Semantic-aware Co-indexing for Image Retrieval

Shiliang Zhang[2], Ming Yang[1], Xiaoyu Wang[1], Yuanqing Lin[1], Qi Tian[2]
[1]NEC Laboratories America, Inc. Cupertino, CA 95014
[2]Dept. of CS, Univ. of Texas at San Antonio San Antonio, TX 78249
{myang,xwang,ylin}@nec-labs.com slzhang.jdl@gmail.com qitian@cs.utsa.edu

Abstract

Inverted indexes in image retrieval not only allow fast access to database images but also summarize all knowledge about the database, so that their discriminative capacity largely determines the retrieval performance. In this paper, for vocabulary tree based image retrieval, we propose a semantic-aware co-indexing algorithm to jointly embed two strong cues into the inverted indexes: 1) local invariant features that are robust to delineate low-level image contents, and 2) semantic attributes from large-scale object recognition that may reveal image semantic meanings. For an initial set of inverted indexes of local features, we utilize 1000 semantic attributes to filter out isolated images and insert semantically similar images to the initial set. Encoding these two distinct cues together effectively enhances the discriminative capability of inverted indexes. Such co-indexing operations are totally off-line and introduce small computation overhead to online query cause only local features but no semantic attributes are used for query. Experiments and comparisons with recent retrieval methods on 3 datasets, i.e.*,* UKbench*,* Holidays*,* Oxford5K*, and 1.3 million images from* Flickr *as distractors, manifest the competitive performance of our method* [1].

1. Introduction

As the well-known saying goes, "A picture is worth a thousand words", images generally convey large amount of information. This leads to one fundamental challenge to content-based image retrieval: retrieval algorithms have no clue which subset of the "thousand words" in a query that a user is searching for. For instance, the query in Fig. 1 shows a rocky coast, then is the user searching for the exact location, rocks of similar shapes, or any coastal scene?

There are three major lines of image retrieval algorithms. All of them have to resort to various assumptions and search criteria to find candidate images, *i.e.*, searching for exact or near-duplicate images [14] by identifying similar local features [13]; finding similar images [20] by comparing hashing codes [21] of global features like GIST [15]; or retrieving objects of the same category by classifying images to multiple classes or attributes [5, 22, 2, 25]. This raises a natural question that how *one* retrieval method might take into account of *multiple* criteria in finding the candidates, *e.g.*, returning near-duplicates to a query if presented in a database or otherwise similar ones related to relevant semantic concepts.

Figure 1. A sample query from the *Holidays* dataset: retrieval using a vocabulary tree of local features (first row); retrieval using 1000 semantic attributes (second row); retrieval based on co-indexing of both local features and semantic attributes (third row).

Different lines of retrieval methods tightly couple their search criteria with dramatically different image representations and indexing strategies. For example, representing images by bags of local features [19, 13] that are indexed by vocabulary trees [14] is very effective for near-duplicate or instance level retrieval, *i.e.*, searching the same object or scene with arbitrary changes. Compact hashing codes for global features [21] or semantic attributes from object recognition [2, 25] are efficient for similar image search. Hence, on one hand, it is very hard to merge these diverse representations and indexing schemes, if not impossible. On the other hand, fusing retrieval results [4] requires online extraction of multiple image feature sets and storage for their respective indexes, which is costly in practice. These challenges leave the effort on using multiple search criteria in one retrieval method rarely explored in the literature.

In this paper, we propose to incorporate two search criteria, *i.e.*, image similarities based on local features and

[1]This work was supported in part to Dr. Qi Tian by NEC Laboratories of America, ARO grant W911NF-12-1-0057, NSF IIS 1052851, 2012 UTSA START-R Research Award, and NSFC 61128007.

semantic attributes, into the inverted indexes. Then the retrieval not only searches for candidate images sharing similar local features but also encourages consensus in their semantic similarities as shown in Fig. 1. Towards these ends, we present a semantic-aware co-indexing algorithm which leverages semantic attributes from advanced object recognition to update the inverted indexes of local features quantized by a large vocabulary tree. Specifically, during *off-line* indexing, we adopt the classification scores of 1000 object categories in the *ImageNet* Challenge [1] as the semantic attributes. Then we perform two steps to embed the semantic clue into the indexes: 1) semantic isolated image deletion which removes those images with dissimilar attributes on an inverted index; and 2) semantic nearest image insertion which adds K-nearest images with similar attributes to the inverted indexes of their local features.

The proposed co-indexing technique does not sacrifice *online* query efficiency. During online retrieval, we conduct conventional vocabulary tree based retrieval only using the local features in a query and do *NOT* compute the semantic attributes. Nevertheless, the retrieval implicitly promotes candidates that are potentially with similar attributes to the query because the updated indexes are *semantic-aware*.

In this paper, we discover that editing the inverted index of a single local feature with multi-class classification scores effectively enhances its discriminative ability. This is because the co-indexing jointly considers strong cues to low-level image contents and their semantic meanings, respectively. The online query remains as efficient as before since only local features are extracted. Meanwhile, we manage to consume an acceptable memory cost in the deletion and insertion of images on the indexes. Last but not the least, we do not assume query or database images are related to any of the 1000 object categories, which assures its generalization capability. Extensive experiments on 3 benchmark datasets, *i.e.*, *UKbench*, *Holidays*, *Oxford5K*, validate the merits of the proposed method in comparison with recent image retrieval algorithms.

Large-scale object recognition and near-duplicate image search largely remain independent efforts due to different focuses on recognition accuracy and retrieval scalability. State-of-the-art recognition approaches [16, 12, 10] generally require substantial computation, which are hardly affordable in online retrieval. Existing retrieval methods using multi-cues all extract multiple features online for a query. To our best knowledge, this work is an original effort on improving near-duplicate image retrieval by efficiently utilizing object recognition in *off-line indexing*.

2. Related Work

This work focuses on improving near-duplicate image retrieval by co-indexing object recognition outcomes, which is closely related to vocabulary tree based image retrieval, learning semantic attributes, and how to incorporate two cues in retrieval. Due to space limit, detailed survey of either direction is beyond the scope of this paper.

Indexing bag of local invariant features [13, 19] by visual vocabulary trees [14] has demonstrated an exceptional scalability for large-scale near-duplicate image retrieval by applying a spatial verification [17], Hamming embedding [9], building high-order features [26], or encoding spatial configurations of local features [28, 27, 24, 18].

Large-scale object recognition has achieved a prominent advance recently. For instance, in the *ImageNet* Challenge [1] the recognition accuracy of top-5 candidates among 1000 categories has improved significantly to about 84% by extracting Fisher vectors [16], coding BoW features [19, 12], and learning deep network models [10]. The outcomes of these multi-class classifiers, often referred as semantic attributes [6, 5, 22], present a strong cue to find similar videos [8], faces [11], or images by hierarchical indexing [2] or mid-level weak attributes [25].

These recognition methods [19, 16, 12, 10] are generally expensive due to the extraction of high-dimensional features and classification of thousands of object categories. Consequently, it is unaffordable to take advantage of the semantic attributes directly in near-duplicate retrieval, either in early fusion of the features [7], or late fusion of the retrieval results [4]. Thus, near-duplicate image retrieval [14, 17, 9, 26, 28, 27, 18] using local features and similar image retrieval with attributes [20, 11, 2, 25] remain largely two separate lines of research. In contrast, our approach co-indexes the similarities *w.r.t.* semantic attributes into the inverted indexes of local features. The semantic attributes are computed off-line for database images but not for online query images. Furthermore, we learn the recognition models on totally independent datasets and do not assume query or database images are relevant to any of the object categories. These characteristics distinguish our work from existing efforts on near-duplicate retrieval.

3. Proposed Approach

Image retrieval using vocabulary trees and object recognition are two cornerstones of the proposed semantic-aware co-indexing, which are described in Sec. 3.1 and Sec. 3.2, respectively. Then we present how to off-line co-index semantic attributes among database images (Sec. 3.3) and how to conduct online query (Sec. 3.4) in details. The entire procedure is summarized in Fig. 2.

3.1. Image retrieval with vocabulary trees

We employ the vocabulary tree based approach [14] as the baseline. Denote q a query image and d an database image, and q is represented by a bag S_q of local descriptors $\{\mathbf{x}_i\}_{i \in S_q}$, where $\mathbf{x}_i \in \mathbb{R}^D$ indicate SIFT descriptors [13] of dimension $D = 128$, so does $\{\mathbf{x}_j\}_{j \in S_d}$ for d.

A visual vocabulary tree T is obtained by hierarchical

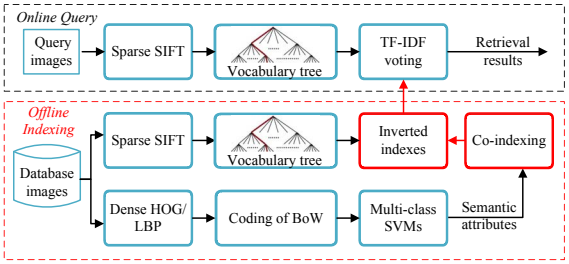

Figure 2. The block diagram of the semantic-aware co-indexing.

K-means clustering of local descriptors (extracted from a *separate* dataset), with a branch factor B and depth L. This tree T typically is very deep and contains millions of leaf nodes, *e.g.*, $B = 10$ and $L = 7$, in order to achieve fast quantization and high distinctiveness. We quantize $\{\mathbf{x}_i\}$ along T to the corresponding tree nodes or visual words $T(\{\mathbf{x}_i\}) \doteq \{v_i\}$. Then the similarity $sim(q, d)$ between q and d is defined as the average TF-IDF (term frequency-inverse document frequency) of these visual words:

$$sim(q,d) \doteq \frac{1}{|S_q||S_d|} \sum_{v \in T(\mathbf{x}_i) \cap T(\mathbf{x}_j)} w(v), \quad (1)$$

$$w(v) \doteq \text{idf}^2(v) = \log^2\left(\frac{M}{M_v}\right), \quad (2)$$

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. Note $T(\{\mathbf{x}_i\})$ allows repeated nodes to model the TF. The list of M_v images and the TFs of v in them are stored in the inverted index of v for fast access, and are denoted by $I(v) = \{d_m\}_{m=1}^{M_v}$ and $\{\text{tf}_{d_m}(v)\}_{m=1}^{M_v}$.

3.2. Semantic attributes from object recognition

We follow the Bag-of-Words (BoW) paradigm to learn $C = 1000$ object category classifiers from the training images in the LSVRC [1], a subset of *ImageNet* dataset. For each training image, we obtain the BoW features from dense HOG and LBP which are further encoded by local coordinate coding to train multiple one-against-all linear SVM classifiers [12]. The recognition accuracy for the top-5 candidate categories is about 65% according to the LSVRC's flat evaluation metric [1]. The SVM margin scores of these 1000 categories are denoted by $\{f_c\}_{c=1}^{C}$ for an image, which are regarded as its semantic attributes.

These 1000 categories in LSVRC certainly cannot cover millions of all possible objects in the real-world. Therefore, different from previous work [2], we do not implicitly assume the query or the database images are related to one object category in these semantic attributes. In fact, our testing query and database images are independent from the *ImageNet* dataset, hence it is likely one image is relevant to *none* or *multiple* categories in these 1000 attributes. Therefore, we do not assume "is a" or "has a" meaning for these attributes but a weaker "relevant to" relation.

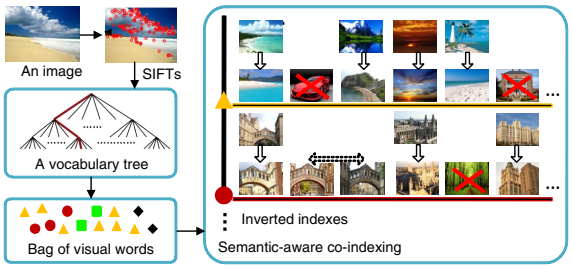

Figure 3. Illustration of the co-indexing process: based on the attributes, isolated images on the inverted indexes (red and yellow lines) of two visual words (red ball and yellow triangle) are deleted, marked by red cross marks; nearest images are inserted to the indexes, indicated by solid arrows; and no insertion for nearest images that are already on one index, linked by a dash arrow.

3.3. Semantic-aware off-line co-indexing

The inverted indexes from visual words to images and their TFs summarize all knowledge about the database in the vocabulary tree based method [14]. Though hundreds of local descriptors are capable of finding the near-duplicates to a query via the inverted indexes, the discriminative capacity of a single local descriptor is limited due to two reasons. First, some similar local descriptors may appear in dramatically different images. Second, local descriptors even in near-duplicate images may fall to different visual words due to quantization errors. These two issues may lead to unsatisfied retrieval results, *e.g.*, returning images with similar local textures but appear irrelevant to users. These motivate us to explore how to embed extra discriminative clue into the individual indexes of local descriptors.

we propose a semantic-aware co-indexing to address the two issues by off-line updating the inverted indexes with the image similarities induced from semantic attributes. The attributes obtained by multi-class recognition may reveal an image's rough high-level semantic contents, which is often complimentary to the low-level descriptors. The proposed co-indexing involves two steps: 1) semantic isolated image deletion which spots the isolated images on an inverted index according to their attributes and removes them from the index; and 2) semantic nearest image insertion which searches for the top nearest neighbors of database images using the attributes and inserts them to the inverted indexes. By these means, the images on one index tend to be more consensus with each other and in return more effective for near-duplicates retrieval, as illustrated in Fig. 3.

3.3.1 Distance of semantic attributes

Let us first define how to measure the distance between semantic attributes given in Sec. 3.2 before proceeding to the specific schemes to alter the inverted indexes. We convert the SVM scores $\{f_c\}_{c=1}^{C}$ to a probabilistic representation $\{p_c\}_{c=1}^{C}$ by fitting a sigmoid function to each dimension, similar as in [2]. For an image d, each entry $p_c(d)$ is proportional to $P(c|d)$, the likelihood of being relevant to

the category c. As discussed in Sec. 3.2, $\{p_c(d)\}_{c=1}^{C}$ is not naturally normalized because the C categories may not cover one image's semantic contents or one image may be related to multiple categories, so we regard it as a partial probability distribution. For two images d_m and d_n, we employ the Total Variance Distance (TVD) to measure the semantic distance between their partial probability vectors:

$$\mathrm{TVD}(d_m, d_n) = \sum_{c=1}^{C} |p_c(d_m) - p_c(d_n)|. \quad (3)$$

In our settings, the TVD indicates the largest possible divergence of probabilities that two images could be recognized as being related to the same object categories. Thus, it reasonably reflects the semantic distance between two images, *e.g.*, for exact or near-duplicate images, the TVD shall be close to zero. In this paper, we do not model the semantic relation among different categories and deem them independent, so we choose the TVD due to its intuition and efficiency. More distance metrics, such as the ones considering hierarchical semantic relationships among concepts [2], *etc.*, are to be investigated in the future work.

3.3.2 Semantic isolated image deletion

Encoding nondiscriminative information [23] into the inverted indexes may not help the retrieval, since it makes the right candidate hard to stand out. An isolated image on an inverted index, whose appearance is quite different from any other image on the same index, would contribute less in image retrieval, since they are less likely to help to find similar images. Hence, we utilize a semantic isolated image deletion procedure to filter out isolated images from the perspective of semantic attributes, so as to obtain more consistent inverted indexes.

For the images indexed to v, *i.e.*, $I(v) = \{d_m\}_{m=1}^{M_v}$, if $M_v \geq 3$ we calculate the semantic distances of attributes as in Eq. (3) among them and delete the isolated images, which are specified as being semantically distinct from all the others in the same index, *i.e.*, an image d_n that satisfies:

$$\min_{m=1, m\neq n}^{M_v} \mathrm{TVD}(d_n, d_m) > \rho, \quad (4)$$

where ρ is a threshold to tune the portion of images to be deleted. The semantic isolated image deletion can effectively reduce the index size without impacting the retrieval precision in our experiments.

3.3.3 Semantic nearest image insertion

After the deletion of isolated images, we take advantage of the attribute feature to insert semantically similar images to the inverted indexes. We compare the attributes of all database images using Eq.(3) to identify their top K nearest images, denoted by $N_K(d)$. If one nearest image d_k to d does not appear on the same inverted index, we insert it to the entry of d's inverted index, whose data structure is illustrated in Fig. 4. Namely, d_k is inserted to $I(v)$, if

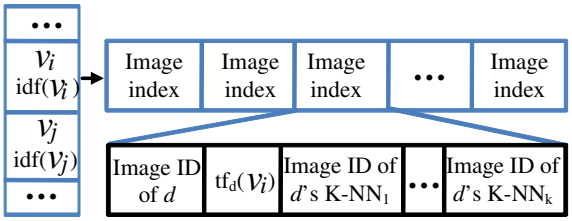

Figure 4. The proposed data structure for co-indexing.

$$d_k \in N_K(d), d \in I(v), d_k \notin I(v). \quad (5)$$

The desired K for d shall equal to the number of semantically similar images to d. However, it is hard to pre-define or learn K because database images commonly have variable numbers of semantically similar images. To address this problem, we determine the value of K specific to individual d. For d, we seek a K_d, after where the similarity between d and d_k drops most sharply, *i.e.*,

$$K_d = \arg\max_{k=1:\mathbb{K}} \frac{\partial^2 (\mathrm{TVD}(d, d_k))}{\partial k^2}, \quad (6)$$

where $\mathbb{K}$ is the maximum number of semantic nearest neighbor images to check. Note, it is very likely that $d_k \in N_K(d)$ already appears on the inverted index that includes d, since semantically similar images shall share some similar local features. For these cases we do not introduce redundant images in the index. As shown in Fig. 3, the two images between the *dash* black arrow are semantic nearest neighbors that are already on the same index. Thus, semantic nearest image insertion does not increase the size of indexes by K times. We will discuss how to further reduce the memory cost of the indexes in Sec. 4.1. The set of d_k that is inserted to the indexes due to d is referred by $G_{\mathrm{KNN}}(d)$, not including those already in the indexes. In Sec. 5, we test the impacts of deleting different portions of isolated images, inserting different numbers of similar images, as well as the effectiveness of Eq.(6).

3.4. Semantic-aware online query

The online query process is almost identical to the conventional vocabulary tree based retrieval, except that we implicitly consider the joint similarity based on the local features and the semantic attributes. Given q and d are the query and a database image, respectively, then the similarity between q and d is conceptually updated to:

$$\widehat{sim}(q, d) \doteq sim(q, d) + \sum_{\{d_g | d \in G_{\mathrm{KNN}}(d_g)\}} \omega \times sim(q, d_g), \quad (7)$$

where ω is a weighting parameter of the contribution from semantic attributes, and the second term includes the set

of images d_g such that d is within their K semantic nearest neighbors (note d is d_g's neighbor, while d_k is d neighbor). In another word, we use $sim(q, d_g)$ computed with Eq. (1) to update $\widehat{sim}(q, d)$, because d is semantically similar to d_g. Consequently, the candidate images, not only sharing a large portion of similar local feature but also being consistent with the semantic attributes, will be ranked higher in the retrieved set. Ideally, the weighting parameter ω shall be determined by the TVD between d and d_g or its rank in the $N_K(d_g)$, but different ω require extra storage in the inverted indexes. According to our experiments, this memory overhead is not worthwhile, thus as a compromise we use a fixed $\omega = 0.2$ in our implementation.

Note that during online retrieval, we do not need to explicitly identify the set $G_{K\text{NN}}(d_g)$ to compute Eq. (7). Instead, we only need to scan image lists attached to the visual words found in the query, as well as the semantic nearest images inserted to the indexes. We summarize the online computation of Eq. (7) in Algorithm 1.

Algorithm 1 Similarity calculation between a query q and all database images.

Input: the inverted indexes $I(v)$ stored as in Fig. 4; the BoW representation of a query image: $\{v_i\}_{i \in S_q}$; the weighting parameter ω.
Output: the similarity vector of $\widehat{sim}(q, d)$.

```
for each visual word v in {v_i}_{i∈S_q} do
  for each database image d in the image list I(v) do
    sim^(q,d) = sim^(q,d) + tf_d(v)tf_q(v)idf(v)
    for each semantic nearest image d_k of d do
      sim^(q,d_k) = sim^(q,d_k) + ω × tf_d(v)tf_q(v)idf(v)
    end for
  end for
end for
```

4. Scalability Analysis

The scalability in terms of computational complexity and memory cost is utmost critical to image retrieval. As our semantic-aware co-indexing focuses mainly on off-line indexing, we discuss the memory cost first and then its impact to online query and off-line indexing efficiency.

4.1. Memory consumption

In the vocabulary tree based retrieval, the total memory cost of inverted indexes is proportional to the total number of local features in the database, *i.e.*, $\sum_{m=1}^{M} |S_{d^m}|$. In semantic nearest image insertion, we add at most K nearest neighbors *per image* to the indexes. In fact, the memory cost shall be at $O(KM)$, if we maintain a separate table for the K semantic nearest neighbors of database images which is a quite marginal memory overhead. This is substantially smaller than $\sum_{m=1}^{M} |S_{d^m}|$ by 2-3 order of magnitudes, because the number of local features extracted from each image, *i.e.*, $|S_{d^m}|$ easily exceeds 1000. However, the online query demands for efficiency as high as possible, therefore we choose to consume the inverted indexes to obtain the semantic nearest neighbors in a *streaming* manner, rather than jumping to a separate table, which minimizes CPU cache misses. This is why we adopt the data structure in Fig. 4, trading memory for efficiency.

In addition, as explained in Sec. 3.3.3, the semantic nearest image insertion avoids adding redundant images if they already present in the inverted index. Thus, our method does not increase the index size if the indexes of local features largely agree with the semantic features. Moreover, the isolated image deletion removes a certain number of images and their associated TFs from the indexes and saves some storage. Therefore, the overall memory cost of the co-indexing is acceptable. For example, after deleting 20% images first and then inserting $K = 2$ neighbors, the proposed co-indexing requires about 50% additional storage using a tree with 1 million leaf nodes, which is even less than the memory cost in some recent approaches which store the spatial configuration or contexts of local features into the inverted indexes [9, 28, 27, 24].

4.2. Computational complexity

The computation of online query using a vocabulary tree is composed of two parts: 1) the local feature extraction and their quantization, and 2) the voting of TF-IDFs along the inverted indexes. The former part is independent from the indexing and remains unchanged in our approach. For the latter, the co-indexing method roughly requires additional one multiplication and one add operation for each semantic nearest image inserted. The computational overhead can be estimated by the average increment of index lengths multiplied by the ratio of these two additional operations in the TF-IDF voting, which is about $5 - 10\%$ query time compared to the baseline in large-scale problems.

The major computational demand in our method is at the off-line stage. The major part is to perform the large-scale object category recognition for all the database images, which are easily parallelized. We finished extracting the 1000D attributes of 1.3 million images within 3 hours using 200 mappers in Hadoop. The search for semantic nearest neighbors in the database also allows for parallel processing. The TVD computation in Eq. (3) is very efficient and allows hashing technique for acceleration. Our experiments validate that the co-indexing strategy is applicable to millions of images.

5. Experiments

We evaluate the proposed method on 3 public benchmark datasets, *i.e.*, the *UKbench* [14], *Holidays* [9], and *Oxford5K* [17]. These 3 datasets represent diverse near-duplicate image retrieval tasks, *i.e.*, search for the same object in the *UKbench* which contains 2,550 objects under 4 different viewpoints; search for the same scene in the

Method	$T16^5$	$T10^7$	SA	GIST	Color
UKbench, N-S	2.85	3.42	3.17	1.93	2.48
Holidays, mAP(%)	59.70	73.79	71.62	40.89	46.33
Oxford5K, mAP(%)	51.32	68.27	42.62	21.92	7.83

Table 1. The retrieval performance of individual methods. The first two columns are the baselines using a vocabulary tree. The last three columns are retrieval performance directly using the semantic attributes (SA), GIST and color, respectively.

Holidays which includes 500 queries from 1,491 annotated scene images; and search for the 55 landmarks in the *Oxford5K* from 5,063 annotated landmark images. Note, the setting in the *Oxford5K* does not favor the semantic attributes because all its database images roughly belong to one category, *i.e.*, street-view landmark buildings. Besides these, we conduct large-scale experiments by mixing the three datasets with 1.3 million images collected from *Flickr* (as distraction sets) [2], respectively. Note none of our test images are from the *ImageNet* or related to the 1000 object categories, to verify the generalization ability to attributes.

5.1. Methods

We implement two variants of vocabulary tree based retreival [14] as the baseline, to show the co-indexing helps for different trees with/out spatial contexts. The first one uses a relatively shallow tree with $B = 16$ and $L = 5$ (about 1 million leaf nodes), denoted by $T16^5$. The other utilizes a deeper tree with $B = 10$ and $L = 7$ (about 3 million leaf nodes) and also records local feature's spatial contexts [24] with 4 bytes, denoted by $T10^7$. Both trees are constructed by hierarchically clustering of 100 million SIFT descriptors extracted from images crawled from the Internet. The number of SIFT features extracted from a query ranges from 500 to 2500 in our experiments.

The proposed co-indexing technique is not restricted to the usage of semantic attributes, so we also compare with the methods using the GIST and color histograms in the co-indexing, to demonstrate the advantage of being semantic-aware. Three types of features are hence tested in co-indexing, *i.e.*, the 1000D semantic attributes explained in Sec. 3.2, the 512D GIST features [15], and 512D color histograms in the HSV color space (256 bins for Hue, and 128 bins for Saturation and Value respectively). We employ the L2 distance for the GIST and color features in determining the isolated and nearest images. These three features are denoted by SA, GIST, and Color hereafter. We hence use $T16^5 + SA$ and $T10^7 + SA$ to denote our semantic-aware co-indexing algorithm.

We adopt the metrics in the original papers of the 3 datasets to evaluate the retrieval performance: the recall rate for the top-4 candidates (referred as the N-S score) for the *UKbench*, and the mAP (mean average precision) for the *Holidays* and *Oxford5K*. The performance of using the two baselines and the 3 features, *i.e.*, SA, GIST, and Color, is summarized in Table 1, *e.g.*, directly using the 1000D attributes to retrieve and rank the candidate images.

[2]The test images, their local features and semantic attributes are available upon request to the first author.

5.2. Performance

We first compare the isolated image deletion and nearest image insertion against the baselines, respectively, including the sensitivity study of the key parameters. Then, we present the overall performance of semantic-aware co-indexing on these datasets, as well as mixed with the large-scale distractor images.

Semantic isolated image deletion. During the deletion of isolated images, we tune the threshold ρ for the three types of features to remove the same ratio Δr of inverted indexes. The retrieval performance is summarized in Fig. 5, where the retrieval precision remains almost unchanged or even improves slightly when using SA to delete 10%-20% of the inverted indexes. This validates that enforcing the semantic consensus among images on one index effectively saves storage without hurting the retrieval precision.

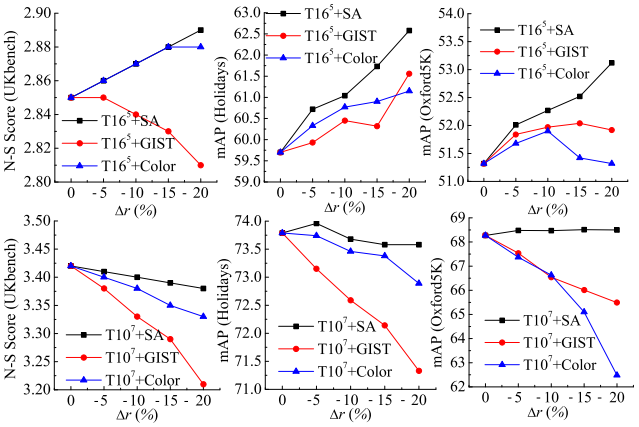

Figure 5. Comparison of isolated image deletion on the *UKbench*, *Holidays*, and *Oxford5K* datasets.

Semantic nearest image insertion. We first test inserting fixed $K = 1$ to 4 of the nearest images to the inverted indexes according to SA, GIST and Color, whose retrieval performance is presented in Fig. 6. On the *UKbench*, the retrieval performance improves considerably, *i.e.*, the N-S scores from 2.85 to 3.39 and 3.42 to 3.61 respectively when $K = 3$ for $T16^5 + SA$ and $T10^7 + SA$. The mAP jumps from 59.70% to 75.60% and 73.79% to 80.99% on the *Holidays* over these two baselines. The improvement on the *Oxford5K* is not that significant compared to the other two datasets, yet the mAP still increases by 7% for $T16^5 + SA$, partly because the 1000D attributes are generic object categories. Fine-grained attributes particularly for buildings may be more appropriate for landmark search. In contrast, the performance gain of using GIST and Color is marginal compared to SA. This verifies the advantages of embedding semantic attribute to the indexes. Details of

Methods	$T16^5 + SA$			$T10^7 + SA$		
Dataset	*UKbench*	*Holidays*	*Oxford*	*UKbench*	*Holidays*	*Oxford*
Performance ($\bar{K_d}$)	3.38 (3.02)	76.13 (2.80)	57.46 (3.82)	3.60 (3.02)	81.60 (2.80)	68.54 (3.82)
Performance (Fixed K)	3.39 (3)	75.60 (2)	58.11 (4)	3.61 (3)	80.99 (3)	68.52 (4)

Table 2. Performance comparison of nearest image insertion with automatically selected K_d and the best fixed K in Fig. 6.

Fig. 5 and Fig. 6 are available in **Supplementary** material.

Next we test the scheme of image-specific K_d in Eq. (6), rather than setting a fixed K. The performance of $T16^5 + SA$ and $T10^7 + SA$ with a varying K_d is summarized in Table 2, which shows the average $\bar{K_d}$ is close to the best fixed K and the performance is quite comparable to those in Fig. 6. This verifies the effectiveness of the automatic selection scheme for K. We set $\mathbb{K}$ in Eq. (6) as 5.

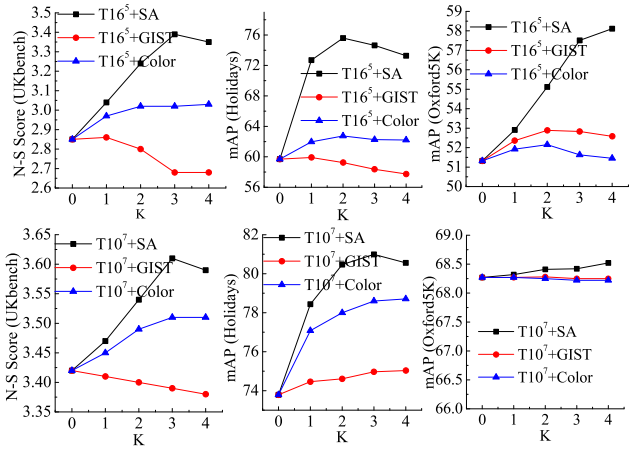

Figure 6. Comparison of nearest image insertion on the *UKbench*, *Holidays*, and *Oxford5K* datasets.

Semantic-aware co-indexing. Now we show the overall performance of the semantic-aware co-indexing and the ratio Δr of memory cost increment. To present the results concisely, we fix the ratio of isolated image deletion to 20% for $T16^5 + SA$ and 5% for $T10^7 + SA$. The results are presented in Table 3, where the first column shows the baseline performance. The retrieval precision improves consistently as in the previous two experiments, *e.g.*, $T10^7 + SA$ achieves N-S=**3.60** on the *UKbench* and mAP=**80.86** on the *Holidays*. The memory overhead Δr is reasonably higher in a deep tree than in a shallow one, since the feature quantization is finer. As discussed in Sec. 4.1, using a separate table for the nearest neighbors can bound the memory usage. The computational overhead of the co-indexing is hardly noticeable on these small-scale datasets, only about *1-3ms*, for the quantization of local features dominates the online computation.

We also conduct the experiment excluding the queries from the database. On the *UKbench*, we employ 1/4 images of the dataset as the query and the rest 3/4 in the database for indexing, thus the query images and dataset

Methods	**Proposed**	[17]	[9]	[27]	[24]	[18]	[3]
UKbench, N-S	**3.60**	3.45	3.42	3.26	3.56	3.52	N/A
Holidays, mAP(%)	**80.86**	N/A	81.3	N/A	78.1	76.2	69.9
Oxford5K, mAP(%)	**68.72**	64.5	61.5	71.3	N/A	75.2	N/A

Table 4. Overall performance comparison with the state of arts.

images are totally separated. We hence run 4 rounds of experiments, the average N-S score of $T16^5 + SA$ jumps from 1.85 to 2.35 (maximum 3), which is consistent with the improvement from 2.85 to 3.37 (maximum 4); similarly $T10^7 + SA$ improves from 2.40 to 2.63 (maximum 3) *vs.* from 3.37 to 3.60 (maximum 4) in Table 3.

The comparison with recent retrieval methods (without re-ranking and query expansion) are shown in Table 4, which demonstrates that the performance of the proposed co-indexing is very competitive. Attributes are also utilized in [3] for image search, yet differently it extracts both the attributes and visual features online from the queries. Sample retrieval results are in the **Supplementary** material.

The large-scale experiments. Using 1.3 million images collected from *Flickr* as distractors, we conduct the retrieval with the original queries in the 3 datasets based on $T16^5 + SA$ and $T10^7 + SA$, where about 20% isolated images are deleted in $T16^5 + SA$ and about 5% isolated images deleted in $T10^7 + SA$, and K is automatically selected with Eq. (6). The experimental results are summarized in Table 5. The average retrieval time $\bar{t}$ (not including feature extraction) only increases slightly over the baseline, which is about 140-210ms among 1.3 million images. We also observe promising improvements by using the co-indexing, *e.g.*, the N-S score increases from 2.42 to 2.83 in $T16^5+SA$ and from 3.14 to 3.39 in $T10^7 + SA$. Hence, these results demonstrate that the semantic-aware co-indexing approach scales up well for image retrieval from millions of images.

5.3. Discussions

The local feature based near-duplicate image retrieval essentially relies on finding a small set of matched local descriptors to retrieve the candidates. The fundamental rational of the proposed co-indexing is that we leverage another strong cue, *i.e.*, the semantic attributes, to enhance the discriminability of *individual* local feature's inverted index, resulting in a prominent improvement on the overall discriminative ability of inverted indexes.

It is not a must to have semantic attributes and their K-nearest neighbors available for all database images in co-indexing. Investigation of selectively co-indexing a portion of database images with reliable attributes and approximate nearest neighbor search will be our future work.

6. Conclusions

In this paper, we present a new approach to jointly indexing both local features and semantic attributes for image retrieval. By updating the indexes of local features guided by the semantic features, the proposed retrieval

K	Base.	0	+1	+2	+3	+4	Base.	0	+1	+2	+3	+4	Base.	0	+1	+2	+3	+4
Datasets	*UKbench* (N-S)						*Holidays* mAP(%)						*Oxford5K* mAP(%)					
$T16^5 + SA$	2.85	2.89	3.06	3.23	**3.37**	3.34	59.70	62.58	73.54	**76.18**	75.57	74.17	51.22	53.12	55.77	56.59	58.27	**58.32**
Δr (%)	0	-20	+8.61	+40.4	+69.1	+104	0	-20	+9.75	+42.4	+72.75	+103	0	-20	+13.6	+46.4	+80.5	+115
$T10^7 + SA$	3.42	3.41	3.46	3.52	**3.60**	3.58	73.79	74.06	78.61	80.45	**80.86**	80.82	68.27	68.27	68.53	68.61	68.63	**68.72**
Δr (%)	0	-5	+26.8	+55.3	+83.7	+112	0	-5	+29.0	+59.3	+89.0	+113	0	-5	+29.1	+59.7	+89.7	+120

Table 3. The overall performance of semantic-aware co-indexing on the *UKbench*, *Holidays*, and *Oxford5K*.

Methods	$T16^5 + SA$						$T10^7 + SA$					
Datasets	*UKbench* (N-S)		*Holidays* mAP(%)		*Oxford5K* mAP(%)		*UKbench* (N-S)		*Holidays* mAP(%)		*Oxford5K* mAP(%)	
K_d	0	3.03	0	3.24	0	3.9	0	3.06	0	3.14	0	4.1
Performance	2.42	**2.83**	53.23	**63.34**	44.25	**48.29**	3.14	**3.39**	54.30	**62.77**	60.39	**60.52**
Δr (%)	0	+86.15	0	+84.13	0	+125.3	0	+87.2	0	+89.4	0	+111.0
$\bar{t}$ (ms)	132	135.1	182.0	186.7	158.7	162.7	93	169.6	101	162.7	125	211.3

Table 5. The performance of semantic-aware co-indexing with 1.3 million distractor images. The columns with $\bar{K_d} = 0$ are the baselines.

algorithm effectively applies two search criteria to enhance the overall discriminative capability of the inverted indexes, leading to more satisfactory retrieval results to users. The co-indexing introduces very small online computation and consumes manageable additional memory. This semantic-aware co-indexing method can be easily reproduced by other motivated researchers. These warrant further investigating incorporation of multiple cues into off-line indexing.

References

[1] Large scale visual recognition challenge. http://www.image-net.org/challenges/LSVRC/2010, 2010. 2, 3

[2] J. Deng, A. C. Berg, and L. Fei-Fei. Hierarchical semantic indexing for large scale image retrieval. In *CVPR*, 2011. 1, 2, 3, 4

[3] M. Douze, A. Ramisa, and C. Schmid. Combinging attributes and fisher vectors for efficient image retrieval. In *CVPR*, 2011. 7

[4] R. Fagin, R. Kumar, and D. Sivakumar. Efficient similarity search and classification via rank aggregation. In *ACM SIGMOD*, 2003. 1, 2

[5] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009. 1, 2

[6] V. Ferrari and A. Zisserman. Learning visual attributes. In *NIPS*, 2007. 2

[7] P. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *ICCV*, 2009. 2

[8] A. G. Hauptmann, M. G. Christel, and R. Yan. Video retrieval based on semantic concepts. *Proc. IEEE*, 96(4):602–622, 2008. 2

[9] H. Jégou, M. Douze, and C. Schmid. Improving bag-of-feature for large scale image search. *IJCV*, 87(3):316–336, 2010. 2, 5, 7

[10] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 2

[11] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In *ICCV*, 2009. 2

[12] Y. Lin, F. Lv, S. Zhu, M. Yang, T. Cour, K. Yu, L. Cao, and T. Huang. Large-scale image classification: fast feature extraction and svm training. In *CVPR*, 2011. 2, 3

[13] D. G. Lowe. Distinctive image features from scale invariant keypoints. *IJCV*, 60(2):91–110, 2004. 1, 2

[14] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006. 1, 2, 3, 5, 6

[15] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001. 1, 6

[16] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*, 2010. 2

[17] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007. 2, 5, 7

[18] X. Shen, Z. Lin, J. Brandt, S. Avidan, and Y. Wu. Object retrieval and localization with spatially-constrained similarity measure and k-NN reranking. In *CVPR*, 2012. 2, 7

[19] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV*, 2003. 1, 2

[20] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *PAMI*, 30(11):1985–1970, Nov. 2008. 1, 2

[21] A. Torralba, R. Fergus, and Y. Weiss. Small codes and large image databases for recognition. In *CVPR*, 2008. 1

[22] L. Torresni, M. Szummer, and A. Fitzgibbon. Efficient object category recognition using classemes. In *ECCV*, 2010. 1, 2

[23] P. Turcot and D. G. Lowe. Better matching with fewer features: the selection of useful features in large database recognition problems. In *ICCV Workshop*, 2009. 4

[24] X. Wang, M. Yang, T. Cour, S. Zhu, K. Yu, and T. X. Han. Contextual weighting for vocabulary tree based image retrieval. In *ICCV*, 2011. 2, 5, 6, 7

[25] F. Yu, R. Ji, M. Tsai, G. Ye, and S.-F. Chang. Weak attributes for large-scale image retrieval. In *CVPR*, 2012. 1, 2

[26] 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*, 2010. 2

[27] Y. Zhang, Z. Jia, and T. Chen. Image retrieval with geometry-preserving visual phrases. In *CVPR*, 2011. 2, 5, 7

[28] W. Zhou, Y. Lu, H. Li, Y. Song, and Q. Tian. Spatial coding for large scale partial-duplicate web image search. In *ACM Multimedia*, 2010. 2, 5