavy clusters and conflicts between our representations. The new feature representations are named as *spatial bag-of-features*.

It should be noted that different spatial bag-of-features are designed for tolerating different spatial variance. Each parameterized spatial bag-of-features could be hoped to weakly capture certain aspect of the geometric information of objects. We will rely on a surprised mechanism to select a powerful subset of features to compose the final bag-of-features-like representations to handle complex cases.

3.1. Translation invariance

From Fig. 4 we can see that, if one object is located in different positions of two images, the same visual features of the object will be located into different bins for the two images using aforementioned linear projection method. To make the proposed spatial bag-of-features more robust to object translation, we adopt a kind of histogram calibration strategy to encode the translation invariance into the spacial

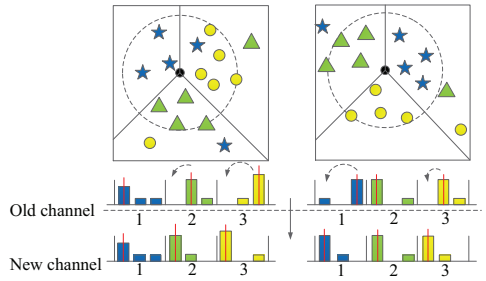

Figure 5. Illustration of the mismatching of two images using circular ordered bag-of-features caused by rotation. A circular object is composed of three major features denoted by triangles, circles and stars. Due to a rotation transformation, corresponding parts locate in different bins. After calibration, all corresponding sectors are matched.

bag-of-features.

Let H^{Θ} denote a histogram generated by a linear projection. If there are V features in the quantified dictionary, there will be V *feature channels*[3] with length L, where L is the number of bins in H^{Θ}. For each feature v, its channel is denoted by H_v^{Θ}:

$$H_v^{\Theta} = [h_v^1, h_v^2, \ldots, h_v^{m-1}, h_v^m, h_v^{m+1}, \cdots, h_v^L,] \quad (3)$$

where h_v^i is the term frequency of word v in bin i (We use h_v^i to denote $h_v^{i,\Theta}$ for short). Denote $m = \arg\max_i\{h_v^i\}$. We reorder this vector by making it start from the position m to get a new histogram as follows:

$$T_v^{\Theta} = [h_v^m, h_v^{m+1}, \cdots, h_v^{L-1}, h_v^L, h_v^1, \ldots, h_v^{m-1}] \quad (4)$$

A new histogram T^{Θ} could be obtained by grouping $T_v^{\Theta}, v = 1, 2, ..., V$ to be one new long histogram by the inverse process of extracting $H_v^{\Theta}, v = 1, 2, ..., V$ from H^{Θ}. By this new representation (see Fig. 4 for an illustration), two images with the same object at different positions could have similar distribution on their feature channels.

3.2. Rotation invariance

Similar to the histogram calibration of linear ordered bag-of-features, we can also calibrate the histogram of circular ones to get a new representation to deal with rotation transformation, which is denoted as R^{Θ}. We omit the generating process of R^{Θ} due to space limitation, since it is similar to Section 3.1. See Fig. 5 for an illustration.

3.3. Scaling invariance

Considering two images containing an identical object with different sizes, if we project them onto a line, the two histograms will have similar distribution curves but with different widths, which causes that the same visual features will fall into different bins. In order to make our features robust to object scaling, a kind of histogram equalization technique is adopted.

[3]In this work, a *feature channel* represents the sub-histogram extracted from the whole histogram if we only considered the distribution of one specific feature (or say visual word).

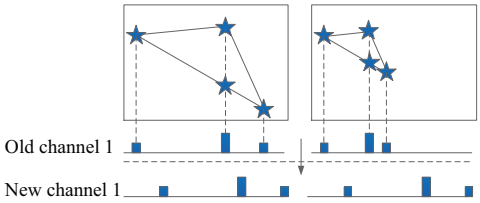

Figure 6. Illustration of the mismatching of two images using linear ordered bag-of-features caused by diverse scales. A triangle is composed of 4 local features represented by stars. Due to different scales, only the left-most feature is matched. By equalization, the histograms expand to the whole space (with the same distribution) and all the corresponding parts are matched.

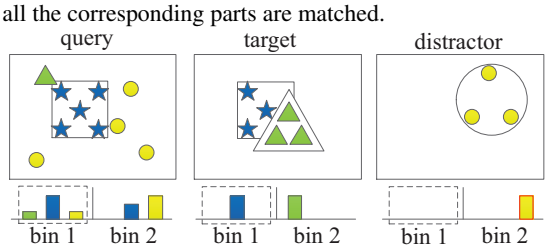

Figure 7. Illustration of false positive caused by heavy clutters. The left image contains a query object (square) and clutters. If we adopt the global histogram, both the target and the distractor have three common features with the query picture. By only selecting the first bin, the distractor is excluded.

We follow the notations in Section 3.1. The term frequency h_v^t which originally belongs to bin t is relocated to bin s by $s = \lceil \frac{\sum_{i=1}^{t} h_v^i}{\sum_{i=1}^{L} h_v^i} \rceil$, by which adjacent bins could be merged together in some feature channels. In this way, we can obtain a new histogram S^{Θ}, in which the distribution of each feature channel extends to the whole space and the new spatial bag-of-features are less sensitive to scaling. As shown in Fig. 6, after equalization, all the parts of the identical object in different sizes are matched.

We can also apply this scheme to circular ordered bag-of-features, which is omitted due to space limitation.

3.4. Long histogram decomposition

Although we have encoded the invariance of different spatial transformations into different spatial bag-of-features, each of them is actually designed for one single case and thus could be considered as a weakly spatial-transformation invariant feature of an whole image. When combined together, if severe conflict exists, the total descriptive power may degenerate. The main reason is that it is still too strict to apply a single rule on the whole image. Our solution is not to select the whole connected histogram, instead, to directly find a combination of individual bins, which is still a histogram. The similarity measure of the final representation is given by:

$$< \mathcal{H}_{\mathcal{P}}, \mathcal{H}_{\mathcal{Q}} >= \sum_{\Phi \in \mathcal{S}} \alpha^{\Phi} sim(h_P^{\Phi}, h_Q^{\Phi}) \quad (5)$$

where $\Phi = \{\Theta, k(k <= L)\}$, in which L is the number of bins of the projection (with or without encoding invariance) parameterized by Θ, and k is the id of the bin. S represents the selected projection set learnt by a supervised manner.

This new mechanism is more flexible, since it decomposes the global spatial constraint into several partial spatial constraints. However, it is not a compromise, for it still has the ability to present the global one if all bins of a projection are selected. In fact, it becomes stronger to describe more complex cases and has a larger chance to avoid conflicts between different types of spatial bag-of-features.

Another advantage is that it discards some insignificant information, which could simplify the final representation and speed up the system a lot. This method has its practical explanation. In real retrieval tasks, heavy clutters or occlusions are unavoidable. This strategy has potential to make the final representation be more concentrated on potential target parts, while neglects meaningless or distractive parts. See Fig. 7 for an illustration.

4. Retrieval Framework

In this section we introduce the overall framework of our retrieval system in detail.

4.1. Feature extraction and quantization

We adopt similar local features as other retrieval systems [12, 14]. To detect salient regions, we adopt the affine-invariant region detection techniques proposed in [8]. For each detected region, a 128D SIFT descriptor[7] is computed. Generally, a high resolution image (*e.g.* 1024×768) produces around 3000 local features.

The quality of feature quantization is quite important for a retrieval system [12, 11]. We use the approximate *k-means* proposed in [12] to generate large vocabularies, which is much faster than traditional *k-means* methods.

4.2. Selection of spatial-bag-of-features

We have proposed a series of spatial-bag-of features with different parameters. However, not all of them are useful for a given dataset. We adopt the RankBoost algorithm [3] to select the most effective configurations. We assume that there is a training set, i.e. query images which have some labeled search results[4]. Given a query image, we can order images in the training set according to their relevance to the query. In this way, we can construct many ordered image pairs. The objective function of RankBoost in each iteration is to select the best weak ranking function which minimizes the number of disordered pairs.

In the learning framework, each feature is regarded as a weak ranker, and *cosine* is adopted to calculate the ranking scores. For linear projection, we enumerate 10 equidistant angles in $[0°, 180°]$. For circular projection, we try all regular grid points ($5 \times 5 = 25$) in the plane as centers. With respect to the histogram resolution L, we set 4 levels (3, 7, 15 and 31). Totally, there are $(10 + 25) \times 4 = 140$ original feature histograms. Since we add three extra variances for translation, rotation and scaling, we can get a family of $140 \times 3 = 420$ spatial-bag-of-features (the translation and the rotation only contribute one to the last multiplier, the other two represent the original form and the scaling). Since each long histogram is decomposed to be single bins, as introduced in Section 3.4, instead of 140 long histograms, there are totally $(10 + 25) \times 3 \times (3 + 7 + 15 + 31) = 5880$ sub-histograms to be further selected.

By running the iteration for N times, we obtain a set of best spatial-bag-of-words, $\mathcal{S}$. Each one in $\mathcal{S}$ defines a similarity function and a corresponding weight α. Let $\mathcal{H}_\mathcal{P}$ and $\mathcal{H}_\mathcal{Q}$ be the final representations of images P and Q, the visual similarity between P and Q is given by Equation 5.

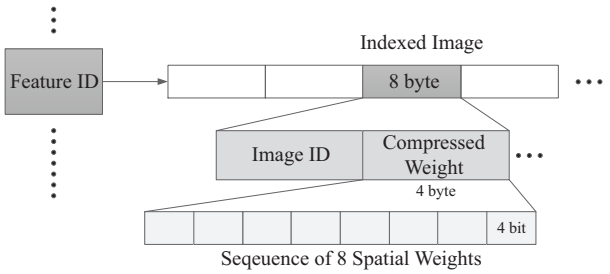

Figure 8. Compressed inverted file structure.

4.3. Indexing structure

Inverted file is a well-studied technique to index high-dimensional sparse feature vectors [12, 13]. Unlike in previous systems, in which an image is often represented by a single histogram and some extra features used for reranking (e.g. spatial information of local features) [10, 12, 14], in our system, an image is represented by a set of selected sub-histogram, while no extra features are needed in ranking process. Therefore, all spatial-bag-of-features, *i.e.* histograms, can be compressed in a single inverted file. We design a data structure for each node in an inverted list as shown in Fig. 8. It takes 4 bits to save the weight of an appeared word of an image under one feature configuration. We uniformly quantize the real value of a histogram entry to be 16 level[5]. In our experiments on large dataset, we select 8 different spatial-bag-of-features. Thus, the extra memory cost is 4 bytes per feature, and the size of the new compressed inverted file is exactly the same with classical inverted file for bag-of-features. With a 8GB memory computer, we can keep all inverted files of 1 million images in memory. A search can be finished within 0.1 second in our experiments.

5. Experiments

A series of experiments were conducted to evaluate the proposed spatial bag-of-features (SBOF). First, several variations of SBOF were evaluated, Second, SBOF was compared with standard bag-of-features (BOF) and BOF with RANSAC reranking (BOF+RANSAC) [12] on Oxford5K dataset. Third, a large scale dataset, i.e. Panoramio1M was leveraged to test the effectiveness and scalability of SBOF, followed by some analysis and visualization of the learnt

[4]In case there is no manually labeled ground truth, we proposed an empirical approach in our experiments.

[5]Actually, previous research demonstrate that binary weights are good enough to represent words in case the vocabulary is very large [5].

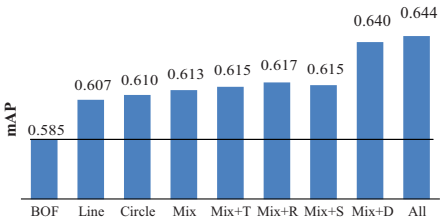

Figure 9. Performance comparison of different combinations of SBOF. The labels on x-axis show the combination type. Mix means both linear and circular projections are used. T, R, and S are abbreviations of translation, rotation, and scaling, which indicate corresponding type of spatial-bag-of-features. D means that long-histogram decomposition is used. Except "Mix+D" and "All", all other combinations did not use long-histogram decomposition.

spatial bag-of-features. Finally, a transfer learning experiment was conducted to show the performance of SBOF without specific training data.

5.1. Datasets and evaluation measure

Oxford5K was first introduced in [12] and have become an evaluation benchmark. It contains 11 different Oxford landmarks and other distractors, totally 5062 high resolution images retrieved from Flickr.

Panoramio1M is provided as distractors. It contains 1 million medium resolution images crawled from the most popular tags in Panoramio. We mixed it with Oxford5K to stress test the performance of the proposed SBOF and retrieval framework on large scale collections.

Paris was first introduced in [11] as a similar dataset to Oxford5K to train an independent visual vocabulary. We adopted it as a training set in RankBoost step to simulate the real-life situation that both the query and the target images are unknown to the retrieval system beforehand.

As in [12], the performance of all experiments is evaluated by the mean average precision (mAP). See [12] for details of mAP.

5.2. Comparison of different SBOF

As aforementioned, several families of SBOF have been proposed with different parameters, and each long-histogram SBOF is decomposed into independent sub-histogram SBOF (or say bins) to avoid severe conflict from different SBOF and improve descriptive power. We first try to use single bin without supervised learning as the feature to retrieve images. The experimental results have shown that for each category in Oxford5K, there will be certain single bin that performs better than BOF, with 1.1% to 223.8% (34.8% on average) improvements for different categories, details of which is omitted here due to space limitation. This result shows that there exists some descriptive SBOF for each category. However, for the whole Oxford5K dataset, single bin did not bring much improvements, i.e. only 1% improvement using the best bin. This observation motivates us to combine descriptive bins together in a supervised manner to capture the spatial properties of different categories for general retrieval tasks.

		50K	100K	500K	1M
Overall	SPM	0.423	0.457	0.534	0.577
	BOF	0.473	0.534	0.603	0.617
	BOF + RANSAC	0.569	0.595	0.643	0.645
	SBOF	0.523	0.571	0.644	0.651
	SBOF+RANSAC	0.575	0.608	0.651	0.655
Testing	SPM	0.432	0.459	0.543	0.586
	BOF	0.476	0.541	0.615	0.633
	BOF + RANSAC	0.576	0.605	0.643	0.646
	SBOF	0.515	0.570	0.644	0.655
	SBOF+RANSAC	0.582	0.623	0.659	0.661

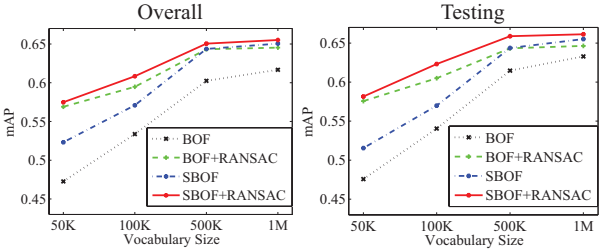

Figure 10. Performance comparison of different algorithms under different vocabulary sizes.

The first two queries of each category in Oxford5K are used as the training query set, and the other queries are used as the testing set. The following combinations are compared in the training set: 1) original BOF without spatial information (BOF), 2) linear ordered BOF without long-histogram decomposition (Linear), 3) Circular ordered BOF without long-histogram decomposition (Circular), 4) Linear+Circular (Mix), 5,6,7) "Mix" method with translation (Mix+T), rotation (Mix+R), scaling (Mix+S) invariance respectively, 8) "Mix" method with long-histogram decomposition (Mix+D), and 9) all spatial bag-of-features families with long-histogram decomposition (All). As shown in Fig. 9, by introducing ordered-bag-of-features to capture geometric information of locality or rotation, the precision has a great improvement over the orderless BOF. Translation, rotation and scaling invariances encoded by SBOF can also enhance the retrieval quality. Another large improvement is caused by long-histogram decomposition to avoid conflicts and heavy clusters. At last, a selection from the complete SBOF family achieves the best precision, which will be used in the following sections. It should be noticed that in spite of using the complete SBOF family, only 10 bins are selected by RankBoost, the dimension of the final feature representation of which was the same as or less than other combinations.

5.3. Comparison with other methods

We compared the proposed SBOF with other methods on Oxford5K under different vocabulary sizes (50K, 100K, 500K and 1M). In order to make our baseline comparable to the results in [12], we adopted their source SIFT descriptors and 1M vocabulary. We also chosen the same AKM method to train other three vocabularies and implemented the same RANSAC algorithm for reranking. The baseline curves of BOF and BOF+RANSAC implemented by us are quite close to the reported results. For each vocabulary, we used Rankboost to select a combination of 10 bins on the same training set in Section 5.2. We report the performance on both all queries and the unseen queries in Fig. 10.

Five methods are compared, which are traditional bag-of-features (BOF), two BOF models with spatial information, i.e. BOF+RANSAC and spatial pyramid matching (SPM), the proposed spatial-bag-of-features (SBOF), and SBOF with RANSAC reranking (SBOF+RANSAC). Several conclusions could be drawn from Fig. 10. First, in spite of encoding spatial information, SPM is much worse than BOF. This result is reasonable since SPM was particularly designed for natual scene categorization, and the horizontal and vertical divisions in spatial space are improper at all for general image retrieval problem. Second, the proposed SBOF outperforms traditional BOF a lot, which shows the effectiveness of the encoding of spatial information in SBOF. Third, compared with BOF+RANSAC, in spite of the lower performances using 50K and 100K vocabularies, with large vocabularies such as 500K and 1M, SBOF has an equal or superior performance. The reason is that, in a smaller vocabulary, quantified features are less discriminative, which affects the precision of capturing correct spatial configurations. Combined with the strength of retrieval efficiency brought from bag-of-feature representation, SBOF outperforms of BOF+RANSAC on 500K or larger vocabularies. Moreover, the performance of BOF+RANSAC depends a lot on the precision of the top results of BOF, which causes the limitation of BOF+RANSAC where the performance of BOF decreases a lot, e.g. in large scale image retrieval (see Section 5.4 for detailed experiments). Furthermore, since SBOF encodes spatial information into the bag-of-features-like representation itself, it is orthogonal to any reranking methods. Therefore, we also tested our method combined with RANSAC reranking, which achieved the best performance on all vocabulary sets. We find that the improvement of SBOF+RANSAC over SBOF decreases as the vocabulary grows, which indicates that on large vocabulary such as 1M, the spatial information encoded by RANSAC reranking has almost been captured by SBOF.

5.4. Comparison on a large scale dataset

Since the proposed SBOF is designed for supporting large scale image retrieval, we tested its performance on Oxford5K + Panoramio1M. The 1M images are added as pure outliers. After feature detection and quantization, this dataset contains totally 1,183,640,886 features, 1184 per image on average. We used an Intel 4×2.4G GHz Quad machine with 32GB memory to conduct our test. For BOF method, we adopted the classical inverted-file index data structure. For SBOF, we used a compressed one mentioned in Section 4.3 to store top 8 selected representations from Oxford5K. Thus the size of the index file is 6,555,743,616 bytes = 6.1GB. The average response time of BOF is 0.042 second, while the time of our method is 0.056 second per query. The results have shown that both the time complexity and the memory cost of the proposed spatial-bag-of-features are comparable to traditional bag-of-features. The performances compared with BOF and BOF+RANSAC are provided in Fig. 11. Notice that although the performances of BOF and BOF+RANSAC are drawn in the figure, they are actually not related with the horizontal axis which represents the number of selected SBOF. From the results we can see that, SBOF not only significantly outperforms BOF, but also significantly outperforms BOF+RANSAC with less than 1000 candidate images for reranking. Although with more than 100,000 candidates, the performance of BOF+RANSAC approaches to SBOF, its time cost is more than 146 times comparing to SBOF. The inferior performance of BOF+RANSAC is caused by the precision drop of BOF in large scale dataset, which cannot provide enough positive images in top results to RANSAC. Actually in real applications, such kind of spatial verifications can only apply on few top images for time concern. However, our method indexes the spatial information of the whole dataset, and thus the precision is higher.

	mAP	Time
BOF	0.408	0.04
SBOF	0.550	0.06
BOF+RANSAC		
100	0.466	0.20
200	0.484	0.34
400	0.494	0.58
800	0.516	0.99
1500	0.525	1.59
100000	0.543	8.16

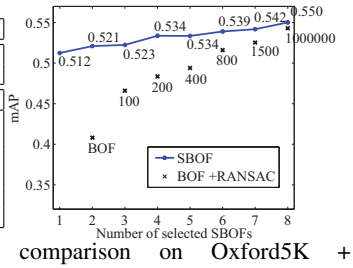

Figure 11. Performance comparison on Oxford5K + Panoramio1M. In the table we provide performances of BOF+RANSAC under our results, where the numbers in the left collum represent the number of used top candidates for reranking.

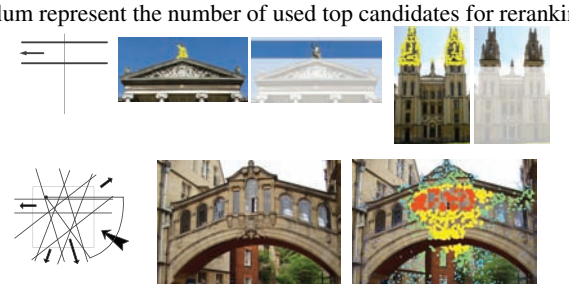

Figure 12. SBOF Visualization. On the first row, we list one single non-invariant SBOF and its examples. Points in the images show features preserved by this SBOF, which indicate the locality information of some characters (a unique sculpture or tower on the top) is captured. On the second row, 5 non-invariant SBOF learned from Oxford5K are applied on a query image. As we can see, this combination could roughly sketch the key structures of this bridge.

5.5. Spatial configuration visualization

An interesting fact observed during experiments is that, Rankboost usually first selects several invariant representations (i.e. ordered BOFs with translation, rotation, or scaling transformations) to handle the most general cases in the dataset. Then at the rest stage of iterations, non-invariant representations (i.e. ordered BOFs without spatial transformations) are chosen to capture specific configurations in this dataset. We reconstruct non-invariant representations learned from 1M vocabulary. Since these spatial information is fixed on each images, it could be regarded as a prototype generated from Oxford5K, which summarizes the common spatial configurations among all the positive examples. We apply it on query images and find that it sketch some essential structures (see Fig. 12 for details).

	Paris	Oxfod
1	0.619	0.613
2	0.619	0.621
3	0.621	0.620
4	0.626	0.626
5	0.632	0.634
6	0.635	0.636
7	0.636	0.641
8	0.631	0.646
9	0.632	0.648
10	0.632	0.651

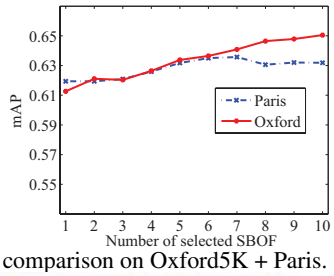

Figure 13. Performance comparison on Oxford5K + Paris.

Figure 14. Illustrations of top retrieved images. For each query, the first line is our result, and the other is obtained from BOF. Yellow line makrs a query object and red line means a false positive.

5.6. Transfer learning

In real situations, we may face the case that there is not a specific training set exactly as the same as users' queries. In this case, we can add some existing training images roughly related with users' queries to the image dataset. Thus, we designed a transfer learning experiment to test the applicability of our representations. In detail, we use Paris dataset to guide the selection of SBOF and test the selected SBOF on the overall queries of Oxford5K. Five major categories of Paris dataset are used and in each category the top 30 images whose content is apparently consistent with its tag are selected. For each category, 5 images are used as queries, and other ones are thrown into Oxford5K as target images. 10 most powerful SBOF are learnt and the queries in Oxford5K are tested using these SBOF.

We compared the transferred result with the SBOF learned from Oxford5K set, as shown in Fig. 13. We can see that, the top several learnt SBOF using Paris dataset also achieve good performance for Oxford5K queries. However, more than several selections, the performance slightly drops. It is reasonable and also consistent with the observation in Section 5.5 that, at first both of the two RankBoost trainers select the general knowledge for retrieving buildings. Then after encoding enough common spatial information of buildings, the trainers then select features to fit for the specific spatial information of each dataset.

6. Conclusions and future work

We have demonstrated a novel technique which is able to adapt the orderless bag-of-features to a so-called spatial-bag-of-features. The new feature has two major merits: 1) it has the same format as the traditional bag-of-features, and 2) it can effectively encode spatial information. Owning to these merits, an effective and efficient index solution were designed, in which all information utilized in ranking is packed in a single inverted index.

In the experiments, the proposed approach obtained comparable accuracies on benchmark dataset as state-of-the-art approaches, and achieved significant improvement both in precision and search time in large scale applications. In the future, we are interested in developing a more principled framework of our feature family and applying SBOF in recognition tasks.

Acknowledgements

The work of the 1st and 4th authors was supported in part by the National Basic Research Program of China (Grant No. 2005CB724301), the Science and Technology Commission of Shanghai (Grant No. 08511501701), and NSFC, China (Grant No. 60775007).

References

[1] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Comm. ACM, 2008*, 2008.

[2] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *ICCV*, 2007.

[3] Y. Freund, R. Iyer, R. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *JMLR'03*.

[4] K. Grauman and T. Darrell. The pyramid match kernel: discriminative classification with sets of image features. In *ICCV*, 2005.

[5] H. Jégou, M. Douze, and C. Schmid. Packing bag-of-features. In *ICCV*, 2009.

[6] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[7] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

[8] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *IJCV*, 2004.

[9] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006.

[10] M. Perdoch, O. Chum, and J. Matas. Efficient representation of local geometry for large scale object retrieval. In *CVPR'09*.

[11] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR 2008*.

[12] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR 2007*.

[13] J. Sivic and A. Zisserman. Video google: a text retrieval approach to object matching in videos. In *ICCV'03*.

[14] Z. Wu, Q. Ke, M. Isard, and et al. Bundling features for large scale partial-duplicate web image search. In *CVPR'09*.

[15] S. Zhang, Q. Tian, and et al. Descriptive visual words and visual phrases for image applications. In *ACM MM'09*.

[16] X. Zhang, Z. Li, L. Zhang, W.-Y. Ma, and et al. Efficient indexing for large scale visual search. In *ICCV'09*.