Building Contextual Visual Vocabulary for Large-scale Image Applications

Shiliang Zhang[1], Qingming Huang[2], Gang Hua[3], Shuqiang Jiang[1], Wen Gao[4], Qi Tian[5]

[1]Key Lab of Intelli. Info. Process., Inst. of Comput. Tech., CAS, Beijing 100190, China

[2]Graduate University of Chinese Academy of Sciences, Beijing 100049, China

[3]IBM Research T. J. Watson Center, NY, 10523, U.S.A.

[4]Peking University, No. 5, Yiheyuan Road, Beijing, 100871, China

[5]Computer Science Depart., University of Texas at San Antonio, TX 78249, U.S.A.

{slzhang, qmhuang, sqjiang, wgao}@jdl.ac.cn, ganghua@gmail.com, qitian@cs.utsa.edu

ABSTRACT

Not withstanding its great success and wide adoption in Bag-of-visual Words representation, visual vocabulary created from single image local features is often shown to be ineffective largely due to three reasons. First, many detected local features are not stable enough, resulting in many noisy and non-descriptive visual words in images. Second, single visual word discards the rich spatial contextual information among the local features, which has been proven to be valuable for visual matching. Third, the distance metric commonly used for generating visual vocabulary does not take the semantic context into consideration, which renders them to be prone to noise. To address these three confrontations, we propose an effective visual vocabulary generation framework containing three novel contributions: 1) we propose an effective unsupervised local feature refinement strategy; 2) we consider local features in groups to model their spatial contexts; 3) we further learn a discriminant distance metric between local feature groups, which we call *discriminant group distance*. This group distance is further leveraged to induce visual vocabulary from groups of local features. We name it *contextual visual vocabulary*, which captures both the spatial and semantic contexts. We evaluate the proposed local feature refinement strategy and the contextual visual vocabulary in two large-scale image applications: large-scale near-duplicate image retrieval on a dataset containing 1.5 million images and image search re-ranking tasks. Our experimental results show that the contextual visual vocabulary shows significant improvement over the classic visual vocabulary. Moreover, it outperforms the state-of-the-art Bundled Feature in the terms of retrieval precision, memory consumption and efficiency.

Categories and Subject Descriptors

I.2.10 [Vision and Scene Understanding]: VISIONS

General Terms

Algorithms, Experimentation, Performance

Keywords

Bag-of-visual Words, Near-duplicate Image Retrieval, Image Search Re-ranking

MM'10, October 25-29, 2010, Firenze, Italy.

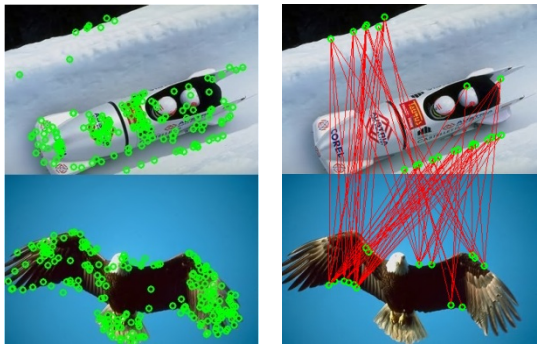
1. INTRODUCTION

Due to the fast development of Internet applications and the popularity of digital sets such as digital cameras, digital video recorders, mobile phones, *etc.*, the amount of multimedia data available in Internet has been explosively increasing. For example, in video and photo sharing websites such as YouTube, Flicker, *etc.*, there are billions of images and millions of hours of digital videos. Moreover, these numbers keep increasing everyday. Thus, the multimedia research community is facing challenging problems including scalable machine learning, feature extraction, indexing and efficient multimedia information retrieval.

The traditional textual information retrieval is successful in processing the large-scale textual information. For example, the Google and Bing search engines could answer users' textual queries responsibly and accurately from billions of web-pages. In the textual information retrieval, the text words, which are compact and descriptive, are used as the basic features for documents. Inspired by the success of text words, researchers are trying to identify basic visual elements from images, namely the visual words and visual vocabulary [1, 2], which could function just like the text words. With descriptive and compact visual words, lots of popular algorithms for textual information retrieval can be leveraged for computer vision and multimedia tasks, such as visual search or recognition. Moreover, the problems caused by large-scale multimedia data might be successfully conquered.

Traditionally, visual words are created by clustering a large number of local features such as SIFT [3] in unsupervised ways. After clustering, each cluster center is taken as a visual word, and a corresponding visual vocabulary is generated. With the visual vocabulary, image can be transformed as Bag-of-visual Words (BoW) representation [1, 2], which is similar as the Bag-of-Words representation in information retrieval. This is simply finished by extracting image local features and quantizing them with their nearest visual words. Attribute to its scalability and simplicity, BoW representation has been very popular in computer vision and visual content analysis in recent years. It has shown promising results for a wide variety of applications such as object recognition [4-19], image and video annotation [20, 21], video event recognition [22-24], *etc*. In addition, combining visual vocabulary and the framework of traditional information retrieval *i.e.*, the inverted file structured indexing and TF-IDF (Term Frequency Inverted Document Frequency) weighting [1, 2], has been illustrated as one of the most promising solutions for the large-scale image and video retrieval [1, 2, 25-29].

Not withstanding its demonstrated success, visual vocabulary is often proven not as effective as the textual words [1, 11, 20, 25,



(a) (b)

Figure 1. The traditional visual word is not descriptive enough.

26]. For example, Figure 1(a) shows the visual words extracted from two images. The identical visual words between them are connected with red lines. It is clear that, although these two images contain different visual objects, lots of visual words are still matched between them. Therefore, it can be inferred that the traditional BoW representation [2] is noisy and non-descriptive. The ineffectiveness of the traditional visual vocabulary might be largely due to its three innate shortcomings, which will be detailed in the following.

Figure 2. The commonly used Difference-of-Gaussian detector detects some noisy and unstable local features in image.

First, many image local features detected by the commonly used detector *i.e.*, the Difference-of-Gaussian (DoG) [3] are not informative and stable enough. In the toy example shown in Figure 2, lots of local features are detected in the cluttered background. Additionally, resizing and stretching the image, cause obvious variation to the number of the detected local features. This implies that some unstable local features may not survive the simple affine transformations. It can be inferred that, under the framework of classic visual word generation, these defects will result in many noisy visual words. Moreover, in near-duplicate image retrieval [1, 2, 26], where the images are commonly edited by resizing, cutting, *etc.*, the performance of classic visual word, which is generated based on the detected image local features will be degraded.

Second, generated from single image local descriptors, classic visual vocabulary is not able to capture the rich spatial contextual information among the local features. However, several previous works have verified that modeling these visual contexts could greatly improve the performance of many visual matching and recognition algorithms [1, 7-10, 19, 25, 26]. This is also the reason that a post geometric verification step [1, 26] is needed to improve the accuracy for vision tasks. An example illustrating this defect is shown in Figure 3.

Previous approaches [1, 7-10, 19, 25, 26] to this spatial context modeling problem predominantly try to identify the combination

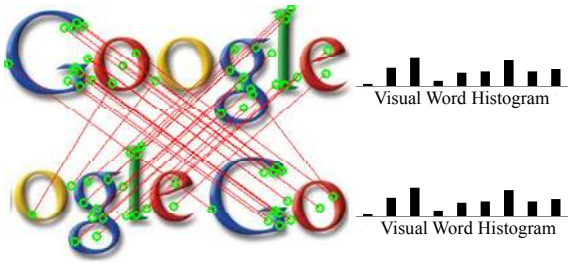


Figure 3. The two images show different semantics. However, their visual word histograms are identical. Traditional BoW representation loses the spatial context in images.

of visual words with statistically stable spatial configurations. This may be achieved, for example, by using feature pursuit algorithms such as AdaBoosting [30], as demonstrated by Liu *et al*. [9]. Visual word correlogram and correlation [10], which are leveraged from the color correlogram [10], are utilized to model the spatial relationships between visual words for object recognition in [10]; In recent work [26], visual words are bundled and the corresponding image indexing and visual word matching algorithms are proposed for large-scale near-duplicate image retrieval. Proposed as descriptive visual word pairs in [7, 25], Visual Phrase captures the spatial information between two visual words and presents better discriminative ability than the traditional visual vocabulary in object categorization tasks. Generally, considering visual words in groups rather than single visual word could effectively capture the spatial configuration among them. However, the quantization error introduced during visual vocabulary generation may degrade the matching accuracy of visual word combination. As illustrated in Figure 4, after quantization, local features that should be matched in the descriptor space may fail to match, and this error may be magnified with general visual word combinations [7-9, 25, 26].

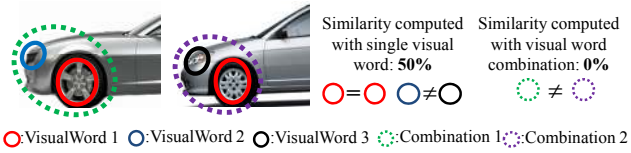


Figure 4. The quantization errors are magnified when combining several visual words together, resulting in the ineffectiveness of the generated combinations. (Refer to the color pdf for a better review)

Third, to generate the visual vocabulary from single local image descriptors, most previous methods employ a general distance metric, such as Euclidean distance or L1-norm, to cluster or quantize the local features. This is unsatisfactory since it largely neglects the semantic contexts of the local features. With a general distance metric, local visual features with similar semantic relationship may be far away from each other, while the features with different semantics may be close to each other. For instance, as illustrated in Figure 5, with unsupervised clustering, the local features with similar semantics can be clustered into different visual words, while the local features with different semantics can be assigned into the same visual words. This defection results in some incompact and non-descriptive visual words, which are also closely related with the mismatches occurred between images. For instance in Figure 5, the non-descriptive visual words (*i.e.*, the black points) are matched between two images with different semantics, *i.e.*, bicycle and motorbike. There have been some previous works attempting to address this phenomenon by posing

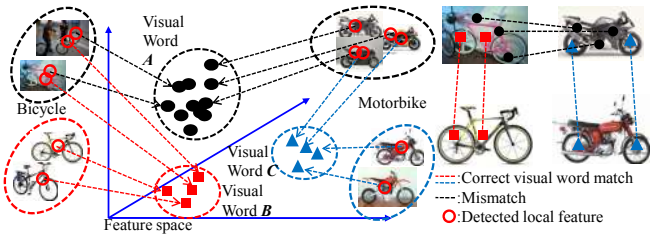


Figure 5. The semantic contextual information is lost in unsupervised visual vocabulary generation, resulting in lots of noisy visual words and mismatched visual words.

supervised distance metric learning [7, 8, 12, 20]. In [12], the classic visual vocabulary is used as the basis, and a semantically reasonable distance metric is learned to generate more effective high-level visual vocabulary. However, the generated visual vocabulary is small-scale problem oriented. In a recent work [20], the authors capture the semantic context in each object category by learning a set of reasonable distance metrics between local features. Then, semantic-preserving visual vocabularies are generated for different object categories. Experiments on large-scale image database demonstrate the effectiveness of the proposed algorithm in image annotation. However, the codebooks in [12] are created for individual object categories, thus they are not universal and general enough, which limits their applications. Generally, although promising progress has been made, most of those methods are small-scale problems oriented [7, 8, 12], or do not take the spatial contexts into consideration [12, 20].

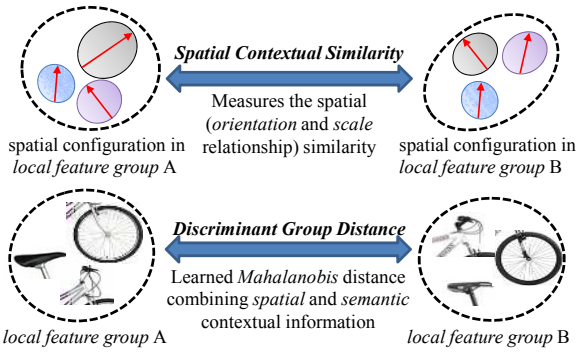


Figure 6. The discriminant group distance measures the spatial context weighted *Mahalanobis* distance between two local feature groups.

We address these three challenges discussed in a unified framework by casting the problem as refining local feature and learning a *discriminant group distance* between *local feature groups*. In contrast to previous methods, we propose an unsupervised local feature refinement strategy to filter the unstable local features in images. Further, we take groups of local features into consideration instead of treating the local features independently. In this way, the spatial contextual information between local features can be modeled and the magnified quantization error of directly combining visual words can be depressed. Based on the spatial configuration of the local features within the feature group, we define the *spatial contextual similarity* between two local feature groups (see Figure 6).

Inspired by the metric learning framework of Globerson and Roweis [31], we propose to learn a spatial context weighted *Mahalanobis* distance metric, namely the *discriminant group distance* between local feature groups, by collapsing groups of local features with same semantic labels (see Figure 6). Due to the weight introduced from the spatial contextual similarity, the metric learning will put more efforts on those local feature groups with same semantic label but small spatial contextual similarities. This is in contrast to the original formulation of [31], where all training examples are treated equally.

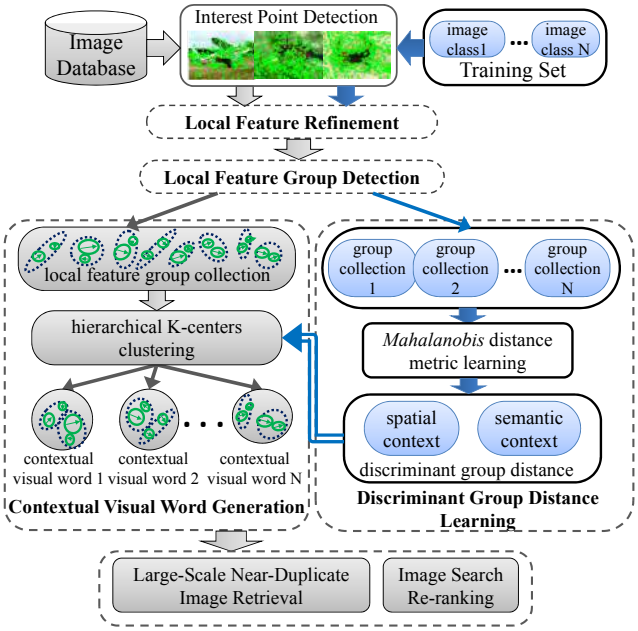


Figure 7. The proposed framework

The learned group distance is further applied to create visual vocabulary form local feature groups, namely the *contextual visual vocabulary*, which incorporates both spatial and semantic level contextual information. This process is illustrated in Figure 7. We hence develop a more descriptive Bag-of-Contextual-visual-Words (BoCW) representation for large-scale image applications. Its superiority, when compared with previous approaches, is demonstrated in two applications: large-scale near-duplicate image retrieval and image search re-ranking. The contribution of our work can be summarized as: 1) we propose an effective local feature refinement strategy to obtain stable local features; 2) we consider local feature groups instead of single local features and define a spatial context weighted discriminant group distance; 3) finally, we combine the spatial and semantic contexts in forming the novel contextual visual vocabulary.

The remainder of this paper is organized as follows. Section 2 illustrates our proposed local feature refinement strategy and local feature group detector. Section 3 formulates the learning problem for the discriminant group distance and presents the details of inducing the contextual visual vocabulary. Section 4 presents and discusses our experimental results in two image applications, followed by the conclusions and future work in Section 5.

2. LOCAL FEATURE GROUP DETECTION

To extract the local feature groups, we firstly use the DoG detector [3] to detect the interest points and extract local features in images. According to [3], from each interest point, three kinds of information can be extracted, *i.e.*, the scale information S, the local feature descriptor D (*i.e.*, the SIFT [3]), and the orientation

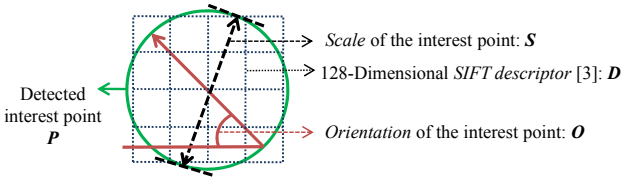


Figure 8. Extracted information from local feature *P*(*S*, *O*, *D*)

information *O*. As illustrated in Figure 8, each local feature is a triple *(S, O, D)*, where *S* and *O* stand for the scale and orientation of the interest point, and *D* represents the 128-dimensional SIFT descriptor [3], which describes the appearance information of the local feature.

2.1 Local Feature Refinement

In the local feature refinement, we intend to extract the most stable and informative local features, and filter the noisy and unstable ones as much as we can. Although SIFT is designed to be invariant to scale, rotation, and small viewpoint changes [3], as shown in Figure 2, after the transformation such as stretching, resizing, *etc.*, the number of extracted local features is decreased. It is reasonable to infer that, the unstable local features which are sensitive to these transformations may not survive. Therefore, in order to find stable local features in an image that are resistant to affine transformation such as resizing, rotation, *etc.*, we first generate new images by performing these transformations to the original image, then extract local features from these images, and finally find the repeated local features across these images.

Suppose the coordinate of a pixel in the original image is [x, y], then the image transformation can be denoted as:

$$\begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} a_5 \\ a_6 \end{bmatrix} \quad (1)$$

where $[\hat{x}, \hat{y}]$ denotes the coordinate of the pixel in the new image after transformation. In Eq. 1, the affine transformation of the original image is controlled by six parameters, *i.e.*, a_1, a_2, a_3, a_4, a_5, and a_6. It should be noted that introducing more complicated transformations means more strict requirements for the stability. However, more affine transformations also slow down the local feature extraction. In this paper, we only use two transformations: horizontal stretching (a_1=0.8, a_2=0, a_3=0, a_4=1, a_5=0, a_6=0), and vertical stretching (a_1=1, a_2=0, a_3=0, a_4=0.8, a_5=0, a_6=0). As illustrated in Figure 9 (a), we first extract local features on the original image and the new images. Then, for each detected local feature in the new image, we compute their corresponding coordinates in the original image with Eq. 2, *i.e.*,

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix}^{-1} \left(\begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} - \begin{bmatrix} a_5 \\ a_6 \end{bmatrix} \right) \quad (2)$$

Therefore, for each transformed new image, we map all its extracted local features back to the original image to get a feature map, *e.g.* the ones in the Figure 9 (b). Finally, we detect the repeated local features across different feature maps. For each local feature *A* in the original image, we first find its most similar local feature *B* from a small neighbor region in each feature map. The similarity is computed with Eq. 3 as:

$$sim(A, B) = A \bullet B / \left(\|A\| \times \|B\| \right) \quad (3)$$

where, *A* and *B* are two 128-dimensional SIFT descriptors. If the similarity between this local feature *A* and the corresponding local

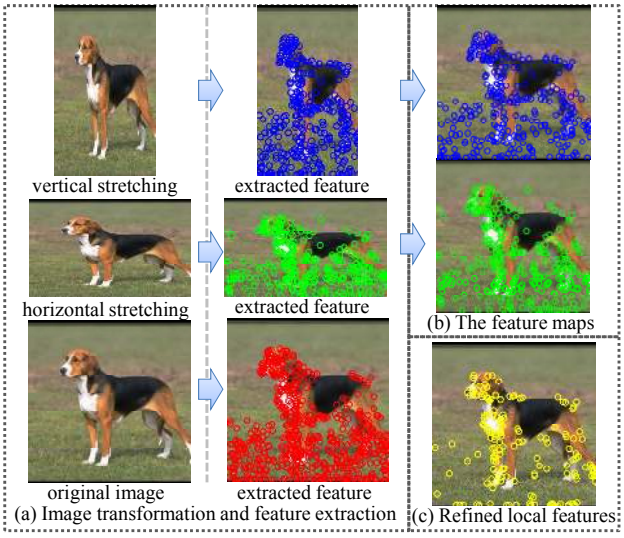


Figure 9. An example of the local feature refinement

feature *B* in each feature map is larger than a threshold, this local feature *A* will be kept as a stable feature. The threshold is experimentally set as 0.8 in our experiments. An example of the refined local feature is illustrated in Figure 9 (c).

This local feature filtering strategy presents two advantages: 1) it is unsupervised, only based on its own information of the image and needs no training data; 2) it effectively decreases the number of local features in each image, hence improves the efficiency and memory consumption of the image applications. We will further test this algorithm in Section 5.

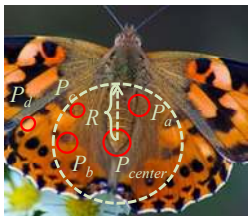
2.2 Local Feature Group Extraction

As shown in Figure 6, each local feature group contains several local features. We denote the local feature as *P*(*S, O, D*), and the local feature group as: $G\{P^{(1)}, P^{(2)}, \ldots, P^{(n)}\}$, where *n* is the number of local features contained in a group. In our formulation, the discriminant group distance is defined between two groups containing the same number of local features to simplify the computation.

Different algorithms can be utilized to detect the local feature groups. To make a tradeoff between efficiency and effectiveness, we define the local feature group as the co-occurred local features within a certain spatial distance threshold. In general, the following factors should be properly considered to generate local feature groups: 1) The local feature group should be scale invariant; 2) the local feature group should be repeatable; 3) the number of local features contained in each group should be small; and 4) the extraction should be efficient, both in terms of computational and memory consumption.

In order to satisfy the first requirement, we use the scale information [3] of local feature as the basis to compute the spatial distance related to the co-occurrence between local features. As for the second and third requirements, according to Liu, *et al.* [9], if too many local features or visual words are combined (*i.e.*, combinations with higher orders), the repeatability of the combination will decrease. In addition, if more local features are contained in each group, there would be more possible feature-to-feature matches between two groups. This may make the computation of the corresponding spatial contextual similarities to be time consuming. We shall detail this in Section 3. Therefore, to

meet the second, third and forth requirements, we fix the maximum number of local features in each local feature group as 3. But it should be noted that our proposed framework does not hinder to use larger number of local features in a feature group.



○: local feature in Image

Detected local feature groups:
(P_{center}, P_a), (P_{center}, P_b) (P_{center}, P_c)
and (P_{center}, P_a , P_b)

Figure 10. The utilized local feature group detector

To detect local feature groups containing two local features, we use the detector illustrated in Figure 10. In the figure, a circle with radius R is centered at a local feature. A local feature group is formed by the centered local feature and another local feature within the circle. The radius R is computed with

$$R = S_{center} \bullet \lambda \tag{4}$$

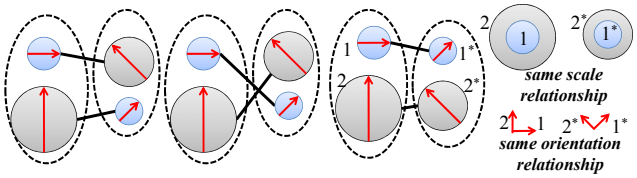
to achieve scale-invariance, where S_{center} is the scale of the centered local feature, λ is a parameter that controls the spatial span of the local feature group, which in turn affects the co-occurrence relation between local features. A larger λ is necessary for identifying stable spatial relations and overcoming the issue of potential sparseness of the local feature groups. However, a large λ also increases the computational cost and is more prone to noise. We experimentally set λ as 6, which shows a good tradeoff between efficiency and performance.

The detector illustrated in Figure 10 is also utilized to detect the local feature groups containing three local features. A local feature group containing three local features is generated by the centered local feature and its two nearest local features within the circle. For example in the Figure 10, a local feature group (P_{center}, P_a, P_b) is generated. Similarly, the radius R is computed with Eq. 4 to achieve scale-invariance and the parameter λ is set as 6.

By scanning each local feature with the detector, two collections of local feature groups containing two local feature and three local features can be generated, respectively. Because of the limited local feature number in each image and the properly selected λ, this operation is efficient.

3. FORMULATION OF DISCRIMINANT GROUP DISTANCE

The discriminant group distance is computed as the spatial context weighted *Mahalanobis* distance based on the local features in two groups. Note that the discriminant group distance is defined between groups containing identical number of local features. With n local features in each group, there are $n!$ possible feature-to-feature matches between two groups. We call each possible match as a match order r. As illustrated in Figure 11(a), there are 2 match orders when n=2. It is reasonable to seek the best match order for the group distance computation. The best match order is defined as the one that maximizes the spatial similarity, *i.e.*, the one in Figure 11(b). Consequently, to define the group distance, we first compute the best match order between two groups, and hence obtain the corresponding spatial contextual similarity based on the spatial relationship of the two feature groups. We shall present more details about this in Section 3.1. After that, we further learn a weighted *Mahalanobis* distance metric to model the distance between two groups, which will be introduced in Section 3.2 and Section 3.3, respectively.



(a): The two match orders if each local feature group contains 2 local features

(b): The best match order based on the spatial similarity

Figure 11. The illustration of match orders and the best match order based on the spatial contextual similarity

3.1 Spatial Contextual Similarity

We define the spatial context of each local feature group as the orientation and scale relationships between the local features inside the group. Because each local feature contains two aspects of spatial information *i.e.*, the scale and orientation, the spatial contextual similarity between local feature groups is defined as:

$$SimCxt^{(I,J)} = \max_{r}\left(SimS_r^{(I,J)} + SimO_r^{(I,J)}\right)/2 \tag{5}$$

where, $SimCxt^{(I,J)}$ denotes the spatial contextual similarity between local feature group I and J. $SimS_r^{(I,J)}$ and $SimO_r^{(I,J)}$ are the scale and orientation similarity with the match order r, respectively. Recall that each match order denotes a possible feature-to-feature match. The $SimS_r^{(I,J)}$ and $SimO_r^{(I,J)}$ are obtained based on the spatial and orientation relationships SR and OR contained in the two local feature groups, respectively. They are first computed as:

$$\begin{cases} SR_r^{(I)} = \sum_{i=1,j>i}^{n} \log\left(1+S^{(i)}/S^{(j)}\right) \\ SR_r^{(J)} = \sum_{\substack{u=r(i),i=1 \\ v=r(j),j>i}}^{n} \log\left(1+S^{(u)}/S^{(v)}\right) \end{cases} \quad \begin{cases} OR_r^{(I)} = \sum_{i=1,j>i}^{n} \mathrm{N}(O^{(i)}-O^{(j)}) \\ OR_r^{(J)} = \sum_{\substack{u=r(i),i=1 \\ v=r(j),j>i}}^{n} \mathrm{N}(O^{(u)}-O^{(v)}) \end{cases} \tag{6}$$

where $\mathrm{N}(\theta)$ normalizes the angle θ between $[0, \pi]$. n is the number of local features in each group. The superscript (i) and (j) index the local features in group I, and $r(i)$ and $r(j)$ return their matched local features in group J with match order r. Since SR and OR are defined based on the relative ratio and difference, it is clear that they are *scale* and *rotation* invariant, respectively.

Then, with the definition of SR and OR, we compute the $SimS_r^{(I,J)}$ and $SimO_r^{(I,J)}$ with match order r, which take values in [0, 1], *i.e.*,

$$\begin{aligned} SimS_r^{(I,J)} &= \min_{K\in\{I,J\}} SR_r^{(K)} \Big/ \max_{K\in\{I,J\}} SR_r^{(K)} \\ SimO_r^{(I,J)} &= \min_{K\in\{I,J\}} OR_r^{(K)} \Big/ \max_{K\in\{I,J\}} OR_r^{(K)} \end{aligned} \tag{7}$$

After computing the scale and orientation similarity between two local feature groups with all possible match orders, we finally obtain the spatial contextual similarity with Eq. 5. The corresponding best match order is denoted as r^* and $i^*=r^*(i)$ stands for a local feature, which matches the local feature i in the other group, under this best match order r^*. A toy example explaining the best match order is illustrated in Figure 11(b). Intuitively, the local features within two feature groups are matched with the best match order to ensure the maximum spatial contextual similarity. The spatial contextual similarity of each local feature group pair and the corresponding best match order is calculated before computing their discriminant group distance.

3.2 Formulation of the Discriminant Group Distance

Recall that each local feature group contains both the appearance (*i.e.*, the SIFT descriptor) and spatial contextual information. Thus, the defined group distance should satisfy the following two requirements: firstly, it should properly combine the spatial and appearance cues; secondly, it should incorporate the semantic context between local feature groups. We address these two requirements by learning a spatial context weighted *Mahalanobis* distance, which is called discriminant group distance, *i.e.*,

$$\begin{aligned} DGD(G_I, G_J \mid A) &= d_{IJ}^A \\ &= W_{IJ} \sum_{k=1}^{n} (D_I^{(k)} - D_J^{(k^*)})^T A (D_I^{(k)} - D_J^{(k^*)}) \\ \text{where } W_{IJ} &= 1 - SimCxt^{(I,J)} \end{aligned} \tag{8}$$

where W_{IJ} denotes the spatial contextual weight that is derived from the spatial contextual similarity in Section 3.1. $D_J^{(k^*)}$ is the local feature matched with $D_I^{(k)}$ under the best match order computed in Section 3.1. A is the *128×128* matrix to be learned from the semantic labels of the local feature groups.

Intuitively, we try to find a good distance metric which makes the feature groups with similar semantics contexts close to each other and those with different semantics appearing far away. To achieve this, suppose we are given a set of M labeled examples: (G_I, y_I), $I=1,\ldots,M$, where G_I and y_I denote feature group and label, respectively. Following Globerson, *et al.* [31], for each group G_I, we define a conditional distribution for all other groups, *i.e.*,

$$p^A(G_J \mid G_I) = \frac{1}{Z_I} e^{-d_{IJ}^A} = e^{-d_{IJ}^A} \bullet \frac{1}{\sum_{K \neq I} e^{-d_{IK}^A}} \qquad I \neq J \tag{9}$$

Since an ideal distance metric would set the distance between pair of groups with the same labels to be zero, and distance between pair of groups with different labels to be infinity, the ideal conditional distribution should be

$$p_0(G_J \mid G_I) \propto \begin{cases} 1 & y_I = y_J \\ 0 & y_I \neq y_J \end{cases} \tag{10}$$

Therefore, our metric learning should seek a matrix A^* such that $p^{A^*}(G_J \mid G_I)$ is as close as possible to the ideal conditional distribution $p_0(G_J \mid G_I)$. We define the objective function as:

$$f(A) = \sum_{I,J} KL[p_0(G_J \mid G_I) \,\|\, p^A(G_J \mid G_I)] \quad s.t. \quad A \in PSD \tag{11}$$

where *PSD* stands for the set of Positive Semi-Definite matrices. Then, similar to [31], we compute the A^* as: $A^* = \arg\min_A f(A)$.

3.3 Optimization of the Discriminant Group Distance

With the Kullback–Leibler divergence computation, *i.e.*, $KL(P \,\|\, Q) = \sum_i P(i) \log[P(i)/Q(i)]$, Eq. 9, and Eq. 10, we may rewrite Eq. 11 as:

$$\begin{aligned} f(A) &= \sum_{I,J} \left[p_0(G_J \mid G_I) \bullet \log\left[p_0(G_J \mid G_I) / p^A(G_J \mid G_I) \right] \right] \\ &= -\sum_{I,J,y_J = y_I} \log p^A(G_J \mid G_I) = \sum_{I,J,y_J = y_I} d_{IJ}^A + \sum_I \log \sum_{J \neq I} e^{-d_{IJ}^A} \end{aligned} \tag{12}$$

A crucial property of this optimization problem, which is manifested by Globerson and Roweis [31], is that the objective function is convex with respect to A. Since the optimization of A is convex, it would have only a unique minimum point which is globally optimal. Thus, it can be optimized with various convex optimization methods, and the most important consideration is the efficiency of different algorithms. As in [31], we also utilize a simple gradient descent method, specifically the projected gradient approach. At each iteration, we take a small step in the direction of negative gradient of the objective function, followed by a projection back in the *PSD* cone to make sure that A is always a *PSD*. This projection is performed by taking the eigen-decomposition of A and removing the components with negative eigen-values. The gradient of Eq. 12 is given in the following equation, more details about its computation can be found in [31].

$$\nabla f(A) = \sum_{I,J} \left(W_{IJ} \left(P_0(G_J \mid G_I) - P(G_J \mid G_I) \right) \sum_{k=1}^{n} (D_I^{(k)} - D_J^{(k^*)})(D_I^{(k)} - D_J^{(k^*)})^T \right) \tag{13}$$

We summarize the details of the optimization algorithm in Algorithm 1. The learned matrix A will render the final distance metric to incorporate the semantic contexts of each local feature group. Since each SIFT descriptor D is a *128* dimensional vector, the eigen-decomposition operation of the *128×128* matrix A can be finished very efficiently. The most time consuming operation is to compute the first order gradient in Eq. 13, which is of $O(M^2)$ computational complexity.

Algorithm1: compute the matrix A

Input: set of labeled local feature groups: $(G_I, y_I), I = 1,\ldots,M$. The maximum iteration time T. The weighting parameter α

Output: the learned matrix A

Initialization: Initialize A_0, which is a *PSD* matrix.

For iteration $t = 1:T$

Set $A_{t+1} = A_t - \alpha \nabla f(A_t)$ [31] where $\nabla f(A_t)$ is computed with Eq. 13.

Calculate the eigen-values and eigen-vectors of the matrix A_{t+1}

$$A_{t+1} = \sum_k \lambda_k u_k u_k^T,$$

Set $A_{t+1} = \sum_k \max(\lambda_k, 0) u_k u_k^T$

End

3.4 Contextual Visual Vocabulary Generation

The contextual visual vocabulary can be generated through clustering the local feature group collection with the learned discriminant group distance. As a popular clustering algorithm, hierarchical K-means is generally efficient for the visual word generation. Additionally, the generated visual words with hierarchical K-means are organized in a hierarchical vocabulary tree, with which the images could be transformed into BoW representations efficiently. However, in K-means clustering, to compute the cluster centers of a cluster C with the defined distance metric, we have to solve the optimization problem:

$$G^* = \arg\min_G \sum_{G_I \in cluster\ C} DGD(G_I, G \mid A) \tag{14}$$

Solving this Eq. 14 in each iteration of the K-means clustering is time consuming. Thus, we employ the hierarchical K-centers

clustering instead. Different form *K*-means, the cluster center of *K*-centers is simply updated as the data point having the maximum similarities with the other data points in the same cluster. It is computed as

$$G_I^* = \arg\min_{G_I} \sum_{G_I, G_J \in cluster\ C, J \neq I} DGD(G_J, G_I \mid A) \quad (15)$$

According to Eq. 15, we need to store a group-to-group similarity matrix for each cluster to update the cluster center. Intuitively, once the similarity matrix of a cluster is computed, the clustering operation in its corresponding sub-clusters can be finished efficiently. Meanwhile, the clustering can be implemented in a depth-first way to lower the memory cost. The clustering finally produces a hierarchical vocabulary tree, and each cluster center of the leaf node is taken as a contextual visual word. Since the contextual visual words are generated from local feature groups rather than the single local features, each of them preserves rich spatial contextual information. Meanwhile, because the distance utilized for clustering is more discriminative, the contextual visual words have more capacity to represent the image concept than the traditional visual words. In addition, after quantizing local feature groups into contextual visual words, we also keep their spatial cues for verification to remove the mismatches. We compare the proposed contextual visual vocabulary with the state-of-the-art algorithms in different applications in the Section 4.

4. EXPERIMENTS AND EVALUATIONS

4.1 Training Set

In order to optimize the discriminant group distance to make it a generic distance metric, a representative training set is required. Since the group distance measures the distance between local feature groups rather than the entire images, it would be ideal to annotate certain amount of local feature groups to form the training set. However, such annotation task is difficult and expensive to conduct. As a tradeoff, we first select 1500 image categories from the ImageNet [32] dataset, which contains visually consistent single objects. Then, to depress the noise from the cluttered background, we first run the local feature refinement and then extract the local feature groups from the most centered patches of each image. The local feature groups extracted from the same image category are tagged with the same labels. From the selected categories, we manage to get about 1 million local feature groups containing two local features and about 0.5 million groups containing three local features for training, *i.e.*, the discriminant group distance learning. After that, we extract about 5 million local feature groups containing two local features and 2 million groups containing three local features for the *K*-centers clustering. With the extracted local feature groups and the learned group distance, contextual visual vocabulary sets with different sizes are generated by different parameters, *i.e.*, layer number and child number in hierarchical clustering. Our experiments are conducted on a computer with a 4-core 2.8GHz CPU and 4GB memory. The distance metric learning and the contextual visual vocabulary generation are finished within two days.

4.2 Near-duplicate Image Retrieval

The goal of near-duplicate image retrieval is to locate the near- and partial-duplicate images in the image database for a given query. In recent years, it has been successfully utilized in copyright violation detection and large-scale web image retrieval [2, 26, 29]. In this experiment, we test the contextual visual vocabulary on two image datasets. The first one is the Ukbench dataset [2]. The Ukbench dataset contains 2550 image categories, each of which contains 4 near-duplicate images. Examples of the

(a) Two examples of the Ukbench dataset [2]

(b) Two examples of the collected near-duplicated images

Figure 12. Examples of the dataset utilized for image retrieval

Ukbench dataset are illustrated in Figure 12 (a). The other dataset is a large-scale image dataset collected with the similar method of Wu *et al.* [26]. We first randomly download about 1.5 million web-images from the Internet. Then, we manually download 12 image categories including "Abbey Road", "Uncle Sam", "Energy Star", *etc.* as the image set with groundtruth. Each category contains about 30 near-duplicate images. Some examples of the collected near-duplicate images are illustrated in Figure 12 (b).

The first experiment is carried out on the Ukbench dataset. The 10200 images are first transformed into Bag-of-Contextual-visual-Words (BoCW) representation and then, inverted file is adopted for image indexing. The TF-IDF weighting [2] computed with Eq. 16 is utilized for image retrieval, *i.e.*,

$$tfidf_i^{(J)} = \frac{n_i^{(J)}}{\sum_k n_k^{(J)}} \bullet \log \frac{|D|}{|\{d : i \in d\}|} \quad (16)$$

where, $tfidf_i^{(J)}$ denotes the importance of visual word i to image J, $n_i^{(J)}$ is the time of occurrence of visual word i in image J. $|D|$ denotes the number of images in the database and $|\{d : i \in d\}|$ is the number of images containing the visual word i. All the 10200 images are used as queries. For each query, we compute the *score*, *i.e.*, the $4\times$ recall at the first four returned images [29]. The overall scores of the compared algorithms, *i.e.*, A-1-A-9 are presented in Figure 13.

- A-1: 57481 classic visual words without Local Feature Refinement (LFR)
- A-2: 44100 Contextual Visual Words (CVWs) based on feature-Group-containing-*Double*-local-Features (GDF) without LFR
- A-3: 44100 CVWs based on GDF with LFR
- A-4: 44100 CVWs based on feature-Group-containing-*Three*-local-Features (GTF) without LFR
- A-5: 44100 CVWs based on GTF with LFR
- A-6: Bundled feature [26] with 758350 classic visual words without LFR
- A-7: Bundled feature [26] with 758350 classic visual words with LFR
- A-8: 90000 CVWs based on GDF with LFR
- A-9: 122500 CVWs based on GDF with LFR

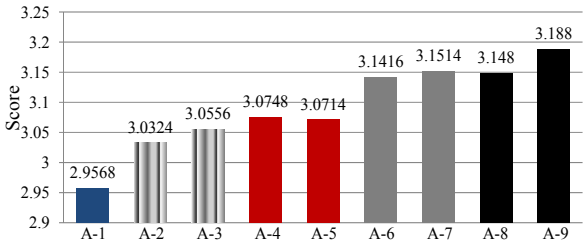


Figure 13. The comparisons of scores between the six groups

From the comparison between A-2 and A-4, we can conclude that Local Feature Group (LFG) containing three local features (A-4) shows better performance than the LFG containing two local features (A-2). This illustrates the effectiveness of preserving more spatial information in contextual visual vocabulary. It is also noticeable that the local feature refinement improves the LFG containing two local features (A-2, A-3, A-8, A-9), but is not helpful for LFG containing three local features (A-4, A-5). This might be because that the local feature refinement makes the local features in the image sparse, making it difficult to detect stable LFGs containing three features with the detector shown in Figure 10. Obviously from the comparisons between A-6 and A-7, the local feature refinement improves the performance of bundled feature. This might be because the unstable noisy local features disturb the meaningful spatial configurations in images, thus removing them significantly improves the robustness of the computed spatial relationship within the bundled feature. From the scores of A-9 and A-8, we can conclude that the contextual visual vocabulary performs better than the bundled feature with more compact vocabulary size *i.e.*, 122500 contextual visual words vs. 758350 classic visual words utilized in bundled feature.

Besides the comparison of score we also compare the memory consumption *i.e.*, the size of the index file, of classic visual word, bundled feature, and contextual visual word. The sizes of their index files after indexing 10200 images are compared in Table 1.

Table 1. The comparison of the size of the index file

A-1	A-2	A-3	A-4	A-5	A-6	A-7	A-8	A-9
35 MB	43.5 MB	39.5 MB	32.7 MB	25.4 MB	92.6 MB	80.3 MB	40.4 MB	41.8 MB

Intuitively from the Table 1, the local feature refinement decreases the index size of both the contextual visual word and the bundled feature. Meanwhile, it can be observed that the bundled feature needs larger memory to load the index file for image retrieval. This is because for each visual word in an image, it needs to store certain numbers of 19-bit "bundled bits [26]", which records the spatial context of visual words in a MSER (Maximally Stable Extremal Regions) [33]. The bundled bit number equals to the number of bundled features where this visual word appears. Thus, in addition to image ID and the visual word frequency information, extra space is needed to store the "bundled bits", resulting in the large index file. Differently, for contextual visual word based image indexing, we only need to store the image ID and visual word frequency for each contextual visual word. The spatial information is combined in individual contextual visual words, *i.e.*, each contextual visual word contains spatial contextual clues. Thus, the contextual visual vocabulary based image indexing captures spatial contextual information with very compact index size. The reason why A-2, A-3, A-8, A-9 show larger index sizes than A-1 is because more local feature groups can be extracted than the local features. As a result, the number of contextual visual words in images is generally larger than the number of classic visual words.

The other experiment is carried out on the large-scale image dataset to test the performance of contextual visual word in large-scale image retrieval. The images with ground truth are indexed together with the other ones. In the retrieval process, all the images with ground truth are used as queries. For each query, we compute the Mean Average Precision (MAP), which takes the average precision across all different recall levels in the first 40 returned images. The overall MAPs of the groups A-7, A-8, A-9 and A-10 are illustrated in Figure 14.

- A-10: 758350 classic visual words with local feature refinement

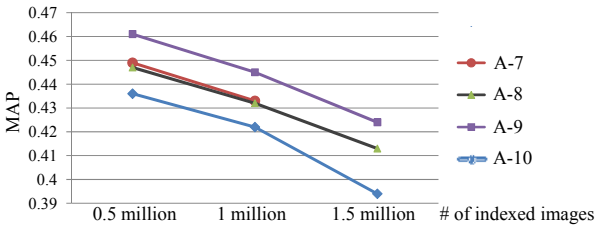


Figure 14. The overall MAP obtained by the four algorithms with different numbers of indexed images

From Figure 14, it is clear that, the bundled feature performs better than the classic visual word, *i.e.*, the A-10. This is because it combines more spatial cues by bundling several visual words together. It is also obvious that the A-9 outperforms the bundled feature when 0.5 million and 1 million images are indexed. This implies the stronger descriptive power of the contextual vocabulary. The MAPs corresponding to different image numbers clearly show that, augmenting the image database causes the performance degradation of both classic visual word and contextual visual vocabulary. However, the contextual vocabulary still performs better than classic visual word with more compact vocabulary size (*i.e.*, 122500 and 90000 contextual vocabularies vs. 758350 classic visual words). The reason why we do not test the bundled feature in larger image databases (*i.e.*, 1.5 million images) is because the index size of bundled feature is large, and thus 1 million is the maximum image number that the 4.0GB memory of our computer could handle with bundled feature.

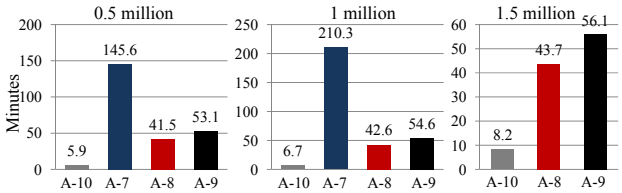


Figure 15. Comparisons of efficiency between the classic visual word, bundled feature, and contextual visual word

Besides the comparisons of MAP, the efficiency is compared in Figure 15. From the figure, it can be observed that bundled feature is time consuming. This is because the online spatial verification between two bundled features is carried out during the retrieval process [26]. The time consumed by A-8 and A-9 is mainly due to the computation of discriminant group distance between local feature groups and the contextual visual vocabulary tree. However, once the image is transformed into BoCW representation, the retrieval operation can be finished efficiently. This is the reason why the A-8 and A-9 show similar efficiency with different numbers of indexed images.

From this experiment, we can conclude that, the contextual visual vocabulary shows better performance than the bundled feature [26] and the classic visual word. In addition, it is proven better than the bundled feature in efficiency and memory consumption. Examples of contextual visual vocabulary based near-duplicate image retrieval are illustrated in Figure 16.

Figure 16. Examples of near-duplicate image retrieval based on contextual visual word. The most left images are the queries.

4.3 Topic Word based Image Re-ranking

Image search re-ranking is getting popular in recent years [34-36]. The goal is to resort the images returned by text-based search engines according to their visual appearances or tag information to make the top re-ranked images more relevant to the query.

In our experiment, for the images retrieved with a query Q, we utilize the Latent Semantic Analysis (LSA) [37] to compute the importance of each visual word to Q. The most important ones are identified as topic words, which are then utilized for image re-ranking. Proposed in natural language processing, LSA analyzes the relationships between a set of documents and terms they contain by producing a set of concepts related to the documents. Similarly, for the image set, we build a $m \times n$ sized term-document matrix M, where n is the number of documents, and each m-dimensional vector is a contextual visual word histogram. According to LSA, M can be decomposed with Singular Value Decomposition,

$$M = U \sum V^T \tag{17}$$

where U and V are orthonormal matrices and $\sum$ is a $k \times k$ sized diagonal matrix. Each diagonal element in $\sum$ represents a latent topic found in M. We keep the largest t elements and set the rest to zero, resulting in a new matrix $\sum^*$. Intuitively, since many returned images are related with the query Q and show similar visual topics, it is reasonable to keep the most dominant latent topics and filter the noisy ones. Hence, t is experimentally set as $0.1 \bullet k$. By replacing $\sum$ with $\sum^*$ in Eq. 17, we obtain the matrix M^*, then we compute the importance of each visual word as:

$$w_i = \sum_{j=1}^{n} M^*_{i,j} \tag{18}$$

where w_i denotes the importance of visual word i to query Q. Consequently, the visual words with high importance can be selected as the topic words for Q.

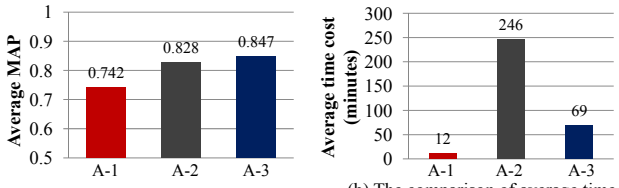
Based on topic words, we utilize the strategy illustrated in Eq. 19 to compute the rank value of each image.

$$Rank^{(I)} = \sum_{j=1}^{T} tfidf_j^{(I)} \bullet w_j \tag{19}$$

where $Rank^{(I)}$ denotes the rank value of image I. T is the total number of the topic words, which is experimentally set as 200. $tfidf_j^{(I)}$, which is computed with Eq. 16, stands for the TF-IDF weighting of the topic word j in image I. With Eq. 19, the image re-ranking task can be completed by sorting the images based on their rank values.

To conduct this experiment, we first download images from Google Image with keywords of location such as "Great Wall", "Eiffel Tower", *etc.* From the downloaded images, we select 40 categories, within which we keep 250 relevant images and 100 irrelevant ones and disarrange them to lose the initial rank information. Finally, we build a dataset containing 14000 images, all of which are annotated as positive or negative. Based on the collected dataset, we compare three algorithms listed below:

- A-1: 758350 classic visual words with topic word based re-ranking.
- A-2: the state-of-the-art VisualRank [34]
- A-3: 122500 contextual visual words with topic word based re-ranking



(a) The comparison of average MAP

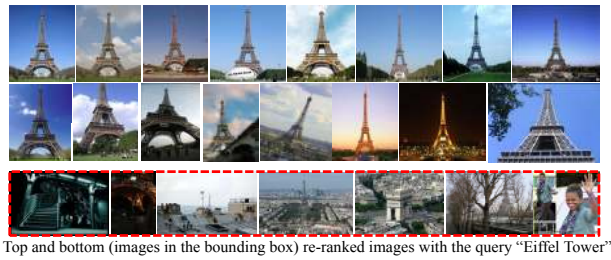
(b) The comparison of average time needed for re-ranking 350 images

Figure 17. The comparisons of MAP and efficiency

The MAP computed in the top 250 re-ranked images is utilized for evaluation. The overall MAP of the 40 categories is presented in Figure 17(a). The average time required to re-rank 350 images is compared in Figure 17(b).

From the experimental result illustrated in Fig. 17(a), it is clear that our contextual visual word outperforms the traditional visual word and shows slightly better performance than the state-of-the-art VisualRank [34]. This demonstrates the effectiveness of the proposed contextual visual vocabulary and the proposed re-ranking algorithm. Moreover, it is necessary to point out that in Fig. 17(b), the topic word based image re-ranking with contextual visual vocabulary is about 3 times faster than the VisualRank. Similar to the image retrieval, the most time consuming operation is transforming each image into the BoCW representation. After that, the image re-ranking can be finished in about 10 minutes for 350 images. The VisualRank needs to compute an image-to-image similarity matrix based on their contained local features with Locality Sensitive Hashing [38]. For each local feature in the image, 120 hash functions, *i.e.*, 40 hash tables, each containing 3 hash functions [34], need to be computed. Therefore for each image, VisualRank has to compute about 200000 hash functions, making it time consuming. Figure 18 presents some examples of the re-ranked images by the topic word based image re-ranking.

Top and bottom (images in the bounding box) re-ranked images with query "The Parthenon"



Top and bottom (images in the bounding box) re-ranked images with the query "Eiffel Tower"

Figure 18. Some examples of the top and bottom re-ranked images based on the proposed algorithm

5. CONCLUSIONS AND FUTURE WORK

In this paper, we propose the local feature refinement strategy and the discriminant group distance for contextual visual vocabulary generation. In contrast to previous methods, we take groups of local features into consideration instead of treating the local features independently. The advantage is that besides the appearance information, rich image local spatial contextual cues (*i.e.*, scale and orientation) are preserved in single contextual visual word. In addition, with the discriminant group distance, the generated contextual visual vocabulary captures more semantic contexts. Comparisons with the state-of-the-art algorithms show that the proposed vocabulary is promising in large-scale near-duplicate image retrieval and image search re-ranking.

Our future work will focus on three aspects. 1) The contextual visual vocabulary will be tested in visual recognition tasks. 2) We will study to leverage the state-of-the-art distance metric learning algorithms in our framework. The benefit of distance metric learning will be further tested. 3) In current work, only images with single labels are considered. We will extend our algorithm to images with multiple labels and multiple objects.

6. ACKNOWLEDGEMENT

This work was supported in part by National Natural Science Foundation of China: 60773136 and 60833006, in part by National Basic Research Program of China (973 Program): 2009CB320906, in part by Beijing Natural Science Foundation: 4092042, and in part by Akiira Media Systems, Inc. for Dr. Qi Tian.

7. REFERENCES

[1] J. Sivic and A. Zisserman. Video Google: a text retrieval approach to object matching in videos. *Proc. ICCV*, 2003.

[2] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. *Proc. CVPR*, pp. 2161-2168, 2006.

[3] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2): 91-110, Nov. 2004.

[4] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. *Proc. ICCV*, pp. 17-21, 2005.

[5] S. Lazebnik and M. Raginsky. Supervised learning of quantizer codebook by information loss minimization. *T-PAMI*, 31(7): 1294-1309, July 2009.

[6] F. Perronnin. Universal and adapted vocabularies for generic visual categorization. *T-PAMI*, 30(7): 1243-1256, July 2008.

[7] J. Yuan, Y. Wu, and M. Yang. Discovery of collocation patterns: from visual words to visual phrases. *Proc. CVPR*, 2007.

[8] Y. Zheng, M. Zhao, S. Y. Neo, T. Chua, and Q. Tian. Visual synset: a higher-level visual representation. *Proc. CVPR*, 2008.

[9] D. Liu, G. Hua, P. Viola, and T. Chen. Integrated feature selection and higher-order spatial feature extraction for object categorization. *Proc. CVPR*, pp. 1-8, 2008.

[10] S. Savarese, J. Winn, and A. Criminisi. Discriminative object class models of appearance and shape by correlations. *Proc. CVPR*, 2006.

[11] F. Perronnin and C. Dance. Fisher kernels on visual vocabulary for image categorization. *Proc. CVPR*, pp. 1-8, 2007.

[12] J. Liu, Y. Yang, and M. Shah. Learning semantic visual vocabularies using diffusion distance. *Proc. CVPR*, 2009.

[13] L. Yang, P. Meer, and D. J. Foran. Multiple class segmentation using a unified framework over mean-shift patches. *Proc. CVPR*, 2007.

[14] J. Winn, A. Criminisi, and T. Minka. Object categorization by learning universal visual dictionary. *Proc. ICCV*, pp. 17-21, 2005.

[15] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. *Proc. CVPR*, pp. 2169-2178, 2006.

[16] K. Grauman and T. Darrell. The pyramid match kernel: discriminative classification with sets of image feature. *Proc. ICCV*, pp. 1458-1465, 2005.

[17] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. *CVPR*, 2009.

[18] F. Moosmann, E. Nowak, and F. Jurie. Randomized clustering forests for image classification. *T-PAMI*, 30(9): 1632-1646, Sep. 2008.

[19] M. Marszalek and C. Schmid. Spatial weighting for bag-of-features. *Proc. CVPR*, pp. 2118-2125, 2006.

[20] L. Wu, S. C. H. Hoi, and N. Yu. Semantic-preserving bag-of-words models for efficient Iimage annotation. *Proc. ACM workshop on LSMRM*, pp. 19-26, 2009.

[21] Y. Jiang, C. Ngo, and S. Chang. Semantic context transfer across heterogeneous sources for domain adaptive video search. *Proc. ACM Multimedia*, 2009.

[22] F. Wang, Y. G. Jiang, and C. W. Ngo. Video event detection using motion relativity and visual relatedness. *ACM Multimedia*, 2008.

[23] D. Xu and S. F. Chang. Video event recognition using kernel methods with multilevel temporal alignment. *T-PAMI*, 30(11): 1985-1997, Nov. 2008.

[24] X. Zhou, X. D. Zhuang, S. C. Yan, S. F. Chang, M.H. Johnson, and T.S. Huang. SIFT-bag kernel for video event analysis. *Proc. ACM Multimedia*, pp. 229-238, 2008.

[25] S. Zhang, Q. Tian, G. Hua, Q. Huang, and S. Li. Descriptive visual words and visual phrases for image applications. *Proc. ACM Multimedia*, 2009.

[26] Z. Wu, Q. Ke, and J. Sun. Bundling features for large-scale partial-duplicate web image search. *Proc. CVPR*, 2009.

[27] O. Chum, M. Perdoch, and J. Matas. Geometric min-hashing: finding a (thick) needle in a haystack. *Proc. CVPR*, 2009.

[28] M. Perdoch, O. Chum, and J. Matas. Efficient representation of local geometry for large scale object retrieval. *Proc. CVPR*, 2009.

[29] H. Jegou, M. Douze, C. Schmid, and P. Petrez. Aggregating local descriptors into s compact image representation. *Proc. CVPR*, 2010.

[30] P. Viola and M. Jones. Robust real-time face detection. *ICCV*, 2001.

[31] A. Globerson and S. Roweis. Metric learning by collapsing classes. *Adv. in Neu. Info. Proce. Sys.*, 18: 451-458, 2006.

[32] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: a large-scale hierarchical image database. *CVPR*, 2009.

[33] J. Matas, O. Chum, M. Urba, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. *Proc. BMVC* , 2002.

[34] Y. Jing and S. Baluja. VisualRank: applying PageRank to large-scale image search. *T-PAMI*, 30(11): 1877-1890, 2008.

[35] X. Tian, L. Yang, J. Wang, Y. Yang, X. Wu, and X. Hua. Bayesian video search reranking. *Proc. ACM Multimedia*, pp. 131-140, 2008.

[36] D. Liu, X. Hua, L. Yang, M. Wang, and H. Zhang. Tag Ranking, *Proc. WWW*, 2009.

[37] S. Deerwester, S. Dumais, and R. Harshman. Indexing by latent semantic analysis. *J-ASIS*, 41(6): 391-407, 1990.

[38] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. *Proc. VLDB*, pp. 518-529, 1999.