Exploring Spatial Correlation for Visual Object Retrieval

Citation for published version:
Shi, M, Sun, X, Tao, D, Xu, C, Baciu, G & Liu, H 2015, 'Exploring Spatial Correlation for Visual Object Retrieval', *ACM Transactions on Intelligent Systems and Technology*, vol. 6, no. 2, 24.
https://doi.org/10.1145/2641576

Digital Object Identifier (DOI):
10.1145/2641576

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
ACM Transactions on Intelligent Systems and Technology

Exploring Spatial Correlation for Visual Object Retrieval

MIAOJING SHI and XINGHAI SUN, Peking University
DACHENG TAO, University of Technology, Sydney
CHAO XU, Peking University
GEORGE BACIU, Hong Kong Polytechnic University
HONG LIU, Peking University

Bag-of-visual-words (BOVW) based image representation has received intense attention in recent years and has improved content based image retrieval (CBIR) significantly. BOVW does not consider the spatial correlation between visual words in natural images, and thus, biases the generated visual words towards noise when the corresponding visual features are not stable. This paper outlines the construction of a visual word co-occurrence matrix by exploring visual word co-occurrence extracted from small affine-invariant regions in a large collection of natural images. Based on this co-occurrence matrix, we first present a novel high-order predictor to accelerate the generation of spatially correlated visual words, and a penalty tree (PTree) to continue generating the words after the prediction. Subsequently, we propose two methods of co-occurrence weighting similarity measure for image ranking: Co-Cosine and Co-TFIDF. These two new schemes down-weight the contributions of the words that are less discriminative because of frequent co-occurrences with other words. We conduct experiments on *Oxford* and *Paris Building* datasets, in which the *ImageNet* dataset is used to implement a large scale evaluation. Cross dataset evaluations between *Oxford* and *Paris* datasets, *Oxford* and *Holidays* datasets are also provided. Thorough experimental results suggest that our method outperforms the state-of-the-art without adding much additional cost to the BOVW model.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms: Algorithms, Experimentation, Performance

Additional Key Words and Phrases: BOVW, spatial correlation, high-order predictor, penalty tree, Co-Cosine, Co-TFIDF

ACM Reference Format:
Shi, M., Sun, X., Tao, D., Xu, C., Baciu, G., and Liu, H. 2014. Exploring Spatial Correlation for Visual Object Retrieval. *ACM Trans. Intell. Syst. Technol.* xx, xx, Article 00 (20), 21 pages.
DOI:http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

Bag-of-visual-words (BOVW) based image representation has received intense attention in recent years and has improved content-based image retrieval (CBIR) significantly [Sivic and Zisserman 2003]. BOVW represents an image as a visual document composed of distinctive visual words which is very important for both the effectiveness

Author's address: M. Shi, X. Sun, C. Xu and H. Liu, Key Laboratory of Machine Perception (Ministry of Education), Peking University, Beijing, 100871, China; e-mail: {shimj, sunxh, xuchao, liuhong}@cis.pku.edu.cn; D. Tao, Centre for Quantum Computation & Intelligent Systems and the Faculty of Engineering & Information Technology, University of Technology, Sydney, 235 Jones Street, Ultimo, NSW 2007, Australia; email: dacheng.tao@uts.edu.au; G. Baciu, Department of Computing, Hong Kong Polytechnic University, Hong Kong; email: csgeorge@comp.polyu.edu.hk.

DOI:http://dx.doi.org/10.1145/0000000.0000000

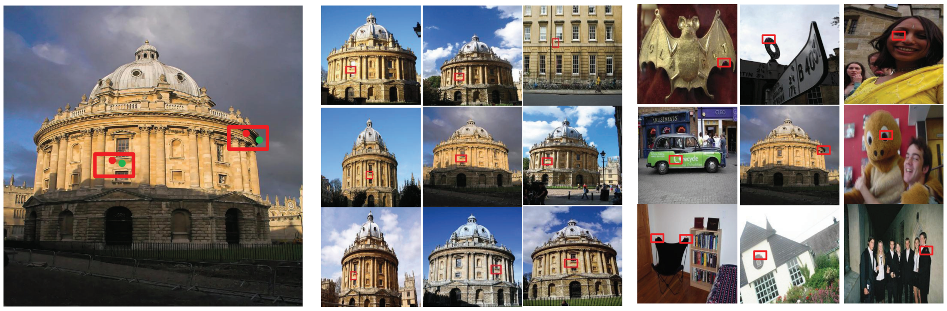


Fig. 1. An illustration of different co-occurring patterns from a $100K$ vocabulary constructed from the *Oxford Building* dataset. First: for one of the selected landmarks, Radcliffe_camera, two pairs of visual words (the red dot and the green dot inside the red frame) are selected with their IDs (17998, 10363), and (43393, 65762); middle: nine relevant images share the same visual word pair (17998, 10363), the locations of the visual word pair vary in the images; third: nine irrelevant images share another pair (43393, 65762).

and efficiency of image retrieval, especially in a large scale database. The visual document is in the same format as a text document, and image retrieval can therefore be improved by many mature text retrieval techniques,and can be run as fast as text retrieval. It has been demonstrated that BOVW is one of the most promising approaches for large scale image retrieval [Nister and Stewenius 2006; Philbin et al. 2007; 2008].

The visual words are derived by clustering and quantizing local features. The scale-invariant feature transform (SIFT) [Lowe 2004] is adopted as the local feature. SIFT features are clustered and quantized into visual words with the K-means algorithm [Nister and Stewenius 2006]. The TFIDF weight [Baeza-Yates and Ribeiro-Neto 1999] is widely used because it up-weights the contribution of a word that occurs frequently in an image with the TF (term frequency), while it down-weights the contribution of a word that commonly occurs in many images with the IDF (inverse document frequency). Image similarity is measured by the cosine distance between the query image and an image in the database, with each entry of the image vector being the TFIDF value. The ranked list is determined according to the values of the similarity scores, it can be re-ranked by taking advantage of user's relevance feedback [Wang and Hua 2011a; 2011b; Tao et al. 2006a; Tao et al. 2006b; Tao et al. 2007; Tao et al. 2009].

Two important issues need to be addressed for BOVW representation: 1) how to quickly map the local feature to the visual word via the correlation between visual words; 2) how to refine the similarity measure by reducing the correlation redundancies between visual words. In respect of the first issue, as is known from state-of-the-art methods, each feature is independently mapped to a word that causes word generation to be one of the most time-consuming steps in BOVW. Regarding the second issue, the TFIDF weight does not take into account the correlation between visual words, which is important for the similarity measure. Like the IDF, a word co-occurring with many words can also be regarded as being less discriminative and should be down-weighted.

This paper presents two approaches: 1) the fast visual word generation method, and 2) the co-occurrence weighting similarity measure. Both approaches are based on the spatial co-occurrence of visual words. Spatial co-occurrence indicates that visual words co-occur in a small spatial region of an image, rather than in the entire image.

We find that the features of natural images correlate substantially, as illustrated in Fig. 1. One local feature's existence can usually indicate the presences or absences of certain other features in its neighborhood. Statistically, we build a visual word co-occurrence matrix to record the co-occurring number of any two visual words in the vocabulary.

Our first perspective is inspired by predictive coding, in which random variables can be predicted from previously observed random variables. As shown in Fig. 1, some visual word pairs, e.g., $(43393{,}65762)$, $(17998{,}10363)$, selected from the $100K$ vocabulary of the *Oxford Building* dataset [Philbin et al. 2007], frequently co-occur in many images, both relevant and irrelevant. In this context, a visual word can be predicted by its co-occurring visual words, which are collected from the image database. We develop a high-order predictor to accelerate the generation of visual words. Here, the high-order predictor refers to the visual word prediction based on multiple co-occurring words. Fewer candidate words can be collected by estimating their posterior probabilities on the multiple co-occurring words. A consequent saving in computation is therefore achieved during the prediction.

The number of the candidate words provided by the high-order predictor is usually limited. It limits the precision of the prediction. To achieve high precision, we introduce a new tree, named the penalty tree (PTree), to continue the search. It is developed from the fast library for approximate nearest neighbor (FLANN) [Muja and Lowe 2009]: each visual word is represented by a leaf node in the tree, given a query feature, its nearest visual word will not be found without hundreds of backtrackings, or iterative searches, in the tree. Since some visual words have already been searched and compared during the prediction of the high-order predictor, it is clear that the searches of successive nodes of the tree are no longer independent, and those nodes that have the bulk of their leaf nodes searched by the high-order predictor are no longer likely to be the optimal entries during the search. Inspired by this, we penalize the search priorities of those nodes in PTree. A sigmoid function is used to calculate the penalty term for each node based on the number of its leaf nodes that have been searched by the high-order predictor.

Our second contribution is to embed the co-occurrence matrix into the similarity measure for image ranking. Conventionally, TFIDF is utilized as a measurement of the importance of a visual word in retrieving an image. It is proposed under the assumption that visual words independently compose an image. However, they are not independent because there exist co-occurrence redundancies between visual words: if a visual word co-occurs with many words many times, its uniqueness and distinctiveness decline, such as the word occurring in many irrelevant images in Fig. 1, the third figure. We design two new measurements to refine either the cosine value or the TFIDF value in a similarity measure. These two new schemes reduce the weight of every visual word by subtracting the co-occurrence redundancies from them.

This paper is an extension of our previous work [Shi et al. 2012], in which we proposed a high-order predictor for visual word generation and a co-occurrence weighting cosine (Co-Cosine) similarity measure for image ranking. In this study, we improve the performance of the high-order predictor by introducing the PTree in the generation, and provide a careful discussion of its computational complexity to show the efficiency and effectiveness of PTree; also, by subtracting the visual word correlation redundancies from TFIDF, we extend the similarity measure from Co-Cosine to co-occurrence weighting TFIDF (Co-TFIDF), and theoretical analysis has been provided to prove the effectiveness of the proposed Co-TFIDF; thorough experimental results of PTree and Co-TFIDF have been carried out in comparison with both our previous work and other representative approaches.

Related works are introduced in Section 2. Our proposed fast visual word generation algorithm and refined similarity measures will be presented in Sections 3 and 4, respectively. Section 5 evaluates the experiment and the work is concluded in Section 6.

2. RELATED WORK

This section reviews the state-of-the-art methods in two related aspects: 1) visual word generation for local features; 2) image ranking by exploiting spatial correlation.

2.1. Visual Word Generation

The common method of visual word generation is to index the visual words through a multi-branch tree. Representative tree based algorithms include KD-tree [Beis and Lowe 1997; Arya et al. 1998; Silpa-Anan and Hartley 2008] and K-means tree [Uhlmann 1991; Liu et al. 2004; Nister and Stewenius 2006]. Arya et al. [1998] designed a priority queue to speed up the search; Anan et al. [2008] utilized multiple random KD-trees (RKD) simultaneously to search words; Uhlmann [1991] proposed an "RkNN" tree, which evaluated an efficient approximative search in arbitrary metric spaces; Nister et al. [2006] presented a new K-means tree by accessing a single leaf hierarchically, that is, the hierarchical K-means (HKM) tree; Muja et al. [2009] selected the two tree structures (RKD and HKM) and utilized a fast library for approximate nearest neighbors (FLANN) to automatically determine the better algorithm and parameters for a given dataset.

Typical algorithms search a word for each feature in an image independently of other features, and some researchers have already tried to exploit the nearest neighbor information for certain word generation, such as reciprocal neighbors [Jegou et al. 2011a] and product quantization [Jegou et al. 2011b]. A weighting tree (W-tree) was presented in [Shi et al. 2013] to optimize the priority search based on the co-occurring probabilities between visual words, each node is assigned with a probabilistic weight to re-direct the searching path to be close to its global optimum within a small number of backtrackings; a high-order predictor was designed in [Shi et al. 2012] to accelerate the search by indexing candidate visual words by their posterior probabilities; to enhance the search precision, in this paper, a PTree is specifically introduced to continue the search after the prediction, which optimizes the priority search based on the number of leaf nodes that have been searched in the high-order predictor.

2.2. Image Ranking

Spatial correlation is extensively explored in image retrieval [Philbin et al. 2007; Zhang et al. 2009; Yang et al. 2010; Zhang et al. 2010; Tang et al. 2011; Li et al. 2011; Zhang et al. 2011; Shen et al. 2012]. General approaches adopt bundling features in concrete structures [Huang et al. 2004] and segments [Guo et al. 2009], for example, the bounding box in [Philbin et al. 2007] was manually initialized and different weighting terms were added to the visual words inside and outside the bounding box [Yang et al. 2010]. In [Zhang et al. 2009; Zhang et al. 2010; Li et al. 2011], features are bundled in their affine-invariant regions and taken as contextual visual phrases; these phrases are leveraged to provide more information for image indexing and retrieval. Furthermore, spatial correlation has been embedded in building a dictionary of contextual synonyms in [Philbin et al. 2008; Tang et al. 2011], or constraining the similarity measure in [Zhang et al. 2011; Shen et al. 2012].

The spatial co-occurrence used in this paper seems to resemble the visual phrase [Zhang et al. 2010], but in fact it is utilized in a totally different way. A visual phrase can be considered as kind of feature expansion to provide better image representation [Zhang et al. 2012; Zhang et al. 2011]. By contrast, our spatial co-occurrence is used to reduce the weight of a single word: in other words, it removes the correlation redundancies from each local feature. Research that shares a similar motivation in down-weighting the TFIDF of visual words in consideration of their correlations can be found in [Jegou et al. 2009; Wang et al. 2011]. Even so, they are intrinsically

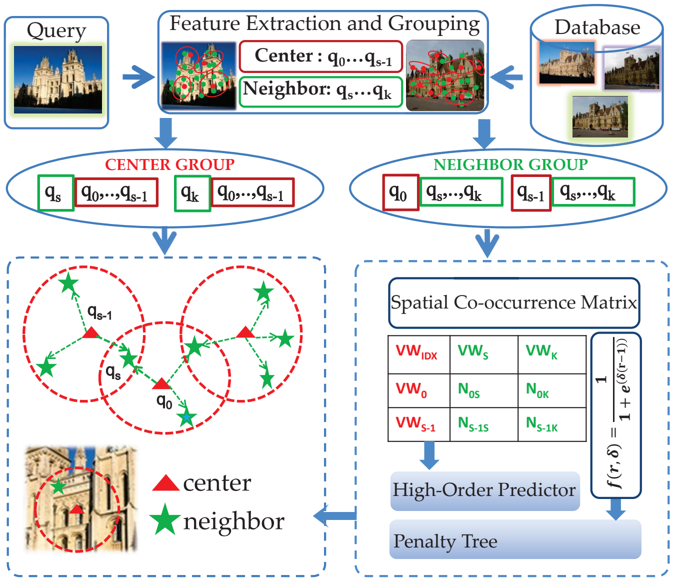


Fig. 2. Fast visual word generation algorithm. For the images in the database, neighboring features are grouped in their affine-invariant regions; in the right block, a co-occurrence matrix is constructed offline after mapping the features in *Neighbor Group* to their corresponding visual words; for test images, we bundle those centers that contain the same feature in their affine-invariant regions together as a *Center Group*. A high-order predictor is built for fast visual word prediction, while the PTree index with penalty function $f(r, \delta) = \frac{1}{1+e^{(\delta(r-1))}}$ is introduced for precise search. The left block indicates the prediction of the high-order predictor after the exact mapping of the center features.

different: in [Wang et al. 2011], the authors down-weight the contribution of visual word from object view, however, Co-TFIDF downweights TFIDF from the word view; in [Jegou et al. 2009], the authors downweight TFIDF by considering the burstiness of the visual words in one image, while Co-TFIDF considers the spatial co-occurrence of visual words over the entire database. Compared to existing approaches, two points make this work unique, one is the observation of the visual word co-occurrence, it is in a small spatial region rather than the entire image. This local information makes the down-weighting in TFIDF more specific and accurate to every single word. The other point is the statistic of visual word co-occurrence, it is over the entire database rather single image, this overall summarization makes the down-weighting more efficient to implement and meanwhile statistically meaningful to every single image. Other important papers striving to increase the discriminative attributes of visual words include those such as the contextual dissimilarity measure [Jegou et al. 2010], reciprocal neighborhoods [Jegou et al. 2011a] [Qin et al. 2011] and co-missing words [Jegou et al. 2012].

3. FAST VISUAL WORD GENERATION

Since many features co-occur in images, we construct a spatial visual word co-occurrence matrix [Shi et al. 2012; Xu et al. 2011] to speed up the word generation of the query image. The proposed scheme is shown in Fig. 2, the word co-occurrence matrix is constructed and utilized in the following way: every feature is grouped with its neighbors in the affine-invariant region, which is extracted using Harris detector [Mikolajczyk and Schmid 2004]. We name every group of features the *Neighbor*

Group and map those features in *Neighbor Groups* into their corresponding visual words. Visual words in the same group are regarded as co-occurring visual words and the visual word co-occurrence matrix is constructed over the entire database. Instead of recording the high-order information, only the co-occurrence of any two visual words is considered in the co-occurrence matrix and the overhead storage is thus saved. On the other hand, we manage to predict a visual word based on its high-order co-occurrences with a couple of neighboring words: we first randomly sample a small number of visual features as centre features in a query image and map them into their exact visual words. We bundle those centers $(q_0, ..., q_{s-1})$ that contain the same neighboring feature (i.e., q_s) in their affine-invariant regions together as a *Center Group*. Consequently, certain feature q_s's candidate visual word set can be collected from the co-occurrence matrix corresponding to the lists of its center words $(w_0, ..., w_{s-1})$. We propose a high-order predictor to calculate the approximate posterior probabilities for these candidate words. Each candidate's high-order co-occurrences with center words are decomposed into its first-order co-occurrence with every center word. We select K-candidates with the top K-maximal probabilities to compute their Euclidean distances to the feature q_s; the one with the minimal Euclidean distance is regarded as the nearest neighbor.

A PTree developed from FLANN is introduced to continue the search after the prediction of a high-order predictor. A search is usually restricted to a maximum number of searched nodes in the tree; increasing the number of the searched nodes improves the search performance. Suppose the maximum number of searched nodes is m, usually, m is larger than the size v of the candidate set in the high-order predictor, so PTree is consequently adopted for the $m - v$ comparisons. If a Euclidean distance between q_s and w_s found by PTree is smaller than the optimal value found by the high-order predictor, we will update the corresponding optimal $\overset{*}{w}_s$.

3.1. High-Order Predictor

Inspired by predictive coding, we propose a high-order predictor to predict the corresponding word of a feature depending on the words of its neighboring features.

Given a couple of visual words as sampled centre words, $S = \{w_0, ..., w_{s-1}\}$, they contain the same feature in their affine-invariant regions and we bundle them together as a *Center Group*, as shown in Fig. 2. To predict the word of a certain feature, q_s, for every center word in its *Center Group*, we look up the co-occurrence matrix and bundle every center's co-occurring visual words. We regard these bundled co-occurring visual words as the possible visual words of q_s, as they all co-occur with the centers in *Center Group*. These possible visual words are taken as the candidate set $W = \{w_{S0}, ..., w_{Sv-1}\}$ of q_s. The Bayesian criterion is adopted to predict the optimal visual word of this feature in W: the word $\overset{*}{w}_s$ in W with the maximum posterior probability is predicted,

$$\overset{*}{w}_s = \arg\max_{\hat{w}_s \in W} p(\hat{w}_s | \, w_0, ..., w_{s-1}) \tag{1}$$

and this conditional probability can be computed from the joint probability,

$$p(\hat{w}_s \,| w_0, ..., w_{s-1}) = \frac{p(\hat{w}_s, w_0, ..., w_{s-1})}{p(w_0, ..., w_{s-1})} \tag{2}$$

where $p(\hat{w}_s, w_0, ..., w_{s-1})$ is the joint probability of $\hat{w}_s, w_0, ..., w_{s-1}$ in the neighborhood. We decompose $p(\hat{w}_s, w_0, ..., w_{s-1})$:

$$p(\hat{w}_s, w_0, ..., w_{s-1}) = p(w_{s-1} | \, \hat{w}_s, w_{s-2}, ..., w_0)...p(w_1 | \, \hat{w}_s, w_0)p(w_0 | \, \hat{w}_s)p(\hat{w}_s) \tag{3}$$

$p(w_{s-1}|\hat{w}_s, w_{s-2}, ..., w_0), ..., p(w_1|\hat{w}_s, w_0), p(w_0|\hat{w}_s)$ measure the conditional probabilities that certain word may co-occur with its nearby words. The prior $p(\hat{w}_s)$ can be estimated from $N(\hat{w}_s)$. To formulate (3), we assume the conditional independence of the sampled words $w_0, w_1, ..., w_{s-1}$, for they have already been generated from their corresponding features and there is no need to measure the probabilities of their possible visual words. Hence, in this manner, $p(w_0, ..., w_{s-1})$ can be approximated as $\prod_{w_i \in S} p(w_i)$, while $p(w_{s-1}|\hat{w}_s, w_{s-2}, ..., w_0)$,...,$p(w_0|\hat{w}_s)$ are only dependent on the unknown word $\hat{w}_s$, so they can be approximated by the first-order probabilities on $\hat{w}_s$,

$$p(\hat{w}_s, w_0, ..., w_{s-1}) \approx p(\hat{w}_s) \prod_{w_i \in S} p(w_i|\hat{w}_s) \tag{4}$$

$p(w_i|\hat{w}_s)$ can be estimated from the co-occurring number of $N(w_i, \hat{w}_s)$ collected in the co-occurrence matrix. The posterior probability in (1) corresponds to the following decomposition:

$$\begin{aligned} \overset{*}{w}_s &= \arg\max_{\hat{w}_s \in W} p(\hat{w}_s | w_0, ..., w_{s-1}) \\ &\approx \arg\max_{\hat{w}_s \in W} \frac{p(\hat{w}_s) \prod_{w_i \in S} p(w_i|\hat{w}_s)}{\prod_{w_i \in S} p(w_i)} \\ &= \arg\max_{\hat{w}_s \in W} \frac{N(\hat{w}_s) \prod_{w_i \in S} \frac{N(w_i, \hat{w}_s)}{N(\hat{w}_s)}}{\prod_{w_i \in S} N(w_i)} \end{aligned} \tag{5}$$

the optimal visual word $\overset{*}{w}_s$ indicates that the small region consisting of the words $w_0, ..., w_{s-1}, \overset{*}{w}_s$ is the most probable co-occurring pattern at the current location.

Since the co-occurrence matrix is sparse, zero terms in the matrix will affect the calculation in (5), which makes the probability a zero value. Zero means that $\hat{w}_s$ doesn't usually co-occur with current word w_i according to the co-occurrence matrix, however, our target is to find $\hat{w}_s$ that usually co-occurs with the majority visual words in its central group in the neighbourhood. When we compute $p(\hat{w}_s | w_0, ..., w_{s-1})$ with more zero terms in $N(w_i, \hat{w}_s)$, it is less likely $\hat{w}_s$ would be the optimum in (5). In this context, to solve the problem, we could first group $\hat{w}_s$ based on its number of zero terms in $N(w_i, \hat{w}_s)$. In the same group, we compute $p(\hat{w}_s | w_0, ..., w_{s-1})$ by simply removing the zero items. For those different groups with different numbers of zero terms, we decide that, for the group with fewer zero terms in $N(w_i, \hat{w}_s)$, its $p(\hat{w}_s | w_0, ..., w_{s-1})$ is larger than that of a group with more zero terms. In fact, in real implementation, we could simply set a very small value ($\ll 1$) for those zero terms $N(w_i, \hat{w}_s)$, which indeed plays the same role.

3.2. Penalty Tree (PTree)

A high-order predictor apparently accelerates visual word generation; however, due to the limited number of the candidates it provids, usually, the generation precision cannot be guaranteed. To achieve high precision, a PTree developed from FLANN is introduced to continue the search: visual words that have been compared in the high-order predictor are taken into account in PTree.

To introduce PTree, we start from a simple example of the priority search in a certain level of a KD-tree. KD-tree is a widely used data-structure for finding nearest-neighbor visual words for image features [Silpa-Anan and Hartley 2008]. The elements stored

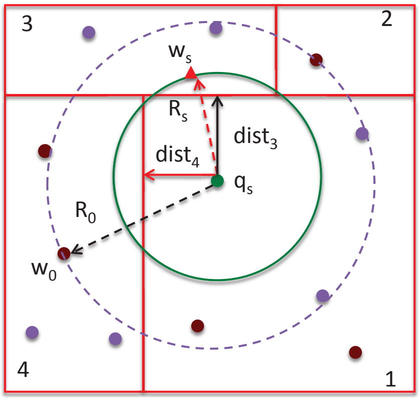


Fig. 3. Priority search of a KD-tree: a query feature q_s is represented by the green dot and its closest neighbor (red triangle) lies in cell 3. A priority search proceeds in the order of the cell distance ($dist$, solid arrow) from the query feature q_s through cell 4. However, the distance (R, dashed arrow) between query feature and candidate visual word is not the same as $dist$. Deep red dots indicate those visual words that have been searched in the prediction of high-order predictor.

in the KD-tree are the high-dimensional visual word vectors in R^d, d is the vector dimension. At the first level (root) of the tree, visual words are split into two halves by a hyperplane orthogonal to a chosen dimension at a threshold value. Each of the two halves of the visual words is recursively split in the same way to create a fully balanced binary tree. At the bottom of the tree, each leaf node corresponds to a single visual word in the vocabulary. It is useful to remark, that each node in the tree corresponds to a cell in R^d, as illustrated in Fig. 3, a search with a query feature lying anywhere in a given leaf cell will lead to the same leaf node. When querying the nearest visual word (w_s, red triangle) of certain feature (q_s, green dot) in the tree, cell distance ($dist$, solid arrow) from query feature to the tree node is calculated at each level. A priority search proceeds in the order of the cell distance through cell 4 to cell 3. The distance (R_0, black dashed arrow) between query feature q_s and the leaf node w_0 in cell 4 is, however, not the same as $dist_4$. As the priority search descends through this cell to the leaf node, it will find that R_0 is indeed larger than the distance to cell 3 ($dist_3$), and thus, more steps are required for the priority search to reach the closest neighbor w_s. A tree utilizing priority search will perform in time $T = O(dlogN) \times n$, where N is the visual word vocabulary size, and n is the number of leaf nodes it visited before reaching the optimal one. Basically, a priority search uses $dist$ to approximate R, n is thereby determined by the minimal distance at each level, $\min dist$.

Considering those leaf nodes (deep red dots) that have already been searched in the high-order predictor, the corresponding non-leaf nodes that have had the bulk of their leaf nodes searched are no longer likely to be the optimal entries during the search. The searches of successive nodes of the tree are no longer independent. Intuitively, if we penalize the search priorities of the nodes according to the number of their searched leaf nodes n_{sb}, so that in Fig. 3, a priority search will not first proceed through cell 4 (two searched leaf nodes in cell 4) but cell 3 (no searched leaf node in cell 3), fewer steps will be taken for q_s to reach its nearest word w_s. Time complexity is therefore determined by both the cell distance $dist$ and the penalty term. In the following, we carefully formulate the proposed PTree using a sigmoid function as the penalty term.

When query feature q_s is searched down a tree, the optimal branch b^* at current level is chosen from the B branches:

$$b^* = \underset{b \in \{1,\ldots,B\}}{\arg\min}\ dist(q_s, c_b) = \underset{b \in \{1,\ldots,B\}}{\arg\min}\ \|q_s - c_b\| \tag{6}$$

where c_b is the descriptor of the tree node associated with the bth branch, B is the branch number per level. In FLANN, for HKM, c_b denotes the 128-d descriptor of current tree node, the best b^* is the one with currently minimal Euclidean distance $dist(q_s, c_b^*)$ from q_s to c_b^*; for RKD, it means the 1-d distance from q_s to the current partitioning dimension center c_b: $dist(q_s, c_b) = |q_s - c_b|$. Notice that although we introduce the penalty term from an example of a KD-tree, the proposed scheme can be implemented on any data structure in FLANN.

The judgment is a local minimal optimization; the global optimum cannot be reached without a large number of backtrackings. Currently, b^* has the locally highest priority to be chosen to proceed down. The other branches cannot be totally pruned, those with shorter distances are added to a priority queue, which is a dynamic structure built while the tree is being searched. The queue will be used to pop the candidate branch during the backtracking, once the local minimum distance to a leaf node is larger than the distance recorded in the queue.

As stated above, those nodes that have the bulk of their leaf nodes (visual words) searched in the high-order predictor, should have their search priorities penalized. We adopt a sigmoid function to calculate the penalty term:

$$f(r_b, \delta) = \frac{1}{1 + e^{(\delta(r_b - 1))}},\ r_b = n_{sb}/n_b, 0 \leq r_b \leq 1, \tag{7}$$

for certain node associated with branch b, n_{sb} is the number of its leaf nodes that have already been searched, and n_b is the total number of its leaf nodes. r_b is the ratio of n_{sb} and n_b, $r_b = n_{sb}/n_b$, the larger r_b is, the heavier the priority of branch b should be penalized. To optimize the search priorities in (6), we use $\frac{dist}{f}$:

$$\tilde{b^*} = \underset{b \in \{1,\ldots,B\}}{\arg\min}\ \frac{dist(q_s, c_b)}{f(r_b, \delta)} = \underset{b \in \{1,\ldots,B\}}{\arg\min}\ ||q_s - c_b||(1 + e^{(\delta(r_b - 1))}), \tag{8}$$

where $f(r_b, \delta)$ is the penalty term for bth branch, it depends on the ratio r_b and a predefined δ. When $\delta \to \infty$, the sigmoid function will be a threshold function. In general, we set δ to be a large value (20, in this paper) so that we only penalize those nodes with large r_b, without greatly affecting the original distance judgment.

For those pruned nodes in the queue, suppose the original sequence in the priority queue Qu is determined by:

$$Qu : \{dist_1 < \ldots < dist_j < \ldots < dist_Q\}, \tag{9}$$

where $dist_j$ denotes the distance from query feature q_s to the jth branch vector c_j, Q is length of the queue. We reorder the sequence with the penalty term f:

$$\tilde{Qu} : \{\frac{dist_1}{f_1} < \ldots < \frac{dist_j}{f_j} < \ldots < \frac{dist_Q}{f_Q}\}, \tag{10}$$

the subscript in (10) only denotes the sequence number rather than the corresponding value in (9). At backtracking stage, the queue will pop the entry with the smallest $\frac{dist}{f}$.

4. IMAGE RANKING

In the BOVW model, once a query image is given, the images in the database are ranked in the order of their similarity scores to the query. One of the general similarity measures is the cosine similarity measure. As previously mentioned, the BOVW model does not take into account the spatial correlation between visual words, so that their co-occurrence redundancies are ignored in the similarity measure. In this section, first, we embed the spatial co-occurrence matrix into the cosine similarity measure, we name the refined similarity measure Co-Cosine; second, we generalize a new co-occurrence weighting scheme Co-TFIDF; third, in Appendix A, we formulate a special case from Co-TFIDF, which is intrinsically related with the similarity measure proposed in [Jegou et al. 2012].

4.1. Cosine Similarity Measure

In BOVW representation, an image is represented by an N-dimensional vector, and an element of the vector is the TFIDF value of a word. The TFIDF weight is commonly used in image ranking [Baeza-Yates and Ribeiro-Neto 1999]. A simple ranking function is the normalized cosine measure:

$$Sim(x, y) = x^T y \tag{11}$$

where y is a query vector, x is the vector of an image in the database, they are all l2-normalized by default. The larger $Sim(x, y)$ is, the more relevant x is to the query image, thus, the higher it is ranked in the returned list.

We embed the co-occurrence information into the cosine similarity measure. Mathematically, the cosine similarity is equivalent to using a unit diagonal matrix I to measure the similarity between two vectors, $Sim(x, y) = x^T Iy$. We then utilize the co-occurrence matrix $\sum$ to refine the similarity measure $x^T \sum y$ for image ranking.

4.2. Co-occurrence Weighting Cosine (Co-Cosine)

We manage to estimate the importance of a visual word depending on its co-occurring attribute with other words. We notice that if a word always occurs in an object, it usually has limited neighboring words; when a word appears in many different objects, it co-occurs with a large number of neighbors; therefore, a visual word with high co-occurrences can be considered to be less discriminative in visual object retrieval. Like IDF, we claim that if a visual word commonly co-occurs with a large number of words, and is therefore less discriminative to the relevance score, we should down-weight its contribution in the similarity measure.

Since our approach focuses on large scale image retrieval, all irrelevant images in the database can be taken as noise and provide negative information in image ranking. We build a co-occurrence matrix $\mathbf{\Sigma} = \{\sigma_{ij}\}$ on the noisy set. Its element $\sigma_{ij} = N(w_i, w_j)$ denotes the co-occurring number between visual word w_j and visual word w_i. Because the number of images relevant to certain query is very small in large scale image retrieval, the negative information of the entire image database obviously overwhelms the trivial positive information of the relevant images. In practice, we do not know the labels of the images before retrieval, so we simply construct the co-occurrence matrix on the entire database and it will naturally reflect the visual word co-occurring distribution in the database. We propose Co-Cosine as follows:

$$Sim(x, y) = x^T (I - \frac{1}{\beta}\Sigma) y = x^T y - \frac{1}{\beta} x^T \Sigma y \tag{12}$$

where $x^T y$ is the basic cosine similarity and $x^T \Sigma y$ is the new term introduced to encode the correlation between two visual words. The coefficient $\frac{1}{\beta}$ allows a continuum of the model between the cosine form and $x^T \Sigma y$. Σ describes the co-occurring distribution of the noisy images, if $x^T \Sigma y$ is comparatively large, it means that the visual words in a current image are more likely to be drawn from the noisy distribution and are irrelevant to the query. To indicate a negative effect, a subtraction is operated.

4.3. Co-occurrence Weighting TFIDF (Co-TFIDF)

Co-TFIDF is proposed from the definition of TFIDF. We know that TFIDF can be interpreted from the information theory: given one image containing T visual words, $w_1, w_2, w_3, ..., w_T$, the occurrence frequency of every visual word in this image is $N_1, N_2, ..., N_T$, the probability of every word occurring in the entire image collection is $D_1, D_2, ..., D_T$. If we assume all visual words in this image are independent, then the joint probability for composing such an image is $X = D_1^{N_1} \times D_2^{N_2} \times ... \times D_T^{N_T}$. The average length of encoding such a visual document (an image) is calculated in terms of its information content $-log(X)/Z$, where Z is the visual document length. TFIDF is the contribution of every word for encoding such a visual document, $\frac{N_i}{Z} \times log\frac{1}{D_i}$, where $\frac{N_i}{Z}$ is TF and $log\frac{1}{D_i}$ is IDF.

As we stated before, those visual words, however, are not independent and mostly occur many times because of their high co-occurrences with other words. To measure their real importance in retrieving one image, the co-occurrence redundancies in their TFIDF values have to be removed. One spontaneous idea is to subtract each visual word's co-occurrences $\frac{1}{\beta}\sum_j N(w_i, w_j)$ with all the other words from its own occurrence, so as to find its independent occurrence frequency $N_i' = N_i - \frac{1}{\beta}\sum_j N(w_i, w_j)$, where β is used to adjust the influence of $\sum_j N(w_i, w_j)$ in N_i'. It is easy, yet not sufficient, because only the co-occurrence redundancy in the TF term N_i is removed, and the removement is not specifically carried out for each image. Inspired by this and Co-Cosine, we propose a new co-occurrence weighting scheme Co-TFIDF to remove the co-occurrence redundancies from the TFIDF values.

We consider that (12) is actually an inner product of a query vector y and vector $x^T(I - \frac{1}{\beta}\Sigma)$, where $x^T(I - \frac{1}{\beta}\Sigma)$ is a linear transformation performed on the vector x. Such a linear transformation down-weights the TFIDF values of those visual words that commonly co-occur with a large number of words, so we utilize it as a way to subtract the co-occurrence redundancies from both query and database image vectors. Suppose the coordinative transformed vectors are $x'^T = x^T(I - \frac{1}{\beta_1}\Sigma)$, $y' = (I - \frac{1}{\beta_2}\Sigma)y$, then each element of x' and y' can be written as:

$$x_i' = x_i - \frac{1}{\beta_1}\sum_j N(w_i, w_j)x_j, y_i' = y_i - \frac{1}{\beta_2}\sum_j N(w_i, w_j)y_j, \tag{13}$$

here, x_i and x_j denote the corresponding TFIDF values of the ith and jth visual words in the database image vector and $N(w_i, w_j)$ is the co-occurring number as a weighting term. y_i and y_j are the corresponding TFIDF values of the query image vector. The weighted TFIDF x_i' and y_i' are named as Co-TFIDF. If we substitute the aforementioned N_i' into its TFIDF form $\frac{N_i'}{Z} \times log\frac{1}{D_i}$, it is actually very similar to Co-TFIDF, except that in (13), the TFIDF values (x_j and y_j) of the co-occurring words are also taken into account, which makes Co-TFIDF more sufficient and specific to remove the co-occurrence redundancy from TFIDF. β_1 and β_2 are coefficients introduced to adjust the influences of the weighting terms on the original TFIDF values for both query and database images.

Suppose $\beta_2 \rightarrow \infty$, y_i' is actually equivalent to y_i, (13) reduces Co-Cosine in (12). Therefore, Co-Cosine is a special case of Co-TFIDF. Moreover, if we set $\beta_1 = \beta_2 = 2\beta$, likewise, the co-occurrence weighting similarity measure $Sim(x', y')$ is,

$$Sim(x', y') = x'^T y' = x^T(I - \frac{1}{2\beta}\Sigma)^2 y = x^T(I - \frac{1}{\beta}\Sigma + \frac{1}{4\beta^2}\Sigma^2)y \approx x^T(I - \frac{1}{\beta}\Sigma)y. \tag{14}$$

In our experiment, as a refined weighting scheme, usually $\beta >> \sigma_{ij}$, thus $\frac{1}{4\beta^2}\Sigma^2$ can be ignored compared to the influence of $\frac{1}{\beta}\Sigma$. In this manner, (14) is approximated to

Alg. 1 Exploring Spatial Correlation for Visual Object Retrieval

```
**Input:** query image $y$, maximum number of searched nodes $m$
**Output:** ranked list
**Image Representation**
randomly sample 15% features and quantize them into visual words,
group the other features with the sampled features (words) in their neighborhoods.
**for** each feature $q_s$ **do**
    search $\upsilon(m)$ candidates through high-order predictor, $\upsilon(m) - \arg\max_{\overset{\wedge}{w}_s \in W} p(\overset{\wedge}{w_s} | w_0, ..., w_{s-1})$
    search $m - \upsilon$ candidates through PTree, $\overset{\sim}{b^*} = \underset{b \in \{1,...,B\}}{\arg\min} ||q_s - c_b||(1 + e^{(\delta(r_b - 1))})$
    obtain $\overset{*}{w}_s$ with respect to $q_s$
**end for**
**Image Ranking**
**for** each image $x$ **in** database **do**
    similarity measure $Sim(x', y')$ to query image $y$, $Sim(x', y') = \{x' | \forall y' : x'^T y'\}$
    entries in $x'$ and $y'$ are calculated by, $x'_i = x_i - \frac{1}{\beta_1} \sum_j N(w_i, w_j) x_j$; $y'_i = y_i - \frac{1}{\beta_2} \sum_j N(w_i, w_j) y_j$
**end for**
return the ranked list with top $S$ images : $S - \arg\max Sim(x', y')$.
```

(12), the proposed Co-TFIDF is consistent with Co-Cosine. Co-TFIDF is formulated from the definition of TFIDF, as we have analyzed that, visual word correlation redundancies have to be subtracted from the TFIDF values to find their real importances in retrieving one image. This is more meaningful and reasonable, and it demonstrates that Co-TFIDF apparently outperforms Co-Cosine.

Alg. 1 shows the overview of our proposed image retrieval approach based on visual word co-occurrence. In the offline process, the co-occurrence matrix is built on the image database. In the online phase, given a query image, we first obtain its BOVW representation through the proposed high-order predictor + PTree; subsequently, Co-TFIDF is evaluated to rank the images in the database according to their similarity scores to query image.

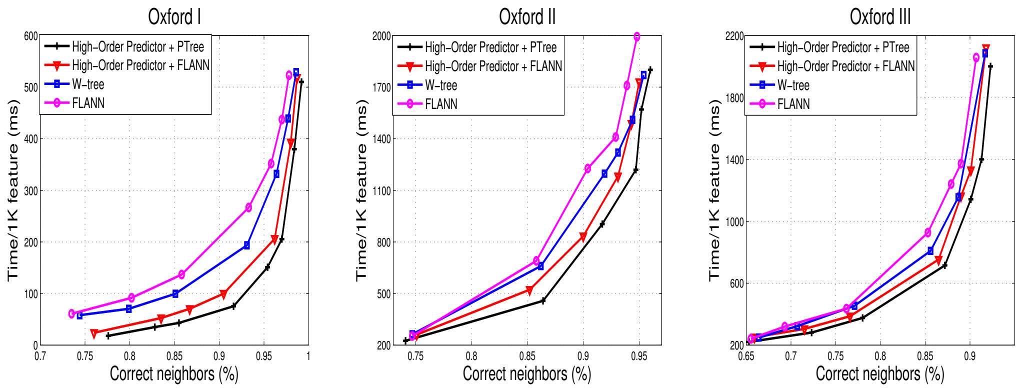
The proportion of the initial center features is set to 15%, because this setting maintains a good balance between a wide coverage of the whole image and the small time cost of exact nearest neighbor search for the center features. Random selection offers a simple yet fair treatment for every feature in an image.

5. EXPERIMENTAL RESULTS

5.1. Dataset and Evaluation

The Oxford dataset. This dataset [Philbin et al. 2007] of 5062 images is a standard image retrieval test set, which we call *Ox*. 55 images of 11 *Oxford* landmarks are selected as the query images, and their ground truth retrieval results are provided.
The ImageNet dataset. Approximately $100K$ and $500K$ images are sampled from $10M$ images in *ImageNet* [Deng et al. 2009], which we respectively call *I1* and *I2*.
The Paris dataset. This dataset contains 6390 images by querying the associated text tags for famous *Paris* landmarks [Qin et al. 2011], such as "*Paris Eiffel Tower*" or "*Paris Arc de Triomphe*".
The Holidays dataset. This dataset is a set of images which mainly contains holidays photos [Jegou et al. 2008]. We call it *Ho*. It includes a large variety of scenes (natural, manmade, water and fire, etc.).

Evaluations are first conducted on a single dataset, the *Oxford* dataset. Then, we combine the *Oxford* dataset with *ImageNet* $100K$ and $500K$ datasets, *Ox+I1*, *Ox+I2*,



(a) *Oxford* ($100K$); (b) *Oxford* ($500K$); (c) *Oxford* ($1M$);

Fig. 4. Visual word generation results on $100K$, $500K$ and $1M$ hard-assigned *Oxford* vocabularies.

to implement the proposed method on a large scale dataset. These two collections are composed of approximately $105K$ and $505K$ images. Evaluations are still carried out on the 55 *Oxford* queries because the images from $100K$ and $500K$ *ImageNet* datasets are not relevant to the 55 queries in the *Oxford* dataset. Moreover, we will test the cross dataset performances between the *Oxford* and *Paris* datasets, and the *Oxford* and *Holidays* datasets.

According to our two contributions, experimental results are given in two parts: visual word generation and image ranking. Ranking performance is measured in terms of the average precision (AP), which is defined as the area under the precision-recall curve [Philbin et al. 2007]. The AP score is computed for each query and averaged to obtain a mean average precision (mAP). Comparisons on hard- and soft-assignment, with or without bounding box, show a competitive performance of our method to the state-of-the-art [Philbin et al. 2007; 2008; Yang et al. 2010; Zhang et al. 2010; Li et al. 2011; Jegou et al. 2011a; Qin et al. 2011; Zhang et al. 2011; Shen et al. 2012; Shi et al. 2012]. Analysis of the computational complexity of our method is presented in Appendix B as well.

5.2. Visual Word Generation

The experiments are tested on the *Oxford* dataset. The entire database is split into two parts: one part contains 90% images while the other contains 10%. The features in the 90% images are used to build the co-occurrence matrix, while the features in the 10% images are used as nearest neighbor queries. Visual word vocabularies are pre-clustered by the HKM tree [Nister and Stewenius 2006]. SIFT descriptors were downloaded from the public VGG website [Philbin et al. 2007]. An average of $3,228$ local SIFT descriptors are extracted from each image in the *Oxford* dataset. We randomly split the dataset 10 times and report the average generation time and precision per $1K$ features. Precision is measured via the percentage of correct nearest neighbors in $1K$ features. Given the desired degree of precision, the best algorithm (RKD [Silpa-Anan and Hartley 2008] or HKM [Nister and Stewenius 2006]) and parameter values (including the maximum number of searched nodes) can be automatically determined in the FLANN system [Muja and Lowe 2009]. We build a PTree on the basis of FLANN, different desired degrees of precision are provided, and the corresponding average generation time of $1K$ features are illustrated in Fig. 4.

We downloaded the public FLANN code from the UBC website and ran the program on our computer, the same training and testing subsets are used in comparison with our method. As shown in Fig. 4, significant improvements of time efficiency without precision loss have been achieved by high-order predictor + PTree. Thanks to the visual word co-occurring information used in the high-order predictor, compared to the large amount of backtrackings in FLANN, fewer words that are more likely to be the nearest neighbors are provided during the prediction.

The blue line shows the performance of the W-tree [Shi et al. 2013], which optimizes the priority search in the tree. Compared to high-order predictor + PTree, it is a bit inferior. It has a better generalization ability on different sizes of datasets with different co-occurring patterns; however, too much offline processing is required to construct the weighting matrix. The construction of the visual word co-occurrence matrix in this paper is much easier.

To demonstrate the effectiveness of the high-order predictor + PTree over high-order predictor + FLANN, looking at Fig. 4, we see that, if we are willing to accept a precision as high as 95%, meaning that only 5% of the generated visual words are not the exact nearest neighbors, but just approximations, we can achieve an average of 24.0% improvement of the time efficiency. Time efficiency gain (%) is calculated by $\frac{T_2 - T_1}{T_1}$, where T_2 and T_1 are the average generation time per $1K$ features for high-order predictor + FLANN and high-order predictor + PTree at 0.95 precision. For $100K$, $500K$, and $1M$ vocabularies, the time efficiency gains are 28.6%, 23.6%, and 19.8%, respectively; when the desired degree of precision is even higher (close to 1), the computational cost of approximate nearest neighbor search will be no better than the brutal search; thereby, in the evaluation of image retrieval, the acceptable precision of visual word generation is usually set as 0.9 or 0.95 in the approximate nearest neighbor search. This paper sets it as 0.95 for the following retrieval evaluation.

On average, compared to FLANN, the performance of a high-order predictor + PTree on the $100K$ vocabulary is sort of better than those of the $500K$ and $1M$ vocabularies. With an increase of the vocabulary size, local patterns in images are specifically constructed by unique pairs of visual words. When we count the co-occurring numbers of those visual word pairs over the entire dataset, for many them, their co-occurring numbers are 1, which makes them hardly distinguished and indexed in the high-order predictor. The precision by the high-order predictor declines with an increase of the vocabulary size, correspondingly, the improvement of the proposed algorithm reduces.

In addition to its promising performance, PTree provides a semi-supervised way in a search tree. Generally, visual features are independently searched in the trees, however, they are spatially correlated in an image. By considering a small amount of exactly mapped features and their spatial correlations with other features, we could have a rough idea of the locations of those unmapped features in the tree, and PTree has succeeded in utilizing this spatial information to quickly generate the visual words. It provides us a perspective to develop a semi-supervised tree with limited knowledge of the data points, which has not been carefully exploited before. Moreover, we know that the logarithmic time complexity $O(dlog^{(O(1))}(dN))$ of a search tree could possibly be traded with $(dN)^{(O(1))}$ additional space complexity [Arya et al. 1998]. In PTree, only those visual words that have been searched in the high-order predictor need to be marked and excluded, thus, the overhead storage are indeed the visual word co-occurrence matrix (certain lists), plus few additional spaces. PTree has actually done very well in decreasing the computation of a search tree.

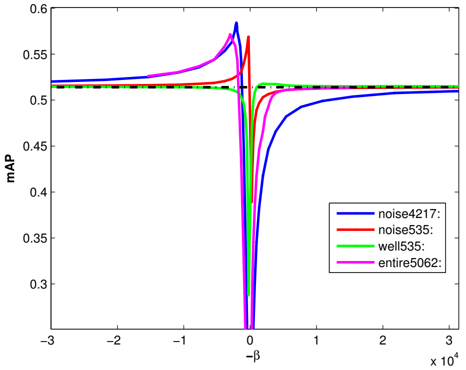


Fig. 5. The mAP values correspond to different β in Co-Cosine. Co-occurrence matrixes are constructed on different parts of the *Oxford* dataset;

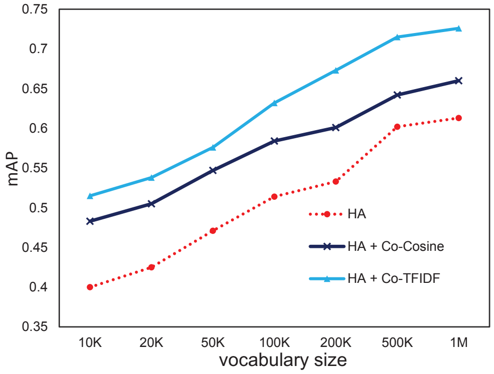


Fig. 6. Comparisons of HA and HA + Co-Sim (Co-Cosine and Co-TFIDF) with different sizes of hard-assigned vocabularies.

5.3. Image Ranking

We report the image ranking results with our proposed co-occurrence weighting similarity (we use Co-Sim to denote both Co-Cosine and Co-TFIDF) on *Oxford*, *Paris*, *ImageNet* and *Holidays* datasets. We carefully test our proposed similarity measures on multiple scenarios: parameter variation, with hard- or soft-assignment, and with or without query bounding box. Moreover, the comparison of the overall performance in different datasets, cross datasets, and large scale datasets are also carefully evaluated. The performance in each case demonstrates that our method has achieved significant improvement.

5.3.1. Co-Sim (Co-Cosine & Co-TFIDF).

Parameter variation: Fig. 5 illustrates the mAP values for different β in Co-Cosine (12). Co-occurrence matrixes are constructed on separated parts of the *Oxford* dataset. The performances prove the noise assumption in Section 4.2: all irrelevant images in the database can be taken as noise that produces negative information in image ranking, as illustrated by the blue line, the highest mAP can be achieved when β is positive ($-\beta$ is negative in (12)). Looking at the purple and blue lines in the figure, the performance of using the entire database images (entire5062) to construct the co-occurrence matrix is close to that of using the irrelevant images (noise4217). Since the number of relevant images to certain query is always very small in large scale image retrieval, the negative information from the entire database obviously overwhelms the trivial positive information. In real implementation, we can simply construct the co-occurrence matrix on the entire database without any prior knowledge of the noise, which will naturally reflect the attribute of all the irrelevant images in the database.

We build the co-occurrence matrix on the 535 *Good (OK)* images; only a slight improvement can be seen with $-\beta$ being positive, as shown in Fig. 5, the green line. This is correct because the number of relevant images to a query is small. Even in the 535 *Good (OK)* images, only a minority contributes positive information to a certain query. A comparison is carried out on the randomly selected 535 images from the noisy set, as illustrated by the red line, and a satisfactory result is still achieved. With an increase of the noisy set size, the performance will be enhanced.

We also investigate the parameter variation of Co-TFIDF. The selection of β is sort of sensitive in Fig. 5. Referring to [Shi et al. 2012], β actually indicates the normalization of each co-occurrence list, $\sum_j \sigma_{ij} = 1$. After normalizing the co-occurrence matrix Σ by

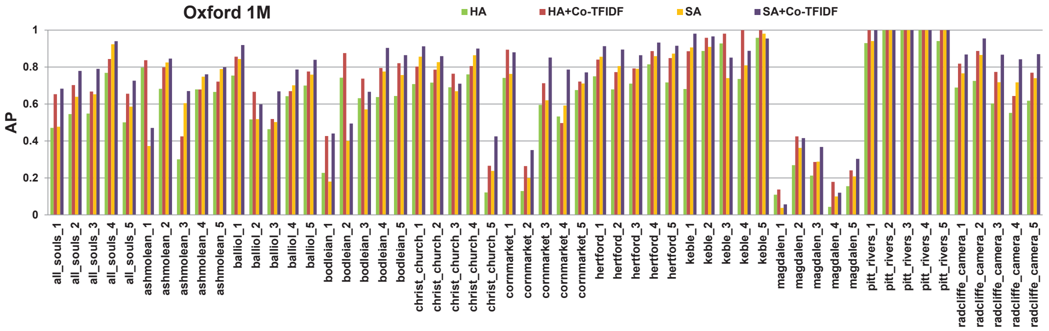


Fig. 7. The AP values of each query on HA and SA vocabularies with Co-TFIDF. The vocabulary size is 1M.

rows, optimal β is around 1.35 for datasets like *Oxford* and *Paris* that have the same capacity; β is equivalent to β_1 in Co-TFIDF, while β_2 is set to be larger than β_1, to less down-weight the TFIDF values in the query vector. In practice, we suggest that β_2 be chosen around 2 according to our experiment results.

Effect of vocabulary size. We evaluate the effectiveness of our Co-Sim on the *Oxford* dataset for different hard-assigned (HA) vocabularies, as shown in Fig. 6. The mAPs of baseline and the corresponding improvements by Co-Cosine and Co-TFIDF are shown by the red line, navy blue line, and sky blue line, respectively. The performance of Co-TFIDF is obviously superior to Co-Cosine and baseline. Significant improvement of Co-TFIDF can be found on any given vocabulary. We focus on those visual words that have a large amount of co-occurrences with other words, and reduce their weights in retrieving an image, significant improvement is thus obtained.

Hard- & soft-assigned vocabulary. The proposed Co-Sim is simply a novel similarity measure that can be embedded into any ranking technique. In Table I we test its performance with both hard- and soft-assigned (SA) vocabularies [Philbin et al. 2008] on the *Oxford* dataset. It can be seen that, the increases (Co-TFIDF) in mAP on different vocabularies are significant (23.0% and 22.9% improvements for the $100K$ HA and SA vocabularies, 18.9% and 18.4% improvements for the $1M$ vocabularies). Meanwhile, Co-TFIDF yields superior results over Co-Cosine. It increases the mAPs by 8.2% and 8.0% increments from Co-Cosine on the $100K$ HA and SA vocabularies, or 10.5% (HA) and 6.3% (SA) increments on the $1M$ vocabularies. The highest mAP can reach 0.776 using $1M$ SA vocabulary.

Fig. 7 demonstrates the AP value of each query on HA and SA vocabularies, with Co-TFIDF. The vocabulary size is $1M$.

With & without a query bounding box. In real implementation, visual bounding boxes are often manually labeled for query images to get rid of the nonsensical parts. We compare the results of our Co-Sim with and without a query bounding box. Table I shows that the new scheme without a bounding box apparently outperforms the results of hard- and soft-assignments even with bounding box. We suggest that this is because the proposed similarity does not need a manually labeled bounding box and functions intrinsically as a virtual bounding box, with the contribution of nonsensical words being smoothed down to trivia. Such a virtual bounding box effect can work better than a manually labeled one because the elimination of redundant information is carried out in the feature space rather than the image space.

We also note that if we add a real bounding box to our scheme, the performance only improves a little, and in some cases, even declines. This is due to the penalizing effect imposed on all the visual words. If we add a real bounding box, the general elimination effect of Co-Sim might not always outweigh the mistakes it makes on the

Table I. Performance of Co-Sim embedded on HA and SA vocabularies with or without bounding box (BB) on the *Oxford* dataset. These results are the corresponding mAP values on $100K$ and $1M$ vocabularies.

	BB	Co-Cosine	Co-TFIDF	*Ox* $(100K)$	*Ox* $(1M)$
HA				0.514	0.613
HA		+		0.584	0.660
HA			+	**0.632**	**0.729**
HA	+			0.514	0.613
HA	+	+		0.577	0.648
HA	+		+	**0.615**	**0.708**
SA				0.529	0.640
SA		+		0.602	0.719
SA			+	**0.650**	**0.758**
SA	+			0.554	0.673
SA	+	+		0.611	0.730
SA	+		+	**0.664***	**0.776***

Table II. Large scale retrieval results compared with state-of-the-art models. Corresponding numbers are their mAP values. Vocabulary size is $1M$.

Vocabulary	*Ox+I1*	*Ox+I2*
HA	0.566	0.499
SA	0.603	0.534
CVV	0.610	0.549
SCQE	0.616	0.574
Co-Cosine	0.630	0.615
Co-TFIDF	**0.661**	**0.630**

matched points, although in general unmatched points are the majority. By embedding the Co-TFIDF into soft-assignment with query bounding box, the final mAP for $1M$ vocabulary reaches 0.776, and for the $100K$ vocabulary it reaches 0.664.

5.3.2. Comparison of Overall Performance.

In this subsection, Co-Sim is first compared with many representative approaches on the *Oxford* dataset. The *ImageNet* dataset is then added to the *Oxford* dataset for large scale evaluation. Finally, we test cross dataset performance between the *Oxford* and *Paris* datasets, the *Oxford* and *Holidays* datasets.

Comparison with the state-of-the-art. Detailed comparisons are evaluated on *Oxford* dataset with $100K$ and $1M$ vocabularies [Philbin et al. 2007; 2008]. Representative approaches include [Yang et al. 2010; Zhang et al. 2010; Li et al. 2011; Jegou et al. 2011a; Qin et al. 2011; Zhang et al. 2011; Shi et al. 2012; Shen et al. 2012]. The same query set and ground truth have been used in all these approaches as a standard evaluation.

Yang et al. [2010] constructed a contextual model (CM) by utilizing a bounding box in a query image. Different weights calculated from the saliency map are added to the visual words inside and outside the bounding box. Zhang et al. [2010] considered the spatial correlation inside the feature's affine-invariant region: several features may co-occur in one region, and they generated these co-occurring features as one contextual visual word in the contextual visual vocabulary (CVV), and this helped to supervise the retrieval performance. Li et al. [2011] proposed a spatial co-occurrence query expansion (SCQE) method. They built a spatial co-occurrence graph from the database. Each query image is expanded with some spatially correlated but unseen visual words according to the spatial graph. The retrieval performance is improved by expanding these visual words appropriately. In [Jegou et al. 2011a; Qin et al. 2011; Zhang et al. 2011; Shen et al. 2012], like Co-Sim, they are all cosine-based or TFIDF-based similarity measures, but notwithstanding, additional techniques were employed in these methods. For instance, in [Zhang et al. 2011; Shen et al. 2012], geometric structure is exploited to constrain the similarity measure and the initial search result is reranked accordingly; by taking into account the neighborhood of the image space, e.g. k-reciprocal nearest neighbors, image dissimilarity measure is learned in [Jegou et al. 2011a; Qin et al. 2011].

Results are summarized in Table III. Most of the values are reported from their papers, except for [Zhang et al. 2010], who did not use the same datasets as we did, so we implemented it ourselves. Our method is superior to most of the existing methods, but inferior to a few [Qin et al. 2011; Shen et al. 2012]. Despite the significant improve-

Table III. mAPs for *Oxford* dataset compared to the results of state-of-the-art. Vocabulary sizes are $100K$ and $1M$.

Dataset	Co-TFIDF	[Philbin et al. 2007]	[Philbin et al. 2008]	[Yang et al. 2010]	[Zhang et al. 2010]	[Jegou et al. 2011a]	[Qin et al. 2011]	[Li et al. 2011]	[Zhang et al. 2011]	[Shen et al. 2012]	[Shi et al. 2012]
Oxford100K	**0.664**	0.514	0.554	0.545	0.565			0.596	0.622		0.611
Oxford1M	0.776	0.613	0.676	0.658	0.661	0.764	0.814	0.708	0.713	**0.884**	0.730

Table IV. Cross dataset performances of Co-Cosine and Co-TFIDF by obtaining the co-occurrence matrix from the *Paris* (*Oxford*) dataset and testing it on the *Oxford* (*Paris*) dataset. $500K$ HA vocabulary is used. The baseline column denotes baseline mAPs without using the co-occurrence matrix [Philbin et al. 2007].

Training / Testing		*Ox*	*Paris*	Baseline
Ox	Co-Cosine	0.642	0.611	0.602
	Co-TFIDF	**0.715**	**0.623**	
Paris	Co-Cosine	0.672	0.704	0.666
	Co-TFIDF	**0.681**	**0.734**	

Table V. Cross dataset performances of Co-TFIDF by obtaining the co-occurrence matrix from the *Oxford* (*Holidays*) dataset and testing it on the *Holidays* (*Oxford*) dataset. $100K$ and $1M$ HA vocabularies are used. The baseline column denotes baseline mAPs [Philbin et al. 2007].

Training / Testing		*Ox*	*Ho*	Baseline
Ho	100K	**0.676**	0.654	0.644
	1M	**0.792**	0.781	0.776
Ox	100K	**0.632**	0.533	0.514
	1M	**0.729**	0.629	0.613

ments they obtained in [Qin et al. 2011; Shen et al. 2012], the computation in these methods is of quadratic complexity in terms of the number of images [Qin et al. 2011]; the whole dataset has to be re-ranked several times for the query and its k-nearest neighbors (k-NN), which is very time consuming for online search. Indeed, the mAP reported in [Shen et al. 2012] without k-NN re-ranking is only 0.752, which is lower than our result. In comparison, our method only exploits the spatial co-occurrence of visual words in the entire dataset. We reduce the weight of the visual word in the similarity measure in terms of its co-occurrences with other words. It is simple and easy to implement. The only information we need is a visual word co-occurrence matrix, which is collected offline and mainly determined by the size of the local region used to confine the co-occurring visual words. Neither labeled data nor re-ranking techniques are required in our method. It is beneficial to plug the proposed scheme into any large-scale retrieval system, and the improvement is significant.

Large scale evaluation. Comparisons with SA [Philbin et al. 2008], Contextual Visual Vocabulary (CVV)[Zhang et al. 2010], and SCQE [Li et al. 2011] on the *Ox+I1* and *Ox+I2* datasets are given in Table II. All these approaches are implemented on HA vocabulary except for [Philbin et al. 2008]. Compared to Co-Sim, [Zhang et al. 2010] and [Li et al. 2011] exhibit inferior performance on the large scale dataset as a result of the noise introduced by *ImageNet*, in contrast, Co-Sim is robust enough to remove the irrelevant images in large scale image retrieval.

Cross dataset performance. To measure the generalization ability of Co-Sim, we evaluate the cross-dataset performance between the *Oxford* and *Paris* datasets, and the *Oxford* and *Holidays* datasets.

Table IV shows the performance by training the co-occurrence matrix from *Paris* dataset and testing it on the *Oxford* dataset, the mAP for the $500K$ vocabulary is mildly improved from 0.602 to 0.623 instead of 0.715 using its own co-occurrence matrix; on the contrary, when we obtain the co-occurrence matrix from *Oxford* dataset and test it on the *Paris* dataset, the mAP is improved from 0.666 to 0.681 compared to 0.734 using its own co-occurrence matrix; in both datasets, tree, sky, and grass can always be regarded as noise, so that Co-Sim still exhibits improvement in the cross dataset performance. Fig. 8 shows some examples of the search results.

Table V shows that it is even beneficial to apply the co-occurrence matrix collected from the building images (the natural scene images) to that of the natural scene im-

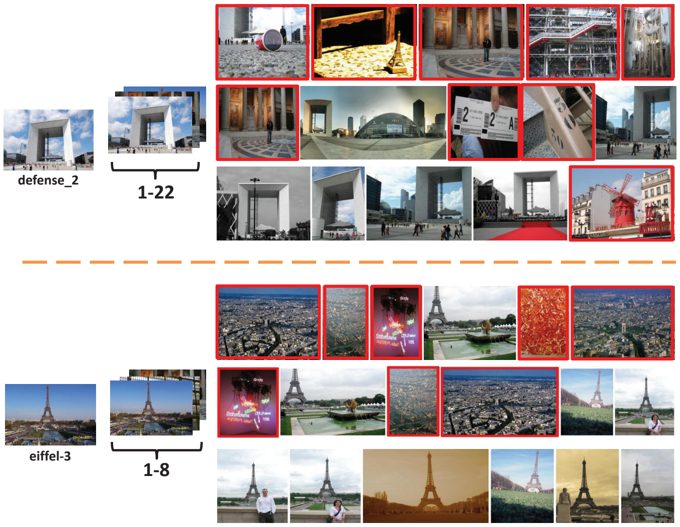


Fig. 8. An illustration of search results: two examples of query images, defense_2 (top), eiffel_3 (bottom), from the *Paris* dataset. The three rows of search results correspond respectively to hard assignment, cross-dataset (co-occurrence matrix built on *Oxford* dataset), and Co-TFIDF. False alarms are marked with red boxes.

ages (the building images). Despite their different contents in the *Oxford* and *Holidays* datasets, the basic co-occurring patterns that construct an object, or an image are similar, which is the reason we can reduce the weight of a visual word that commonly co-occurs with a large number of visual words, even if the co-occurrence matrix we utilized is built on a different dataset. Note that, the mAPs for the *Holidays* dataset using the *Oxford* co-occurrence matrix are even higher than using its own, because images in the *Holidays* dataset are all query-related, noisy information cannot be collected.

6. CONCLUSION

This paper proposes a novel image retrieval approach that explores the spatial correlation of visual words. It improves the retrieval performance by presenting two novel methods: fast word generation via candidates prediction and refined similarity measure via down-weighting. By exploring the visual word co-occurrence information, high-order predictor + PTree and Co-Sim are developed for visual word generation and image ranking, respectively.

Word generation is faster than approaches that use the conventional tree index, and the refined similarity measure is more precise than the cosine similarity measure, so that the overall retrieval performance is improved. The theoretical analysis presented in this paper also proves the effectiveness of our method. These two novel techniques can be used independently and can be embedded in most image retrieval systems. They can also be used in other applications, such as image classification, object recognition and video surveillance.

ACKNOWLEDGMENTS

This paper is supported by NBRPC 2011CB302400, NSFC 61121002, 61375026 and JCYJ 20120614152136201.

REFERENCES

S. Arya, D.M. Mount, N.S. Netanyahu, R. Silverman, and A.Y. Wu. 1998. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM (JACM)* 45, 6 (1998), 891–923.

R. Baeza-Yates and B. Ribeiro-Neto. 1999. *Modern information retrieval*. Vol. 463.

J.S. Beis and D.G. Lowe. 1997. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*. 1000–1006.

J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, and L. Fei-Fei. 2009. Imagenet: a large-scale hierarchical image database. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*. 248–255.

D. Guo, H. Xiong, V. Atluri, and N. Adam. 2009. Object discovery in high-resolution remote sensing images: a semantic perspective. *Knowledge and Information Systems* 19, 2 (2009), 211–233.

Y. Huang, S. Shekhar, and H. Xiong. 2004. Discovering co-location patterns from spatial datasets: a general approach. *IEEE Trans. Knowledge and Data Engineering* 16, 12 (2004), 1472–1485.

H. Jegou, O. Chum, and others. 2012. Negative evidences and co-occurrences in image retrieval: the benefit of PCA and whitening. In *Proc. IEEE Eur. Conf. Computer Vision*.

H. Jegou, M. Douze, and C. Schmid. 2008. Hamming embedding and weak geometric consistency for large scale image search. In *Proc. IEEE Eur. Conf. Computer Vision*. Springer, 304–317.

H. Jegou, M. Douze, and C. Schmid. 2009. On the burstiness of visual elements. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*. 1169–1176.

H. Jegou, M. Douze, and C. Schmid. 2011a. *Exploiting descriptor distances for precise image search*. Technical Report.

H. Jegou, M. Douze, and C. Schmid. 2011b. Product quantization for nearest neighbor search. *IEEE Trans. Pattern Analysis and Machine Intelligence* 33, 1 (2011), 117–128.

H. Jegou, C. Schmid, H. Harzallah, and J. Verbeek. 2010. Accurate image search using the contextual dissimilarity measure. *IEEE Trans. Pattern Analysis and Machine Intelligence* 32, 1 (2010), 2–11.

Y. Li, B. Geng, Z. Zha, Y. Li, D. Tao, and C. Xu. 2011. Query expansion by spatial co-occurrence for image retrieval. In *Proc. ACM Int'l Conf. Multimedia*. 1177–1180.

T. Liu, A.W. Moore, A. Gray, and K. Yang. 2004. An investigation of practical approximate nearest neighbor algorithms. *Advances in Neural Information Processing Systems* 17 (2004), 825–832.

D.G. Lowe. 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60, 2 (2004), 91–110.

K. Mikolajczyk and C. Schmid. 2004. Scale & affine invariant interest point detectors. *International Journal of Computer Vision* 60, 1 (2004), 63–86.

M. Muja and D.G. Lowe. 2009. Fast approximate nearest neighbors with automatic algorithm configuration. In *Proc. Int'l Conf. Computer Vision Theory and Applications*. 331–340.

D. Nister and H. Stewenius. 2006. Scalable recognition with a vocabulary tree. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Vol. 2. 2161–2168.

J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. 2007. Object retrieval with large vocabularies and fast spatial matching. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*. 1–8.

J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. 2008. Lost in quantization: improving particular object retrieval in large scale image databases. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*. 1–8.

D. Qin, S. Gammeter, L. Bossard, T. Quack, and L. Van Gool. 2011. Hello neighbor: accurate object retrieval with K-reciprocal nearest neighbors. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*. 777–784.

X. Shen, Z. Lin, J. Brandt, S. Avidan, and Y. Wu. 2012. Object retrieval and localization with spatially-constrained similarity measure and k-NN re-ranking. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*. IEEE, 3013–3020.

M. Shi, X. Sun, D. Tao, and C. Xu. 2012. Exploiting visual word co-occurrence for image retrieval. In *Proc. ACM Int'l conf. Multimedia*. 69–78.

M. Shi, R. Xu, D. Tao, and C. Xu. 2013. W-tree indexing for fast visual word generation. *IEEE Trans. Image Processing* 22, 3 (2013), 1209–1222.

C. Silpa-Anan and R. Hartley. 2008. Optimised KD-trees for fast image descriptor matching. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*. 1–8.

J. Sivic and A. Zisserman. 2003. Video google: a text retrieval approach to object matching in videos. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*. 1470–1477.

W. Tang, R. Cai, Z. Li, and L. Zhang. 2011. Contextual synonym dictionary for visual object retrieval. In *Proc. ACM Int'l Conf. Multimedia*. 503–512.

D. Tao, X. Li, and S.J Maybank. 2007. Negative samples analysis in relevance feedback. *IEEE Trans. Knowledge and Data Engineering* 19, 4 (2007), 568–580.

D. Tao, X. Li, X. Wu, and S.J Maybank. 2009. Geometric mean for subspace selection. *IEEE Trans. Pattern Analysis and Machine Intelligence* 31, 2 (2009), 260–274.

D. Tao, X. Tang, X. Li, and Y. Rui. 2006a. Direct kernel biased discriminant analysis: a new content-based image retrieval relevance feedback algorithm. *IEEE Trans. Multimedia* 8, 4 (2006), 716–727.

D. Tao, X. Tang, X. Li, and X. Wu. 2006b. Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *IEEE Trans. Pattern Analysis and Machine Intelligence* 28, 7 (2006), 1088–1099.

J.K. Uhlmann. 1991. Satisfying general proximity/similarity queries with metric trees. *Inform. Process. Lett.* 40, 4 (1991), 175–179.

J. Wang and X. Hua. 2011a. Interactive image search by color map. *ACM Trans. Intelligent Systems and Technology* 3, 1 (2011), 12.

M. Wang and X. Hua. 2011b. Active Learning in multimedia annotation and retrieval: a survey. *ACM Trans. Intelligent Systems and Technology* 2, 2 (2011), 10.

X. Wang, M. Yang, T. Cour, S. Zhu, K. Yu, and T.X. Han. 2011. Contextual weighting for vocabulary tree based image retrieval. In *Proc. Int'l conf. Computer Vision*. 209–216.

R. Xu, M. Shi, B. Geng, and C. Xu. 2011. Fast visual word quantization via spatial neighborhood boosting. In *Proc. Int'l Conf. Multimedia and Expo.* 1–6.

L. Yang, B. Geng, A. Hanjalic, and X.S. Hua. 2010. Contextual image retrieval model. In *Proc. ACM Int'l Conf. Image and Video Retrieval*. 406–413.

S. Zhang, Q. Huang, G. Hua, S. Jiang, W. Gao, and Q. Tian. 2010. Building contextual visual vocabulary for large-scale image applications. In *Proc. ACM Int'l Conf. Multimedia*. 501–510.

S. Zhang, Q. Tian, G. Hua, Q. Huang, and W. Gao. 2012. Generating descriptor visual words and visual phrases for large-scale image applications. *IEEE Trans. Image Processing* 20, 9 (2012), 2664–2677.

S. Zhang, Q. Tian, G. Hua, Q. Huang, and S. Li. 2009. Descriptive visual words and visual phrases for image applications. In *Proc. ACM Int'l Conf. Multimedia*. 75–84.

Y. Zhang, Z. Jia, and T. Chen. 2011. Image retrieval with geometry-preserving visual phrases. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*. 809–816.

Online Appendix to: Exploring Spatial Correlation for Visual Object Retrieval

MIAOJING SHI and XINGHAI SUN, Peking University
DACHENG TAO, University of Technology, Sydney
CHAO XU, Peking University
GEORGE BACIU, Hong Kong Polytechnic University
HONG LIU, Peking University

A. ANALYSIS OF CO-TFIDF AND CO-MISSING WORDS

If we assume that each visual word's co-occurrence numbers with other words are the same in the co-occurrence matrix, after we normalize the corresponding list, without considering the parameters in (13), Co-TFIDF turns out to be,

$$x'_i = x_i - \tilde{u_i}, y'_i = y_i - \tilde{v_i}, \tag{15}$$

we denote by $\tilde{u_i}, \tilde{v_i}$ the mean TFIDF values of the co-occurring visual words for x_i and y_i. Recall the similarity measure proposed in [Jegou et al. 2012], co-missing words in the bag-of-words vectors are considered, a naive subtraction of the mean bag-of-words vector has been carried out for both x and y, so that their similarity measure is discriminative for those visual words with their TFIDF values being zeros in the original bag-of-words vectors. (15) actually did the same thing except for two points: 1) for certain visual word, in [Jegou et al. 2012], the mean TFIDF value of the same word over the entire database is subtracted; while in (15), the mean TFIDF value of its co-occurring words over the entire database is subtracted. Since the TFIDF values of the frequently co-occurring visual words are correlated, it is rational to characterize the mean TFIDF value of certain word by that of its co-occurring words; 2) in [Jegou et al. 2012], the mean TFIDF value subtracted from the same visual word is the same over different images; while in (15), only those visual words in current image, their TFIDF are used for the calculation, x_j could be zero even if $N(w_i, w_j)$ is not zero, thus, the mean TFIDF values used for the same visual word are different over different images. Despite the differences, as suggested in [Jegou et al. 2012], the choice of the subtraction could be various as long as the co-missing words are considered and eliminated. Moreover, instead of using a global subtraction in [Jegou et al. 2012], a local subtraction for each image is more semantically meaningful, and superior in the ranking evaluation.

In real implementation, if we simply record the co-occurring IDs for each visual word in the co-occurrence matrix (co-occurrence numbers are all set to 1), we still achieved significant improvement. The mAP values could reach 0.602 and 0.712 for the $100K$ and $1M$ *Oxford* vocabularies, respectively. It clearly outperforms [Jegou et al. 2012], as reported in the paper, the mAP values are respective 0.539 and 0.644. Compared to (13), the implementation of (15) is faster, yet a bit inferior. In (13), TFIDF values are weighted by their co-occurring frequencies with certain words, in default, we implement (13).

B. ANALYSIS OF COMPUTATIONAL COMPLEXITY

The space complexity $O(N \bullet Len)$ of the co-occurrence matrix is mainly determined by the vocabulary size N and the average length of the co-occurrence list Len. The gen-

© 20 ACM 2157-6904/20/-ART00 $15.00
DOI: http://dx.doi.org/10.1145/0000000.0000000

erally used vocabulary size is $1M$. Len is determined by the size of the local region that we use to confine the visual word co-occurrence in neighborhood. Usually we set it to five times affine-invariant region of every visual feature. Thus, 1) the overhead storage for the *Oxford* and *Paris* datasets over different vocabularies, $100K$, $500K$, and $1M$ are around 500MB. For large-scale datasets, such as *ImageNet* dataset, storing the entire co-occurrence matrix is not wise, we suggest storing the co-occurring visual word IDs and numbers separately to save the system overhead. In addition, we could always guarantee a similar overhead storage by setting a threshold to limit the length of the co-occurrence list: only those co-occurring visual words with their co-occurring numbers larger than a threshold will be saved, otherwise, they are discarded; 2) the average computation time for the *Oxford* dataset reaches $245ms$ per query for the improved similarity measure, Co-TFIDF, rather than $28ms$ for the cosine similarity. However, this time increment is compensated by the time decrement in word generation via high-order predictor + PTree (i.e., in Fig. 4, at 0.95 precision, the time cost per image (around 3K features) drops from $1100ms$ by FLANN [Muja and Lowe 2009] to $450ms$). Notwithstanding this compensation, the time complexity could be dominated by the calculation of the image similarity measure in very large scale retrieval, in this very case, we set a threshold to limit the length of the co-occurrence list Len, it will help to accelerate the calculation of (13), meanwhile, the effectiveness will also be improved in this manner as those unstable co-occurring visual words are eliminated.