Contextual Weighting for Vocabulary Tree based Image Retrieval

Xiaoyu Wang[2], Ming Yang[1], Timothee Cour[1], Shenghuo Zhu[1], Kai Yu[1], Tony X. Han[2]
[1]NEC Laboratories America, Inc.
Cupertino, CA 95014
{myang,timothee,zsh,kyu}@sv.nec-labs.com
[2]Dept. of ECE, Univ. of Missouri
Columbia, MO 65211
{xw9x9,hantx}@missouri.edu

Abstract

In this paper we address the problem of image retrieval from millions of database images. We improve the vocabulary tree based approach by introducing contextual weighting of local features in both descriptor and spatial domains. Specifically, we propose to incorporate efficient statistics of neighbor descriptors both on the vocabulary tree and in the image spatial domain into the retrieval. These contextual cues substantially enhance the discriminative power of individual local features with very small computational overhead. We have conducted extensive experiments on benchmark datasets, i.e.*, the* UKbench*,* Holidays*, and our new* Mobile *dataset, which show that our method reaches state-of-the-art performance with much less computation. Furthermore, the proposed method demonstrates excellent scalability in terms of both retrieval accuracy and efficiency on large-scale experiments using 1.26 million images from the* ImageNet *database as distractors.*

1. Introduction

Retrieval of visually similar images from large databases has attracted tremendous research efforts in recent years. In particular, we consider a specific application scenario where query images are captured by phone cameras and database images are millions of original digital copies with a single image for each object. This scenario presents some unique challenges: the digital copies may appear quite different from their physical counterparts, especially because of lighting, reflections, motion and out-of-focus blur, not to mention significant viewpoint variations, as shown in Fig. 1. Moreover, we care mostly about the top-1 hit rate in this case, as very few candidates can be retrieved in a practical setting. This scenario is different from conventional near-duplicate image retrieval [8, 22] and web image search [17, 21, 23, 24], where both query and database images are from the same sources, either different digital versions or solely captured by phone cameras [12, 4].

This challenging task motivates us to further improve

Figure 1. Sample images in our new *Mobile* dataset: *1st* and *4th* columns (in green boxes) are database images and 2nd, 3rd, 5th, and 6th columns (in blue boxes) are queries captured by phones.

the successful large-scale image retrieval approach based on vocabulary trees [12]. Nistér and Stewénius [12] have demonstrated very inspiring retrieval performance using a large hierarchical vocabulary tree, where local features are encoded into a bag-of-words (BoW) histogram [16] with millions of visual words. Therefore, this histogram is so sparse that inverted index files are well suited to implement the indexing and searching efficiently. Visual words are weighted by the TF-IDF (term frequency-inverse document frequency) [16, 12], where the IDF reflects their discriminative abilities in database images and the TF indicates their importance in a query image. Only the feature descriptors, without the scale and orientation, are used in this method. Naturally, exploration of rich and discriminative contextual information pertinent to individual images is crucial to further boost the performance. However, the key question is how to incorporate them effectively and efficiently for retrieval on large-scale image databases.

In this paper, we explore two types of image-dependent contextual information of local features to boost their discriminative ability. Our context is defined by neighboring local features, both in terms of the vocabulary tree and in terms of the spatial image domain. The intuition is that even for identical local features their importance varies across images depending on 1) the image-specific frequencies of similar features, and 2) the compatibility of the spatial layout of adjacent features. For example, features detected in large textural regions like grass or carpets are generally not very informative for retrieval, in contrast, a small grass

2011 IEEE International Conference on Computer Vision
978-1-4577-1102-2/11/$26.00

region on a book cover or certain special patterns on carpets could be quite helpful. The TF-IDF weighting based on the entire image database cannot adequately address either of these two issues. Thus, we propose a *descriptor contextual weighting* (DCW) and a *spatial contextual weighting* (SCW) of local features. The DCW leverages the frequencies of descriptors' quantization paths on a vocabulary tree in one image, where less informative features are softly down-weighted. The SCW utilizes efficient spatial contextual statistics for each local feature, including the density of its adjacent features and their scales and orientations. These statistics are purely local and translation, scale, and rotation invariant, which preserve rich descriptive information of single local features.

The proposed contextual statistics differ from existing efforts [14, 4, 6, 18, 21, 23] utilizing geometrical relations among local features in that we do not assume global geometrical transforms or identify feature groups. A global transform between the query and database images is explicitly [14] or implicitly [4, 6] assumed when imposing strong geometrical constraints in a verification stage [14] or weak geometry constraints during retrieval [4, 6], which is restrictive. Constructing reliable high-order features [18, 21, 23] is a difficult and open task itself. Furthermore, most of these methods induce considerable computations, *e.g.*, Mahalanobis distance calculations [23] or random projections [6] for *each* feature descriptor. In contrast, our contextual statistics involve no high dimensional operations.

These two methods, descriptor and spatial contextual weighting, are complimentary to each other since they rely on different cues. Integrating them in the vocabulary tree based retrieval improves the performance remarkably with small computational overhead compared with [12]. We have conducted extensive experiments to validate our proposed approach on two benchmark datasets, *i.e.*, the *UKbench* [12] and *Holidays* datasets [4], and a new *Mobile* dataset that we collected where all queries are taken by phone cameras in different locations. We further employ the 1.26 million images in the training set of the *ImageNet* Challenge [2] as distractors to assess the large-scale retrieval performance. Our method demonstrates excellent scalability in both retrieval accuracy and efficiency in that it leads to consistent large performance gains on the large-scale database, *e.g.*, the mean average precision (mAP) improves by 10.0% and 12.5% respectively using the *UKbench* and *Holidays* datasets as queries compared with [12].

The scalability of an image retrieval algorithm can be assessed in terms of accuracy, efficiency and memory usage. The major contributions of this paper, *i.e.*, the descriptor and spatial contextual weighting schemes, focus on the accuracy and efficiency. Our approach is practical and can be easily reproduced by other researchers. We employ the training set of the *ImageNet* Challenge [2] as distractors in the large-scale experiments, which includes 1,261,392 images covering 1000 categories of objects. Several proprietary databases were used as distractors in the literature [3, 14, 1, 15, 4, 21, 5, 6, 23, 24] where the images are not publicly available. Hence, researchers cannot directly compare the performance of different large-scale experiments. Since the *ImageNet* dataset is publicly available and contains sufficient large variations, it is well suited to benchmark the retrieval accuracy, computation, and memory usage for the large-scale image retrieval task.

2. Related Work

In recent years, retrieval of visually similar images based on local invariant features [10, 11] and the BoW representation [16] has been significantly scaled up by the use of large hierarchical vocabulary trees [12]. Such trees typically contain millions of leaf nodes (representing visual words), resulting in very sparse BoW histograms. Thus, each visual word only appears in a small number of images, indexed by inverted files, making retrieval of images containing that visual word very efficient. The use of a tree structure dramatically reduces the computation time required to quantize a feature descriptor into one of millions of words, *e.g.*, for a tree with 7 layers and branch factor 10, only 7×10 inner products of 128-D SIFT descriptors are needed, instead of 10^7 for a flat structure.

Inspired by [12], several researchers have improved the retrieval accuracy of this approach from different perspectives, *e.g.*, by introducing a post spatial verification by RANSAC [14]; applying the query expansion which reissues the initial retrieval results as queries [1]; softly assigning descriptors to multiple words [15]; imposing weak geometry constraints [4]; building high-order features [18, 21, 23]; and extending to local BoW models [9]. Most of these methods induce considerable computations, *e.g.*, the spatial verification or query expansion are much slower than the retrieval, identification of *reliable* composite features is also costly. Instead, our method addresses both retrieval accuracy and efficiency. Compact feature representations [17, 13] are not addressed in this paper.

Another line of research [4, 5, 6, 7] employs a flat structure codebook with much less visual words (typically from 20K to 30K [5, 7]), which results in a coarse quantization of the descriptor space. This leads to three issues: 1) these methods are generally computationally intensive due to more computation during quantization and longer inverted image lists for each visual word; 2) the quantization error tends to be large which is addressed by thresholding a Hamming embedding distance [4, 6] to discard some descriptors or learning a context dissimilarity measure [7] of visual word vectors; and 3) many descriptors may be quantized to the same word, *i.e.*, the *burstiness* phenomenon [5], which is alleviated by down-weighting

their IDFs. The combination of these methods has achieved very good performance. The *burstiness* problem bears similar motivation as our descriptor contextual weighting. However, the fundamental difference is that the *burstiness* phenomenon rarely occurs for large vocabulary trees with millions of nodes, even though the descriptors may be similar. Thus, our DCW leverages the frequencies of node paths on a tree to re-weight the IDF. In addition, the IDF is only altered by database image *burstiness* in [5], in contrast, the DCW considers both query and database images. Our proposed approach combining DCW and SCW achieves comparable performance as [4, 5, 6, 7], but with a much smaller computational cost.

3. Proposed Approach

We first present the voting interpretation of vocabulary tree based image retrieval [12], then describe our new descriptor and spatial contextual weighting methods.

3.1. Image retrieval with vocabulary trees

Let T be a vocabulary tree obtained by hierarchical K-means clustering of local descriptors (extracted from a *separate* dataset), with a branch factor K and depth L. Each node v^{ℓ,h_ℓ} (v for short) in the tree represents a visual word, where ℓ indicates its layer and h_ℓ is the node index at that layer. We do not discern tree nodes and visual words in the following descriptions.

A query image q is represented by a bag I_q of local descriptors $\{\mathbf{x}_i\}_{i\in I_q}$, in our case $\mathbf{x}_i \in \mathbb{R}^D$ represents SIFT descriptors of dimension $D = 128$. Each $\mathbf{x}_i$ is mapped to a path of visual words from the root to a leaf of T, resulting in the quantization $T(\mathbf{x}_i) = \{v_i^{\ell,h_\ell}\}_{\ell=1}^{L_i}$. Note $L_i \leq L$ since the tree may be incomplete. Thus, a query image is eventually represented by the set of node paths obtained from its descriptors, *i.e.*, $\{T(\mathbf{x}_i)\}_{i\in I_q}$.

The database images are denoted by $\{d^m\}_{m=1}^M$, and we will omit superscript m when the context is clear. Following the same hierarchical quantization procedure, the local descriptors $\mathbf{y}_j$ in d are mapped to the collection of node paths $\{T(\mathbf{y}_j)\}_{j\in I_d}$. The similarity score $sim(q,d)$ between query q and database image d is specified by the average matching score among all pairs of descriptors or node paths [12]:

$$sim(q,d) \doteq \frac{1}{|I_q||I_d|} \sum_{i\in I_q, j\in I_d} f(\mathbf{x}_i, \mathbf{y}_j), \tag{1}$$

where the matching function f of two descriptors can be further expressed by a matching function f_v on tree nodes:

$$f(\mathbf{x}_i, \mathbf{y}_j) \doteq f_T(T(\mathbf{x}_i), T(\mathbf{y}_j)) \doteq \sum_{v_i\in T(\mathbf{x}_i), v_j\in T(\mathbf{y}_j)} f_v(v_i, v_j). \tag{2}$$

In [12], f_v is defined via a weighting function $w(v)$ over visual words v:

$$f_v(v_i, v_j) = w(v_i)\mathbb{1}(v_i = v_j), \tag{3}$$

where $\mathbb{1}(\cdot)$ is the indicator function. The authors [12] use the following expression:

$$w(v) = idf(v) = \log\left(\frac{M}{M_v}\right), \tag{4}$$

where M is the total number of database images and M_v is the number of images containing at least one descriptor that quantizes to the node v; it is computed recursively for non-leaf nodes. Note, multiple descriptors quantized to the same visual word v in the query image will contribute $w(v)$ multiple times to the matching score, which is equivalent to the term frequency, TF.

Usually, the number of descriptors in a query is up to several thousands, so the accumulation of matching scores in Eq. (1) is akin to a voting procedure for the most similar images in the database. The images d with highest similarity $sim(q,d)$ are returned as the retrieval set.

Since the vocabulary tree is very large, the number of images whose descriptors are quantized to a particular node is rather small. Therefore, inverted index files attached to leaf nodes allow a very efficient implementation of this voting procedure. Due to efficiency concerns, only deeper layer nodes are used in Eq. (2), using a stop level or stop list [12]. Using the leaf nodes only in Eq. (2) yields fastest retrieval speed, but usually with limited accuracy. The storage of inverted index files is proportional to the total number of descriptors in the database, *i.e.*, $\sum_{m=1}^M |I_{d^m}|$.

3.2. Descriptor contextual weighting

The discriminative power of descriptors, even the identical ones, varies in different images. Descriptors detected on large textural regions, *e.g.*, carpets, grass, and soils, are less informative, although their numbers could dominate in an image. In the TF–IDF weighting, their IDFs, which are not necessarily small when calculated from a large image database, contribute to the matching score many times via their TFs. This over-counting may lead to noisy retrieval results. Such descriptors should be down-weighted to avoid this. Different from a flat codebook, for a large tree T, these descriptors fall into a number of leaf nodes or sub-trees, so the *burstiness* of visual words [5] seldom happens with leaf nodes, and penalizing the IDF according to visual word counts in the database [5] is not effective with large trees.

Motivated by this observation, we propose to incorporate inverse weighted counts of a node path as descriptor contextual weights in addition to the IDF weight. Suppose descriptor $\mathbf{x}_i$ in q and $\mathbf{y}_j$ in d are quantized to the same node $v \in T(\mathbf{x}_i) \cap T(\mathbf{y}_j)$, with the knowledge of q and d we

modify the weighting function $w(v)$ in Eq. (4) to:

$$w_{i,j}^{q,d}(v) = w_i^q \; w_j^d \; idf(v). \tag{5}$$

Denoting $n^q(v)$ as the number of descriptors in image q that are quantized to v, we define the DCW term w_i^q based on the node counts along the quantization path of $\mathbf{x}_i$:

$$w_i^q = \sqrt{\frac{\sum_{v \in T(\mathbf{x}_i)} \omega(v)}{\sum_{v \in T(\mathbf{x}_i)} \omega(v) \times n^q(v)}}, \tag{6}$$

where $\omega(v)$ is a weighting coefficient, set to $idf(v)$ empirically. Note the subtlety that the weight w_i^q depends on the descriptor only, and is shared for all nodes v along the path $T(\mathbf{x}_i)$. As shown in Fig. 2, if two descriptors $\mathbf{x}_i$ and $\mathbf{x}_{i'}$ only differ at the leaf node, their common parent node will have different weights $w_{i,j}^{q,d}(v)$ and $w_{i',j}^{q,d}(v)$ in Eq. (5).

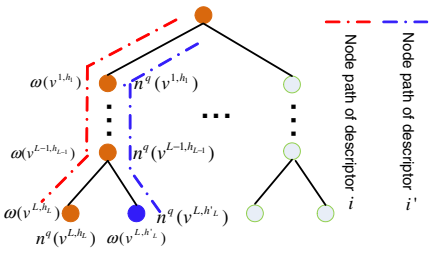

Figure 2. Illustration of the descriptor contextual weighting.

The choice of using inverse weighted counts of a *node path* in Eq. (6) is justified by the characteristics of vocabulary trees. Since the tree is deep, the majority of leaf nodes (over 95% in the experiments) only have one descriptor being quantized to it for one image. In view of this, Eq. (6) actually takes into account the descriptors quantized to neighbor tree nodes to determine the importance of a descriptor in retrieval, where nodes in a sub-tree with more descriptors is softly down-weighted compared to a sub-tree with fewer descriptors. The square root in Eq. (6) is due to the weighting of both query and database images. In practice, the descriptor contextual weights are mainly determined by the tree nodes in deeper layers. Fig. 3 shows the impacts of DCW, where green discs indicate the original IDF weights and red discs indicate the DCW weights, with the radius proportional to their strengths. Small red discs indicate less discriminative descriptors are being heavily down-weighted according to DCW for a particular image.

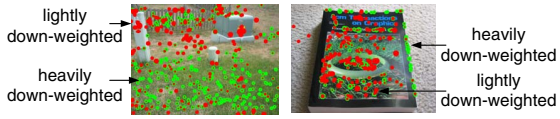

Figure 3. Impact of DCW: small red discs represent features that are heavily down-weighted (up to 400 features are drawn).

The DCW dramatically enhances the discriminative ability of local descriptors as shown in the experiments. Furthermore, it has a negligible computation overhead. The only cost is the additional storage of w_j^d for each descriptor $\mathbf{y}_j$ of database images in inverted index files, which is one additional byte per descriptor in our implementation.

3.3. Spatial contextual weighting

Local descriptors are not independent and their neighborhoods contain much rich information. As shown in Fig. 4(a), descriptors on clubs are all similar and unable to distinguish *club A* or *club 8*, unless we explore their neighborhoods. Nevertheless, in general it is costly to exploit high order information of local descriptors. We propose to employ simple statistics in the local neighborhood of an invariant feature as its spatial context to enhance its discriminative ability.

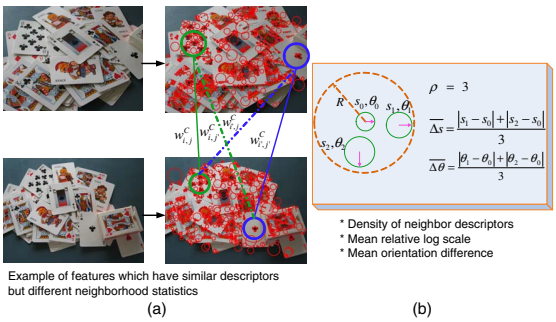

Figure 4. Illustration of the spatial contextual weighting.

A SIFT feature $\mathbf{f}_0 = \{\mathbf{x}_0, \mathbf{u}_0, s_0, \theta_0\}$ includes the descriptor $\mathbf{x}_0 \in \mathbb{R}^D$, location $\mathbf{u}_0$, characteristic scale s_0 (in the log domain), and orientation θ_0. Let $C(\mathbf{f}_0)$ denote the neighborhood of this feature given by the disc $(\mathbf{u}_0, R)$. Empirically we set the radius $R = 12 \times 2^{s_0}$ (maximum 150 pixels). We calculate 3 statistics of $C(\mathbf{f}_0)$, *i.e.*, the descriptor density ρ, the mean relative log scale $\overline{\Delta s}$, and the mean orientation difference $\overline{\Delta\theta}$, *w.r.t* $\mathbf{f}_0$, defined as

$$\rho = |C(\mathbf{f}_0)|, \tag{7}$$

$$\overline{\Delta s} = \frac{1}{|C(\mathbf{f}_0)|} \sum_{\mathbf{f} \in C(\mathbf{f}_0)} |s - s_0|, \tag{8}$$

$$\overline{\Delta\theta} = \frac{1}{|C(\mathbf{f}_0)|} \sum_{\mathbf{f} \in C(\mathbf{f}_0)} |\theta - \theta_0|, \tag{9}$$

where $|C(\mathbf{f}_0)|$ is the number of descriptors within $(\mathbf{u}_0, R)$. These statistics are translation, scale and rotation invariant.

Given two descriptors quantized to the same tree node, we measure the consistency of their local neighborhoods and add a spatial context term $w_{i,j}^C$ in the matching score. The matching for each statistic, in the range of $[0, 1]$, is

defined as follows

$$w^{\rho}_{i,j} = \frac{\min(\rho_i, \rho_j)}{\max(\rho_i, \rho_j)}, \quad (10)$$

$$w^{s}_{i,j} = \frac{\min(\overline{\Delta s}_i, \overline{\Delta s}_j)}{\max(\overline{\Delta s}_i, \overline{\Delta s}_j)}, \quad (11)$$

$$w^{\theta}_{i,j} = \frac{\min(\overline{\Delta \theta}_i, \overline{\Delta \theta}_j)}{\max(\overline{\Delta \theta}_i, \overline{\Delta \theta}_j)}. \quad (12)$$

Thus, the matching score of the spatial context is given by

$$w^{C}_{i,j} = w^{\rho}_{i,j}\ w^{s}_{i,j}\ w^{\theta}_{i,j}. \quad (13)$$

Plug it into Eq. (5), the final matching function is updated to

$$w^{q,d}_{i,j}(v) = w^{C}_{i,j} w^{q}_{i}\ w^{d}_{j}\ idf(v). \quad (14)$$

These simple contextual statistics effectively enhance the descriptive ability of individual features with a small computational overhead. It avoids explicit identification of reliable feature groups as required in [23]. In addition, these statistics are purely local. We do not implicitly assume a global geometrical transform exists among all descriptors, contrary to the weak geometric constraints used in [4, 6]. The search of neighbor features $\mathbf{f} \in C(\mathbf{f}_0)$ is shared by the 3 statistics and can be computed efficiently. In the inverted index files, we need to store ρ, $\overline{\Delta s}$, and $\overline{\Delta \theta}$ for each descriptor in the database. We quantize each of them to one byte, so there are 3 additional bytes per descriptor.

4. Time and Memory Complexity

In this section, we briefly study the computational complexity and memory requirement of the proposed image retrieval algorithm.

For one query image, besides SIFT feature detection, our retrieval approach involves tree-based quantization, context statistics calculation, followed by matching score calculation for the selected database image descriptors, and finally sorting of the retrieved candidate images d according to $sim(q, d)$. In terms of scalability, there are two different kinds of computations: independent or proportional to the database size. The descriptor quantization and calculation of spatial contexts are determined by the number of descriptors in the query image. The tree quantization involves KL inner products of D dimensional vectors per descriptor. Search of the neighborhood $C(\mathbf{f})$ of a feature $\mathbf{f}$ can be implemented by a branch-and-bound method on the image coordinate $\mathbf{u}$. In the large-scale case, the matching score calculation in Eq.(14), which is related to the number of images returned by the inverted lists, consumes most of the running time. The matching scores of ρ, $\overline{\Delta s}$, and $\overline{\Delta \theta}$ are obtained by look-up tables in our implementation. Overall, besides the descriptor quantization, the proposed algorithm does not induce any extra high dimensional operations, such as matrix multiplication in a Mahalanobis distance or random projections of descriptors, therefore the computational overhead compared with [12] is very low.

During retrieval, all the information about the database images is accessed from the inverted index files. We use 8 bytes per database image descriptor: one 4-byte integer for the image index [1], 1 byte for quantized descriptor contextual weight w^{d}_{j}, and 3 bytes for quantized ρ, $\overline{\Delta s}$, and $\overline{\Delta \theta}$. This doubles the memory for the inverted indexes compared to [12]. To optimize the memory usage, we can use compact data structures [6] or employ lossless compression for descriptors linked to the same leaf node, since they are always consumed at the same time during retrieval. Note, if all inverted index files are not loaded to memory, the disk access will slow down the process by orders of magnitude.

5. Experiments

After explaining the datasets and evaluation criteria, we first evaluate the retrieval performance of the descriptor and spatial contextual weighting using vocabulary trees with different branch factors and depths, then present the large-scale image retrieval experiments with 1.26 million images, and compare with the state-of-the-art methods.

The common settings for all experiments are summarized here. All the images are resized to no larger than 640×640. Up to 2500 SIFT features are detected for each image using the VLFeat library [19]. The program is implemented in C++ with moderate code optimization. All timings are based on a *single* core of an 8-core Xeon 2.44GHz blade server with 32G RAM.

5.1. Datasets

In the following, we list the datasets in the experiments and the corresponding evaluation criteria.

UKbench was collected by [12], and includes 2550 different objects or scenes. Each one has four images taken from different viewpoints. All the 10200 images are both indexed as database images and used as queries. The retrieval performance is measured by $4\times$ recall at the first four returned images, referred as N-S score [12, 7] (maximum is 4), and the mean average precision (mAP).

Holidays contains 1491 personal holiday photos undergoing various transformations; it was first used in [4]. There are 500 image groups where the first image of each group is the query. The performance is measured by mAP in a leave-one-out fashion.

Mobile includes images of 300 objects, *e.g.*, movie posters, book and magazine covers [2]. The 2000 query images were captured by an iPhone and an Android phone. The database indexes 300 images of the digital copies

[1] We assume $M \leq 2^{32}$. For our experiments, the image indexes can be saved with 3 bytes, but we prefer 4-byte aligned memory access.

[2] The *Mobile* dataset is available at http://vision.ece.missouri.edu/~wxy/

Dataset	# images	# labels	# queries	# descriptors
UKbench	10,200	2,550	10,200	25,483,644
Holidays	1,491	500	500	3,726,796
Mobile	2,300	300	2000	5,864,285
ImageNet-T	1,261,392	N/A	N/A	2,753,233,269
ImageNet-V	50,000	N/A	N/A	109,509,968

Table 1. Statistics of the image datasets in the experiments.

Tree	K	L	# leaves	# nodes
$T-8^6$	8	6	257,886	295,112
$T-8^7$	8	7	1,531,063	1,800,014
$T-10^6$	10	6	891,153	999,161
$T-10^7$	10	7	2,785,337	3,429,762

Table 2. Vocabulary trees with different parameters. Note, # leaves are smaller than the ideal case when the tree is not complete.

downloaded from internet blended by the 10200 images from the *UKbench* dataset as distractors. We evaluate the top-1 (τ_1) and top-10 (τ_{10}) hit rates on this dataset.

ImageNet is an image database crawled from internet according to the WordNet hierarchy [2]. The *ImageNet* Large Scale Visual Recognition Challenge 2010 (ILSVRC2010) clearly specifies 1.26M images of 1000 categories as the training set (denoted by *ImageNet-T*) and 50K images as the validation set (denoted by *ImageNet-V*).

We employ the *ImageNet-T* as distractor images to perform the large-scale experiment. *ImageNet-V* is used to train all the vocabulary trees in the paper. Note, *ImageNet-V* has no overlap with the *UKbench*, *Holidays*, *Mobile*, and *ImageNet-T* datasets. These datasets cover large varieties of images, *e.g.*, indoor/outdoor scenes and objects. The statistics are summarized in Table 1.

5.2. Retrieval performance

We study the retrieval performance with different vocabulary trees, which are trained on the independent *ImageNet-V* dataset by performing hierarchical K-means on 8 million randomly sampled SIFT descriptors. The parameters of trees are shown in Table 2. The baseline method [12] using the leaf nodes only is denoted by *B1* and the method that uses tree nodes up to 3 layers is denoted by *B3*. We set a stop ratio $r = 0.015$ to limit the maximum number of database images attached to a non-leaf node, *i.e.*, we exclude the node if the length of its inverted list exceeds rM. Our methods that extend *B3* are denoted by *B3+DCW*, *B3+SCW*, and *B3+DCW+SCW*. We mainly compare our method against *B3*, while *B1* is present to show the possibly fastest retrieval speed.

As shown in Table 3, 4 and 5, the proposed method demonstrates consistent improvement on the *UKbench*, *Holidays*, and *Mobile* datasets and is insensitive to vocabulary tree parameters. Using the largest tree $T-10^7$ as an example, compared with B3 the proposed method improves the N-S score from 3.38 to 3.56 while the mAP jumps from 87.76% to 91.70% on the *UKbench* dataset. The mAP on the *Holidays* dataset is improved from 71.71% to 78.05%. On the *Mobile* dataset, the top-1 hit rate jumps from 68.80% for B3 to 83.80% [3]. These tables also report the average retrieval time t_r per query, which includes descriptor quantization, context statistics matching, calculation of all the weights and matching scores, and sorting of candidate images. It does not include SIFT feature extraction, which could be done efficiently on the GPU. We observe almost no overhead results from incorporating DCW, and overall t_r increases by 30ms over the baseline B1.

5.3. Large-scale experiments

The common practice [12, 3, 14, 1, 4, 15, 21, 6, 23] to evaluate large-scale image retrieval performance is to employ a large image database as distractors included in the retrieval database. We follow the same scheme and employ different numbers of images from *ImageNet-T* as distractors, *i.e.*, the first 120K, 265K, 590K images and the entire set of 1.26M images. These large-scale experiments are extremely time-consuming. With the help of the Hadoop [20] distributed computing framework, we have extracted the SIFT features and generated inverted index files in 10 and 4 hours, respectively, using 20 blade servers.

The contextual weighting demonstrates excellent generalization capability and scalability on large-scale datasets as shown in Table 6 and Fig. 5. First, the retrieval performance degrades gracefully *w.r.t* increasing numbers of database images, *e.g.*, the N-S score only drops from 3.56 to 3.30 for 1.26M database images. Second, on all three test datasets, the performance gains over *B3* keep increasing along with the database size. Especially for the *Mobile* dataset, the top-1 accuracy improves by 23.1%, that is indeed significant for our application scenario. Third, the retrieval time scales up very well to merely 676ms for 1.26M images. These validate the excellent scalability of the proposed method. The memory usage of the inverted index files increases *w.r.t* the database size. For 1.26M images, the baseline requires 10.4Gbytes using 4 bytes for image indexes, while our method requires 20.8Gbytes.

We have investigated how to compare large-scale retrieval performance with recent methods on both accuracy and efficiency, and found out a fair comparison is not straightforward since different private image collections were used. This motivated us to use instead the public *ImageNet-T* dataset. We compare the retrieval accuracy using the standard settings on the *UKbench* and *Holidays* datasets, and report t_r per query on at least 1M images, as summarized in Table 7 [4]. Note, t_r depends on many

[3] *Oxford5K* [14, 1, 15] is another popular dataset, our method (B3+DCW+SCW) improves the mAP of B3 from 60.1% to 68.6%.

[4] The result of [21] is reported according to the comparison in [23].

UKbench	$T-8^6$		$T-8^7$		$T-10^6$		$T-10^7$		
	N-S	mAP(%)	N-S	mAP(%)	N-S	mAP(%)	N-S	mAP(%)	t_r(ms)
B1	3.18	83.54	3.06	80.40	3.11	81.69	3.08	80.63	**50**
B3	3.23	84.59	3.23	84.68	3.24	84.86	3.38	87.76	54 (+ 4)
B3+SCW	3.33	86.84	3.32	86.52	3.33	86.79	3.43	89.04	78 (+28)
B3+DCW	3.26	84.99	3.37	87.73	3.36	87.43	3.46	89.55	55 (+ 5)
B3+DCW+SCW	**3.41**	**88.32**	**3.47**	**89.93**	**3.46**	**89.66**	**3.56**	**91.70**	80 (+30)

Table 3. Image retrieval performance on the *UKbench* dataset using different vocabulary trees.

Holidays	$T-8^6$	$T-8^7$	$T-10^6$	$T-10^7$	
	mAP(%)	mAP(%)	mAP(%)	mAP(%)	t_r(ms)
B1	66.07	62.91	64.98	61.21	**50**
B3	66.05	66.45	69.73	71.71	51 (+ 1)
B3+SCW	**70.20**	70.18	72.30	74.88	75 (+25)
B3+DCW	68.73	68.90	70.35	74.55	52 (+ 2)
B3+DCW+SCW	69.96	**73.37**	**74.41**	**78.05**	76 (+26)

Table 4. Image retrieval performance on the *Holidays* dataset using different vocabulary trees.

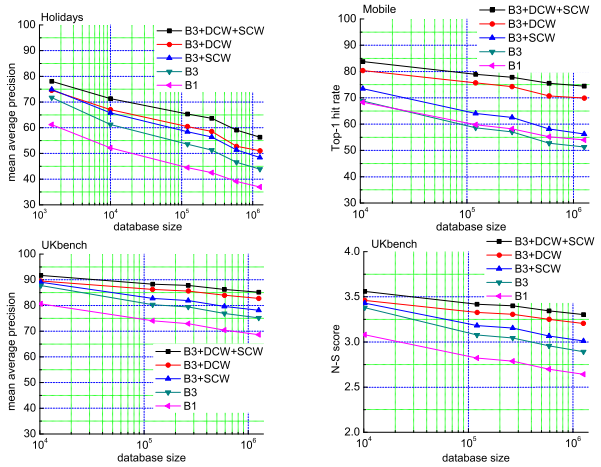

Figure 5. Large-scale image retrieval performance with different numbers of distractor images from the *ImageNet-T*.

factors (*e.g.*, a quad core was used in [6]), which are not directly comparable, so we list them as a rough indication of efficiency. It is worthwhile to clarify the comparison conditions. We only compare with the methods where 1) the K-means clustering is carried on an independent set, 2) no re-ranking is used, and 3) the query images are not somewhat used in a learning procedure to determine parameters of the retrieval system. As reported in previous works [12, 6, 7], the retrieval performance could be improved largely if query or database images are used to obtain vocabulary trees. Learning the feature dissimilarity on the database, which contains the queries, achieves N-S score 3.61 in [7], which may not scale to large-scale datasets [5]. These comparisons show that our method reaches the state-of-the-art performance on the retrieval accuracy with less computation.

	UKbench		*Holidays*	≥1M
Method	N-S	mAP(%)	mAP (%)	t_r (s)
Nistér et al. [12]	3.19	76.0	n/a	1
Wu et al. [21]	3.15	n/a	n/a	1.9
Zhang et al. [23]	3.19	n/a	n/a	3
Lin et al. [9]	3.29	n/a	n/a	n/a
Jégou et al. [6]	3.42	87.80	81.3	4.2
Jégou et al. [5]	3.54	90.70	**83.9**	6.2
Proposed	**3.56**	**91.70**	78.0	**0.676**

Table 7. Comparison with recent methods.

5.4. Discussions

We show sample failure cases of our method in Fig. 6, from which we observe that lighting, especially severe reflections, motion blur, and clutter are the major issues. In particular, for the *Mobile* dataset, the original digital copy may appear quite distinct from the queries captured by phone cameras and the database images do not contain multiple near-duplicate copies of an image. Additional retrieval results are in the **supplemental materials**. We only employ SIFT features in the retrieval. Some failure cases may thus be resolved by combining with other cues such as GIST [17] or color features. Post processing such as spatial verification or re-ranking [14] and the query expansion [1] can be readily combined with our algorithm.

6. Conclusions

In this paper, we propose two new weighting schemes to improve the vocabulary tree based image retrieval, which consider contextual information of a local feature in both descriptor and spatial domain. These approaches are very efficient and demonstrate excellent scalability for large-scale databases. They can be easily implemented by other researchers to reproduce the results. We have conducted the large-scale experiments using the *ImageNet-T* dataset, thus,

Mobile	$T-8^6$		$T-8^7$		$T-10^6$		$T-10^7$		
	τ_1 (%)	τ_{10} (%)	τ_1 (%)	τ_{10} (%)	τ_1 (%)	τ_{10} (%)	τ_1 (%)	τ_{10} (%)	t_r(ms)
B1	64.35	74.25	63.05	72.60	64.40	73.15	68.39	77.15	**50**
B3	63.30	73.35	62.90	73.25	64.10	74.70	68.80	78.00	54 (+ 4)
B3+SCW	68.20	78.10	68.20	78.10	69.40	78.85	73.50	81.85	84 (+34)
B3+DCW	68.30	76.95	73.65	82.25	73.70	82.10	80.40	86.65	55 (+ 5)
B3+DCW+SCW	**74.70**	**82.75**	**79.50**	**86.50**	**79.25**	**85.90**	**83.80**	**89.55**	85 (+35)

Table 5. Image retrieval performance on the *Mobile* dataset using different vocabulary trees.

Datasets	*UKbench*						*Holidays*		*Mobile*	
Metrics	N-S		mAP(%)		t_r (ms)		mAP(%)		τ_1 (%)	
ImageNet-T	B3	Proposed	B3	Proposed	B3	Proposed	B3	Proposed	B3	Proposed
0	3.38	3.56 (+0.18)	87.76	91.70 (+3.94)	54	80 (+ 26)	71.71	78.05 (+ 6.3)	68.80	83.80 (+15.0)
120K	3.08	3.42 (+0.34)	80.32	88.28 (+7.96)	87	115 (+ 28)	53.57	65.33 (+11.7)	58.60	78.95 (+20.3)
265K	3.04	3.40 (+0.36)	79.40	87.79 (+8.39)	115	152 (+ 37)	51.30	63.67 (+12.3)	57.05	77.80 (+20.8)
590K	2.96	3.34 (+0.38)	76.88	86.24 (+9.36)	175	276 (+101)	46.55	59.08 (+12.4)	52.75	75.55 (+22.8)
1.26M	2.89	**3.30 (+0.41)**	75.17	**85.17 (+10.0)**	**311**	676 (+365)	43.87	**56.32 (+12.5)**	51.30	**74.45 (+23.1)**

Table 6. Comparison of large-scale retrieval performance of B3+DCW+SCW (Proposed) against B3 with increasing numbers of distractors.

Figure 6. Sample failures in the *Mobile* datasets. The first image in each row is the query (in blue boxes) followed by the top-5 retrieved images (in green boxes), and the ground truth images are surrounded by red boxes.

all the images in this paper are publicly available, which can serve as a common benchmark for further large-scale image retrieval studies.

Acknowledgement

This work was done during the internship of the first author at NEC Laboratories America in Cupertino, CA.

References

[1] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *ICCV*, Oct. 14-17, 2007.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, June 20-26, 2009.

[3] F. Fraundorfer, H. Stewénius, and D. Nistér. A binning scheme for fast hard drive based image search. In *CVPR*, June 17-22, 2007.

[4] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, Oct. 12-18, 2008.

[5] H. Jégou, M. Douze, and C. Schmid. On the burstiness of visual elements. In *CVPR*, June 20-25, 2009.

[6] H. Jégou, M. Douze, and C. Schmid. Improving bag-of-feature for large scale image search. *Int'l Journal of Computer Vision*, 87(3):316–336, 2010.

[7] H. Jégou, C. Schmid, H. Harzallah, and J. Verbeek. Accurate image search using the contextual dissimilarity measure. *IEEE Trans. Pattern Anal. Machine Intell.*, 32(1):2–11, Jan. 2010.

[8] Y. Ke, R. Sukthankar, and L. Huston. Efficient near-duplicated detection and sub-image retrieval. In *ACM Multimedia*, 2004.

[9] Z. Lin and J. Brandt. A local bag-of-features model for large-scale object retrieval. In *ECCV*, Sept. 5-11, 2010.

[10] T. Lindeberg. Feature detection with automatic scale selection. *Int'l Journal of Computer Vision*, 30(2):77–116, 1998.

[11] D. G. Lowe. Distinctive image features from scale invariant keypoints. *Int'l Journal of Computer Vision*, 60(2):91–110, 2004.

[12] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *CVPR*, volume 2, June 17-22, 2006.

[13] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier. Large-scale image retrieval with compressed fisher vectors. In *CVPR*, June 13-18, 2010.

[14] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, June 17-22, 2007.

[15] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization:improving particular object retrieval in large scale image databases. In *CVPR*, June 23-28, 2008.

[16] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV*, volume 2, Oct. 13-16, 2003.

[17] A. Torralba, R. Fergus, and Y. Weiss. Small codes and large image databases for recognition. In *CVPR*, June 23-28, 2008.

[18] L. Torresani, M. Szummer, and A. Fitzgibbon. Learning query-dependent prefilters for scalable image retrieval. In *CVPR*, 2009.

[19] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008.

[20] T. White. *Hadoop: The Definitive Guide*. OReilly Media, 2010.

[21] Z. Wu, Q. Ke, M. Isard, and J. Sun. Bundling feature for large scale partial-duplicated web image search. In *CVPR*, June 20-26, 2009.

[22] D. Xu, T.-J. Cham, S. Yan, and S.-F. Chang. Near duplicate image identification with patially aligned pyramid matching. In *CVPR*, June 23-28, 2008.

[23] S. Zhang, Q. Huang, G. Hua, S. Jiang, W. Gao, and Q. Tian. Building contextual visual vocabulary for large-scale image applications. In *ACM Multimedia*, Oct. 25-29, 2010.

[24] W. Zhou, Y. Lu, H. Li, Y. Song, and Q. Tian. Spatial coding for large scale partial-duplicate web image search. In *ACM Multimedia*, Oct. 25-29, 2010.