Improving bag-of-features for large scale image search

Pre-print version of July 27th 2009, corrected March 15th 2011*

Hervé Jégou Matthijs Douze Cordelia Schmid

March 15, 2011

Abstract

This article improves recent methods for large scale image search. We first analyze the bag-of-features approach in the framework of approximate nearest neighbor search. This leads us to derive a more precise representation based on 1) Hamming embedding (HE) and 2) weak geometric consistency constraints (WGC). HE provides binary signatures that refine the matching based on visual words. WGC filters matching descriptors that are not consistent in terms of angle and scale. HE and WGC are integrated within an inverted file and are efficiently exploited for all images in the dataset. We then introduce a graph-structured quantizer which significantly speeds up the assignment of the descriptors to visual words. A comparison with the state of the art shows the interest of our approach when high accuracy is needed.

Experiments performed on three reference datasets and a dataset of one million of images show a significant improvement due to the binary signature and the weak geometric consistency constraints, as well as their efficiency. Estimation of the full geometric transformation, i.e., a re-ranking step on a short-list of images, is shown to be complementary to our weak geometric consistency constraints. Our approach is shown to outperform the state-of-the-art on the three datasets.

1 Introduction

We address the problem of searching for similar images in a large set of images. Similar images are defined as images of the same object or scene viewed under different imaging conditions, cf. Fig. 16 for examples. Many previous approaches have addressed the problem of matching such transformed images [1, 2, 3, 4, 5]. They are in most cases based on local invariant descriptors, and either match descriptors between individual images or search for similar descriptors in an efficient indexing structure. Various approximate nearest neighbor search algorithms such as kd-tree [1] or sparse coding with an over-complete basis set [6] allow for fast search in small datasets. The problem with these approaches is that all individual descriptors need to be compared to and stored.

In order to deal with large image datasets, most of the recent image search systems build upon the bag-of-features representation, introduced in the context of image search in [4]. Descriptors are quantized into visual words with the k-means algorithm. An image is then represented by the frequency histogram of visual words obtained by assigning each descriptor of the image to the closest visual word. Fast access to the frequency vectors is obtained by an inverted file system. Note that this approach is an approximation to the direct matching of individual descriptors and somewhat decreases its performance. It compares favorably in terms of memory usage against other approximate nearest neighbor search algorithms, such as the popular Euclidean locality sensitive hashing (LSH) [7, 8]. LSH typically requires 100–500 bytes per descriptor to index, which is not tractable, as a one million image dataset typically produces up to 2 billion local descriptors.

Some recent extensions of the BOF approach speed up the assignment of individual descriptors to visual words [5, 9] or the search for frequency vectors [10, 11]. Others improve the discriminative power of the visual words [12], in which case the entire dataset has to be known in advance. It is also possible to increase the performance by regularizing the neighborhood structure [10] or using multiple assignment of descriptors to visual words [10, 13] at the cost of reduced efficiency. Finally, post-processing with spatial verification, a re-occurring technique in computer vision [1], improves the retrieval performance. Such a post-processing is evaluated in [9].

In this article we present an approach complementary to those mentioned above. We make the distance between visual word frequency vectors more significant by using a more informative representation. Firstly, we add binary signatures to the descriptors, which are compared with the Hamming distance, resulting of a *Hamming Embedding* (HE) of the SIFT descriptors. The idea of using short binary codes was recently proposed in [14], where they are used to compress global GIST descriptors [15]. Secondly, we integrate a *weak geometric consistency* (WGC) check within the inverted file system which penalizes the descriptors that are not consistent in terms of angle and scale. A priori knowledge of the transformations can be combined with WGC. This contribution can be viewed as a partial answer to the question in [9] on how to integrate geometrical information in the index for very large datasets. Both HE and WGC require to store additional information, hence increasing the memory usage of the index. However the

*Thank you to Yusuke Uchida for reporting a problem with Figure 13.

efficiency of the search is not significantly modified when these methods are jointly used.

We then propose two strategies to improve the assignment of SIFT descriptors to visual words. First, we introduce a graph-structured quantizer that improves the efficiency of the descriptor assignment. Second, we propose an asymmetric multiple assignment strategy that reduces the probability of missing matching descriptor pairs in case of mismatched visual word assignment.

This article is organized as follows. The evaluation of a BOF representation as an approximate nearest neighbor search approach is presented in Section 2. Our contributions, HE and WGC, are described in sections 3 and 4. Section 5 analyzes the complexity of querying the inverted file, and our strategy to assign descriptors to visual words. The practical complexity of our approach within an inverted file system and the memory usage are discussed in Section 6. Finally, Section 7 presents the experimental results.

2 Voting interpretation of bag-of-features

In this section, we show how image search based on BOF vectors compared with the cosine similarity (or equivalently, with the L_2 distance), can be interpreted as a voting system which matches individual descriptors with an approximate nearest neighbor (NN) search. We then evaluate BOF from this perspective. The main notations used in this article are summarized in Fig. 1.

2.1 Voting approach

Given a query image represented by its local descriptors $y_{i'}$ and a set of database images $j = 1..n$ represented by their local descriptors $x_{i,j}$, a voting system can be summarized as:

1. Dataset image scores s_j are initialized to 0.

2. For each query image descriptor $y_{i'}$ and for each descriptor $x_{i,j}$ of the dataset, update the score s_j of the corresponding image by

$$s_j := s_j + f(x_{i,j}, y_{i'}), \tag{1}$$

where f is a matching function that reflects the similarity between descriptors $x_{i,j}$ and $y_{i'}$. For a matching system based on ε-search or $k-$NN, $f(.,.)$ is defined as

$$f_\varepsilon(x,y) = \begin{cases} 1 & \text{if } d(x,y) < \varepsilon \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

and

$$f_{k\text{-NN}}(x,y) = \begin{cases} 1 & \text{if } x \text{ is a } k\text{-NN of } y \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where $d(.,.)$ is a distance (or dissimilarity measure) defined in descriptor space. SIFT descriptors are typically compared using the Euclidean distance.

3. The image score $s_j^* = g_j(s_j)$ used for ranking is obtained from the final s_j by applying a post-processing function g_j:

$$s_j^* = g_j\left(\sum_{i'=1}^{m'}\sum_{i=1}^{m_j} f(x_{i,j}, y_{i'})\right). \tag{4}$$

The simplest choice for g_j is the identity: $s_j^* = s_j$. In this case the score reflects the number of matches between the query and each database image. Note that this score counts possible multiple matches of a descriptor. Another popular choice is to take into account the number of image descriptors, for example $s_j^* = s_j/m_j$. The score then reflects the rate of descriptors that match.

2.2 Bag-of-features: voting and approximate NN interpretation

Bag-of-features (BOF) image search uses descriptor quantization. A quantizer q is formally a function

$$\begin{array}{rccc} q: & \mathbb{R}^d & \rightarrow & [1,k] \\ & x & \mapsto & q(x) \end{array} \tag{5}$$

that maps a descriptor $x \in \mathbb{R}^d$ to an integer index. The quantizer q is often obtained by performing k-means clustering on a learning set. The quantizer $q(x)$ is then the index of the centroid closest to the descriptor x. Intuitively, two descriptors x and y which are close in descriptor space satisfy $q(x) = q(y)$ with a high probability. The matching function f_q defined as

$$f_q(x,y) = \delta_{q(x),q(y)}, \tag{6}$$

allows the efficient comparison of the descriptors based on their quantized index. Injecting this matching function in (4) and normalizing the score by the number of descriptors of both the query image and the dataset image j, we obtain

$$s_j^* = \frac{1}{m_j\, m'}\sum_{i'=1}^{m'}\sum_{i=1}^{m_j}\delta_{q(x_{i,j}),q(y_{i'})} = \sum_{l=1}^{k}\frac{m'_l}{m'}\,\frac{m_{l,j}}{m_j}, \tag{7}$$

where m'_l (respectively $m_{l,j}$) denotes the number of descriptors of the query (respectively dataset image j) that are assigned to the visual word l. In this equation, the normalizing value m' does not affect the ordering of the dataset images. Note that these scores correspond to the inner product between two BOF vectors. They are computed very efficiently using an inverted file, which exploits the sparsity of the BOF, i.e., the fact that $\delta_{q(x_{i,j}),q(y_{i'})} = 0$ for most (i,j,i') tuples.

At this point, the scores do not take into account the *tf-idf* scheme [4], which weights the visual words according to their frequency: rare visual words are assumed to be more discriminative and are assigned higher weights. In this case the matching function f can be defined as

$$f_{\text{tf-idf}}(x,y) = \left(\text{tf-idf}\left(q(y)\right)\right)^2 \delta_{q(x),q(y)}, \tag{8}$$

n	number of images in the dataset
d	dimension of the local descriptors
m_j	number of descriptors describing image j of the dataset
m'	number of descriptors describing the query
m'_l	number of descriptors describing the query assigned to the visual word l
k	number of centroids (=visual words) defining the quantizer
$x_{i,j}$	i^{th} descriptor of image j
$y_{i'}$	i'^{th} descriptor of the query image
$q(.)$	quantizer: $q(x_{i,j})$ is the quantized index associated with $x_{i,j}$
s^*_j	final score of dataset image j
$\delta_{x,y}$	Kronecker delta function: $\begin{cases} 1 & \text{if } x = y, \\ 0 & \text{otherwise.} \end{cases}$
$f(.,.)$	descriptor matching function, see (1)
$h(.,.)$	Hamming distance (9)
n_d	total number of descriptors ($= \sum_{j=1}^{n} m_j$)

Table 1: Notations.

such that the *tf-idf* weight associated with the visual word considered is applied to both the query and the dataset image in the BOF inner product. Using this new matching function, the image scores s_j become identical to the BOF similarity measure used in [4]. This voting scheme normalizes the number of votes by the number of descriptors of the database image (normalization by the L_1 norm of the visual word histogram of the database image). In what follows, we will use the L_2 normalization instead. For large vocabularies, the L_2 norm of a BOF is very close to the square root of the L_1 norm. In the context of a voting system, the division of the score by the L_2 norm is very similar to $s^*_j = s_j/\sqrt{m_j}$, which is a compromise between measuring the number and the rate of descriptor matches.

2.3 Weakness of quantization-based approaches

Image search based on BOF combines the advantages of local features and of efficient image comparison using inverted files. However, the quantizer significantly reduces the discriminative power of the local descriptors. Two descriptors are assumed to match if they are assigned the same quantization index, i.e., if they lie in the same Voronoi cell. Choosing the number of centroids k is a compromise between the quantization noise and the descriptor noise (due to changing imaging conditions).

Fig. 1(b) shows that a low value of k leads to large Voronoi cells: the probability that a noisy version of a descriptor belongs to the correct cell is high. However, this also reduces the discriminative power of the descriptor: different descriptors lie in the same cell. Conversely, a high value of k provides good precision for the descriptor, but the probability that a noisy version of the descriptor is assigned to the same cell is lower, as illustrated in Fig. 1(a). Moreover, for a flat visual vocabulary, the complexity of assigning the query descriptors is $\mathcal{O}(k \times d \times m'_l)$, hence the computing cost is significantly higher for larger vocabulary sizes.

Fig. 2 shows the impact of this trade-off when matching a pair of images. The matches obtained with a BOF quantization are analyzed. A coarse quantization clearly leads to many incorrect matches, as shown in Fig. 2(a). We can observe that many of the corresponding regions are quite different. Using a finer quantization, many incorrect matches are removed (see Fig. 2(b)), but at the same time many correct matches are also removed.

To evaluate quantitatively the approximate nearest neighbor search performed by BOF, Fig. 3 measures the trade-off between

- the average recall of the ground truth nearest neighbor (NN recall)
- and the average rate of vectors that are retrieved from the dataset.

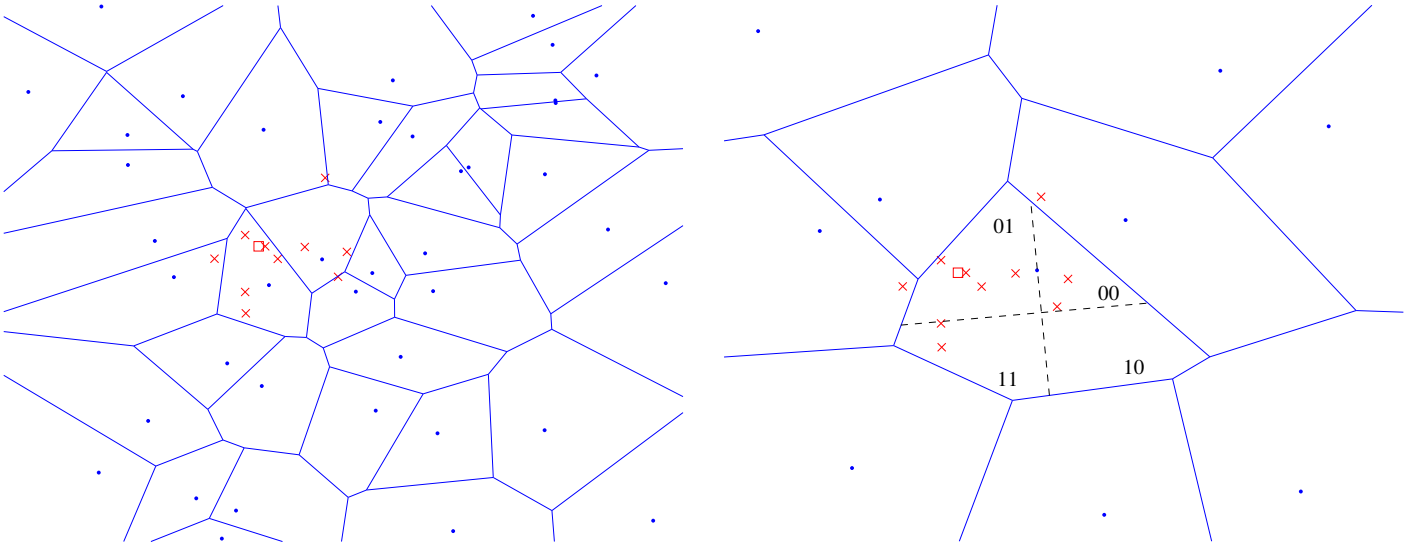
Clearly, a good approximate nearest neighbor search algorithm should retrieve the nearest neighbor with high probability and arbitrary vectors with low probability. In BOF, the trade-off between these two quantities is managed by the number k of clusters.

For the evaluation, we have used the approximate nearest neighbor evaluation set available at [16]. It has been generated using the affine covariant features program of [17]. A one million vector set to be searched and a test query set of 10000 vectors are provided. All these vectors have been extracted from the INRIA Holidays image dataset described in Section 7.

Fig. 3 shows the performance of BOF as an ANN search algorithm for this dataset. We can observe that the accuracy is good, for $k = 1000$, the NN recall is of 45% and the proportion of the retrieved dataset vectors is 0.1%.

One key advantage of BOF is that its memory usage is much lower than competing approximate nearest neighbor search algorithms. For instance, with 20 hash functions the memory usage of LSH [7] is 160 bytes per descriptor[1] com-

[1]For each hash function, we count 4 bytes for the descriptor identifier and 4 bytes for the hash value, see [8].



(a) (b)

Figure 1: Illustration of k-means clustering and our binary signature. (a) Fine quantization (high k). (b) Low k and binary signature: the similarity search within a Voronoi cell is based on the Hamming distance. Key: ·=centroid, □=descriptor, ×=noisy versions of this descriptor.

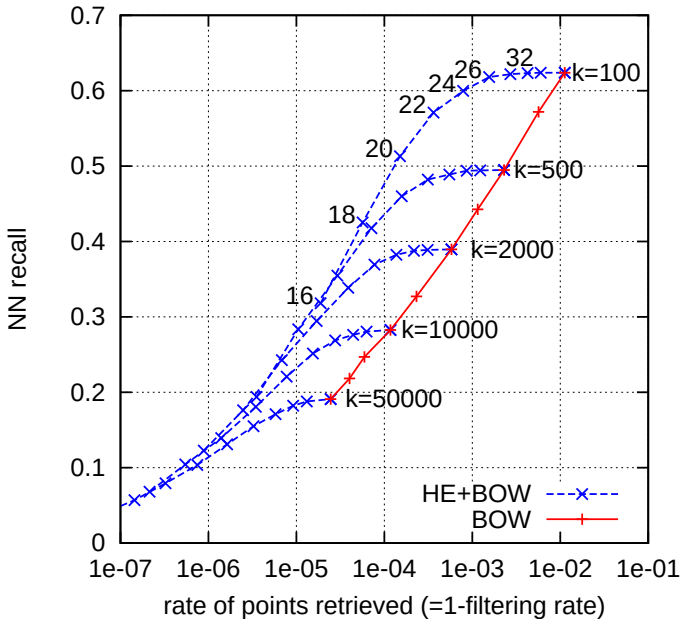


Figure 3: SIFT descriptors: approximate nearest neighbor search accuracy of BOF (dashed) and Hamming Embedding (plain) for different numbers k of centroids and Hamming thresholds h_t (for a 64-bit signature). This figure depicts the probability of retrieving the NN (NN recall) as a function of the fraction of vectors that are returned. For instance, by returning on average 0.1% of the vectors, i.e., 1000 vectors in this experiment, the probability of having the NN returned is about 0.45 with standard k-means and 0.60 with HE.

pared with the 4 bytes used in BOF to store the image identifier.

3 Hamming embedding of local descriptors

In this section, we present an approach which combines the advantages of a coarse quantizer (low number of centroids k) with those of a fine quantizer (high k). It consists in refining the quantized index $q(x_i)$ with a d_b-dimensional binary signature $b(x_i) = (b_1(x_i), \ldots, b_{d_b}(x_i))$ that encodes the location of the SIFT descriptor within the Voronoi cell, see Fig. 1(b). It is designed so that the Hamming distance

$$h(b(x), b(y)) = \sum_{i=1}^{d_b} |b_i(x) - b_i(y)| \qquad (9)$$

between two descriptors x and y lying in the same cell reflects the Euclidean distance $d(x, y)$: the Hamming distance h between a descriptor and its NNs in the Euclidean space is small. This mapping from the Euclidean space into the Hamming space, is referred to as Hamming Embedding (HE).

Note that this method is different from the Euclidean version of LSH (E2LSH) [7, 8], which produces several hash keys per descriptor. LSH assumes that two descriptors are similar if they have the same hash values for at least one hash function, i.e. if the descriptors lie in the *same* cell of one of the space partitioning. This corresponds to a distance having only two distinct values: same cell or not. In contrast, HE defines a single partitioning of the feature space and uses the Hamming metric between signatures in the embedded space to measure their similarity.

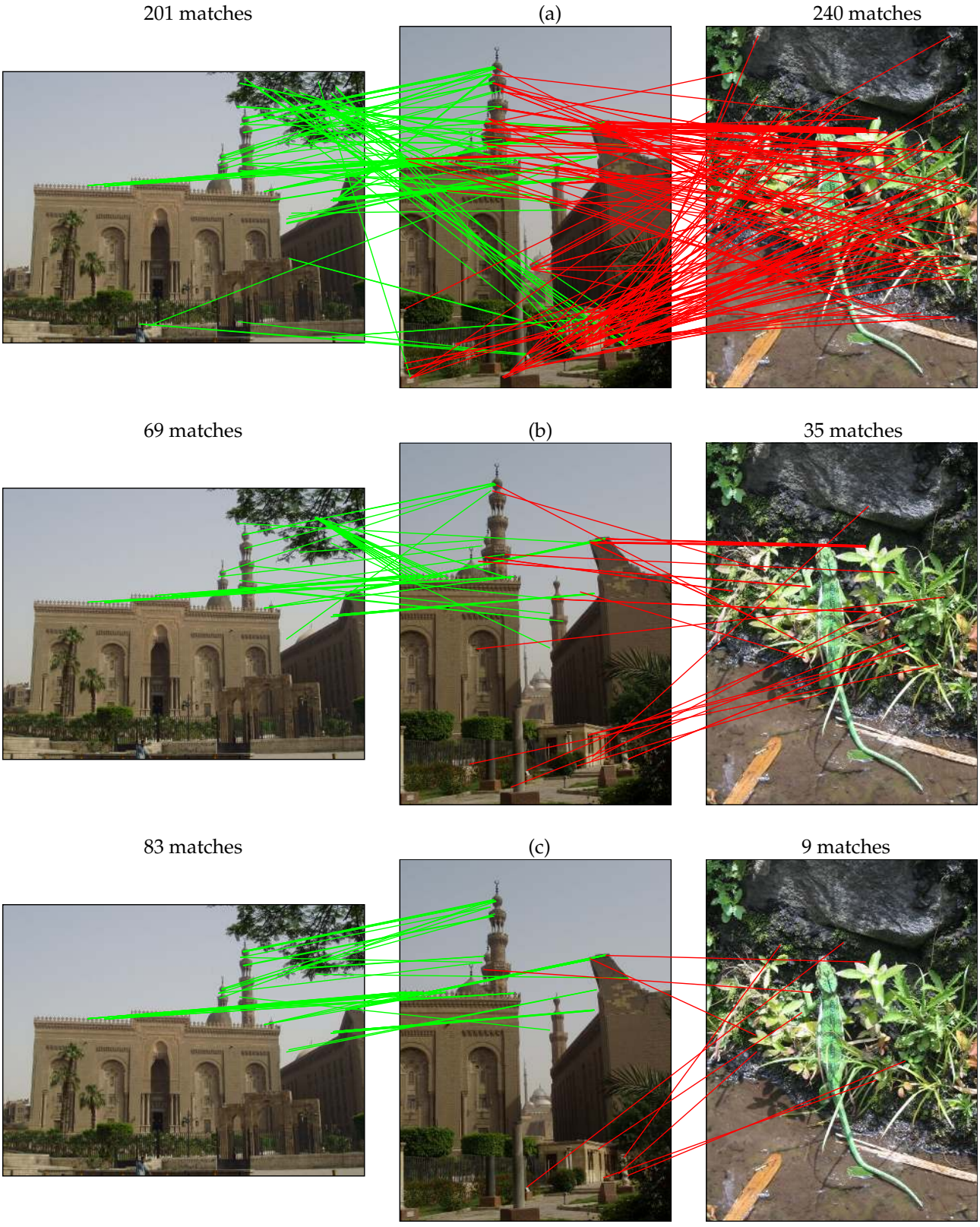


Figure 2: Matching the points of a query image (center) with a corresponding (left) and non-corresponding (right) image, for different quantizers : (a) Coarse quantization ($k = 20$k), (b) Fine quantization ($k = 200$k), (c) Coarse quantization with Hamming Embedding ($k = 20$k, $h_t = 24$).

3.1 Binary signature generation

In the following we present an approach for generating binary signatures for SIFT descriptors[2]. We first describe the *off-line* procedure that determines a projection matrix P and $k \times d_b$ median values (d_b values for each of the k clusters). To compute these median values, we use the same D_{train} dataset as for the k-means clustering, see Section 7. We then describe the assignment procedure for a descriptor.

The off-line learning of the parameters consists in three steps.

1. **Random matrix generation:** A $d_b \times d$ orthogonal projection matrix P is generated. We randomly draw a matrix of Gaussian values and apply a QR factorization to it. The first d_b rows of the orthogonal matrix obtained by this decomposition form the matrix P.

2. **Descriptor projection and assignment:** The descriptors x_i from the dataset D_{train} are assigned to their closest centroid $q(x_i)$ and projected by P to $(z_{i1}, ..., z_{id_b})$.

3. **Median values of projected descriptors:** For each centroid l and each projected component $h = 1, \ldots, d_b$, we compute the median value $\tau_{l,h}$ of the set $\{z_{ih} | q(x_i) = l\}$ that corresponds to the descriptors assigned to cell l.

The signature generation of a descriptor x proceeds as follows. The quantizer q, the projection matrix P and the $k \times d_b$ median values $\tau_{h,l}$ are used to perform the HE:

1. **Assign** x to its closest centroid, resulting in $q(x)$.

2. **Project** x using P, producing a vector $z = Px = [z_1, \ldots, z_{d_b}]^\top$.

3. **Compute the signature** $b(x) = (b_1(x), \ldots, b_{d_b}(x))$ as

$$b_i(x) = \begin{cases} 1 & \text{if } z_i > \tau_{q(x),i}, \\ 0 & \text{otherwise.} \end{cases} \quad (10)$$

At this point, a descriptor is represented by $q(x)$ and $b(x)$. We can now define the HE matching function as

$$f_{\text{HE}}(x, y) = \begin{cases} (\text{tf-idf}(q(x)))^2 & \text{if } q(x) = q(y) \\ & \text{and } h\left(b(x), b(y)\right) \leq h_t \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

where h is the Hamming distance defined in (9) and h_t is a fixed Hamming threshold such that $0 \leq h_t \leq d_b$. It has to be sufficiently high to ensure that the Euclidean NNs of x match, and sufficiently low to filter many points that lie in a distant region of the Voronoi cell.

[2]This approach has also been applied to GIST descriptors in [18].

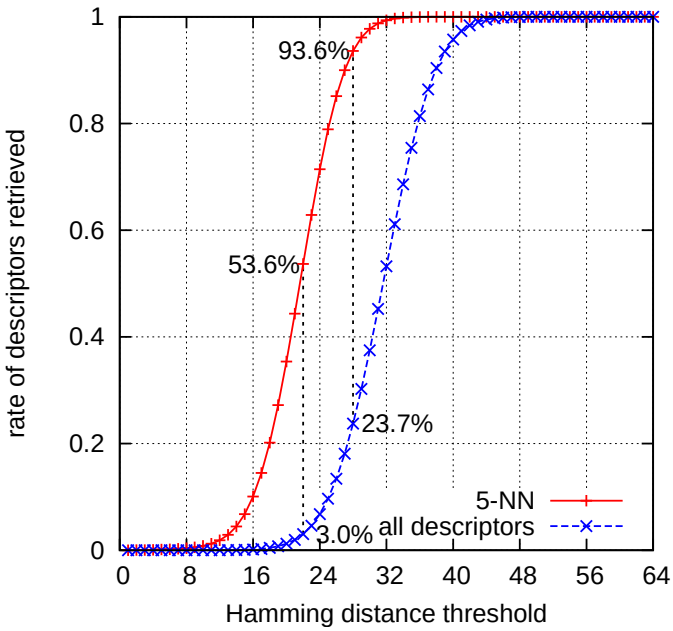


Figure 4: Rate of SIFT descriptors retrieved as a function of the Hamming distance threshold for (a) all descriptors in a cell and (b) the 5 NNs. The number of bits is $d_b = 64$.

Remarks:

- We use the same projection matrix P for all the visual words. For a 200k visual vocabulary and $d_b = 64$, storing a projection matrix per visual word would require about 6 GB of memory. Also, learning a projection per cell, for instance using principal component analysis (PCA), would require several times more learning data than simply adjusting the median values $\tau_{h,l}$. We typically use the same learning set as for the k-means clustering.

- We have evaluated the retrieval performance using a global PCA projection matrix for $d_b = 64$, instead of a random one. The PCA is performed on the matrix of all descriptor coordinates relative to the centroid they are assigned to. This method does not significantly improve the results.

- Our binary signature generation procedure is not optimal with respect to the trade-off between 1) memory usage and 2) preserving the neighborhood in the embedded Hamming space. The spectral hashing proposed in [19] specifically addresses this optimization. Note that this method uses a PCA. Therefore, learning the embedding function for each visual word leads to the same practical issue.

3.2 Evaluation of nearest neighbor search using HE

Fig. 4 and Fig. 5 illustrate the impact of the parameters on the quality of the approximate nearest neighbor search provided by HE for SIFT descriptors. There is a compromise between the filtering rate, i.e. returning a limited number of descriptors, and finding the nearest descriptors. These

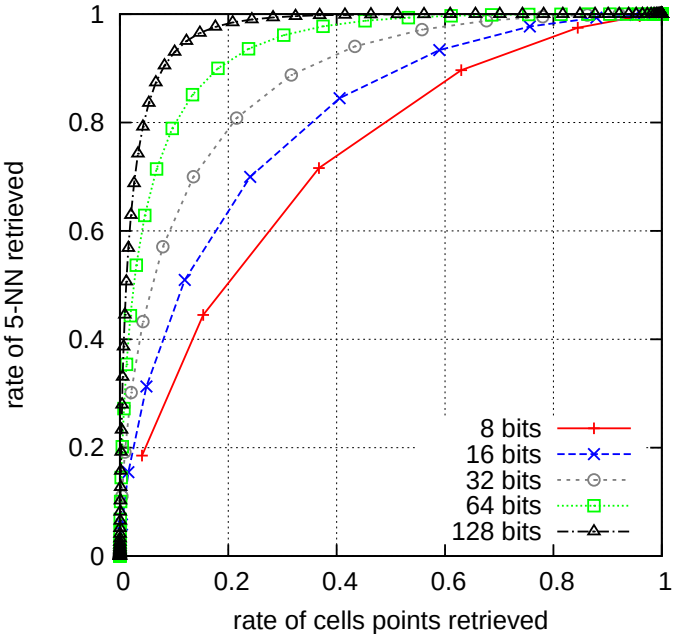


Figure 5: HE: Filtering effect on the SIFT descriptors within a cell and on the 5 NNs for different Hamming thresholds. The number of bits d_b of the binary signature varies between 8 and 128.

plots have been generated by analyzing a set of 1000 descriptors assigned to the same centroid. Given a descriptor x we compare the average rate of descriptors that are retrieved by the matching function to the rate of 5-NN that are retrieved.

Fig. 4 shows that choosing an appropriate threshold h_t (here between 20 and 28) ensures that most of the cell's descriptors are filtered out and that the descriptor's NNs are preserved with a high probability. For instance, setting $h_t = 22$ filters about 97% of the descriptors while preserving 53% of the 5-NN. A higher value $h_t = 28$ keeps 94% of the 5-NN and filters 77% of the cell descriptors. Fig. 5 represents this trade-off for different binary signature lengths. Clearly, the longer the binary signature d_b, the better the HE filtering quality. In the following, we have fixed $d_b = 64$, a good compromise between HE accuracy and memory usage (8 bytes per signature).

Fig. 6 shows the Hamming distance probability mass functions (PMF) obtained for corresponding and non-corresponding SIFT descriptors. It is averaged over all the Voronoi cells. The corresponding descriptors are defined as the descriptors 1) which, according to the ground-truth, correspond to matching images and 2) which have been geometrically verified[3]. These PMF have been obtained empirically using the full Holidays dataset (see Section 7). Here again, one can clearly see the impact of the signature length on the quality of the comparison. It is worth noticing that the PMF of the non-corresponding descriptors is close to the binomial distribution $\mathcal{B}(d_b, 0.5)$. This distribution corresponds to the case where the binary signatures are uniform on the Hamming hypercube, i.e., if all the bits have probability 0.5 and are independent.

A comparison with standard BOF shows that the approximate nearest neighbor search performed by HE is much better. This is qualitatively shown in Fig. 2(c), where one can observe that many matches have been removed without removing most correct ones. With HE, the query image has many more point matches with the relevant image than with the irrelevant one.

This is confirmed by the quantitative evaluation of Fig. 3. Using HE for the same number of vectors that are retrieved increases the probability that the NN is among these voting vectors.

3.3 Weighting the Hamming distance

In this section, we propose a weighting based on the Hamming distance, i.e., smaller distances result in higher matching scores. In the spirit of the *tf-idf* weighting scheme, the weight $w_d(a)$ associated with an observed distance $a = h(b(x), b(y))$ is obtained as the minus log-probability of having the Hamming distance between binary signatures lower than or equal to a. We assume that the PMF of the binary signatures is uniform on the Hamming hypercube $\{0,1\}^{d_b}$, which is motivated by the binomial form of the distances between non-corresponding descriptors in Fig. 6. The weights are then given by

$$w_d(a) = -\log_2 \left(\frac{1}{2^{d_b}} \sum_{i=0}^{a} \binom{i}{d_b} \right). \qquad (12)$$

These weights are stored in a look-up table of $d_b + 1$ elements, corresponding to all possible Hamming distances. They are used in combination with the *tf-idf* weight of (11). High Hamming distances, in particular those above $d_b/2$, have a very low impact on the score. We can therefore set them to zero, as done in (11) by using the threshold h_t. This improves the efficiency of the indexing structure.

4 Large-scale geometric consistency

Image search based on BOF ranks the database images without exploiting geometric information. Accuracy is improved by adding a *re-ranking* step [9] that computes a geometric transformation between the query and a short-list of database images returned by the BOF search. To obtain an efficient and robust estimation of this transformation, the model is often kept as simple as possible [1, 9]. In [1] an affine 2D transformation is estimated in two steps. First, a Hough scheme estimates a transformation with 4 degrees of freedom. Each pair of matching regions generates a set of parameters that "vote" in a 4D histogram. In a second step, the sets of matches from the largest bins are used to estimate a finer 2D affine transform. In [9] further efficiency is obtained by a simplified parameter estimation and an approximate local descriptor matching scheme.

Despite these optimizations, geometric matching algorithms are costly and cannot reasonably be applied to more than a few hundred images. In this section, we propose to

[3]This geometrical verification uses the exact Euclidean distance to compare the descriptors.

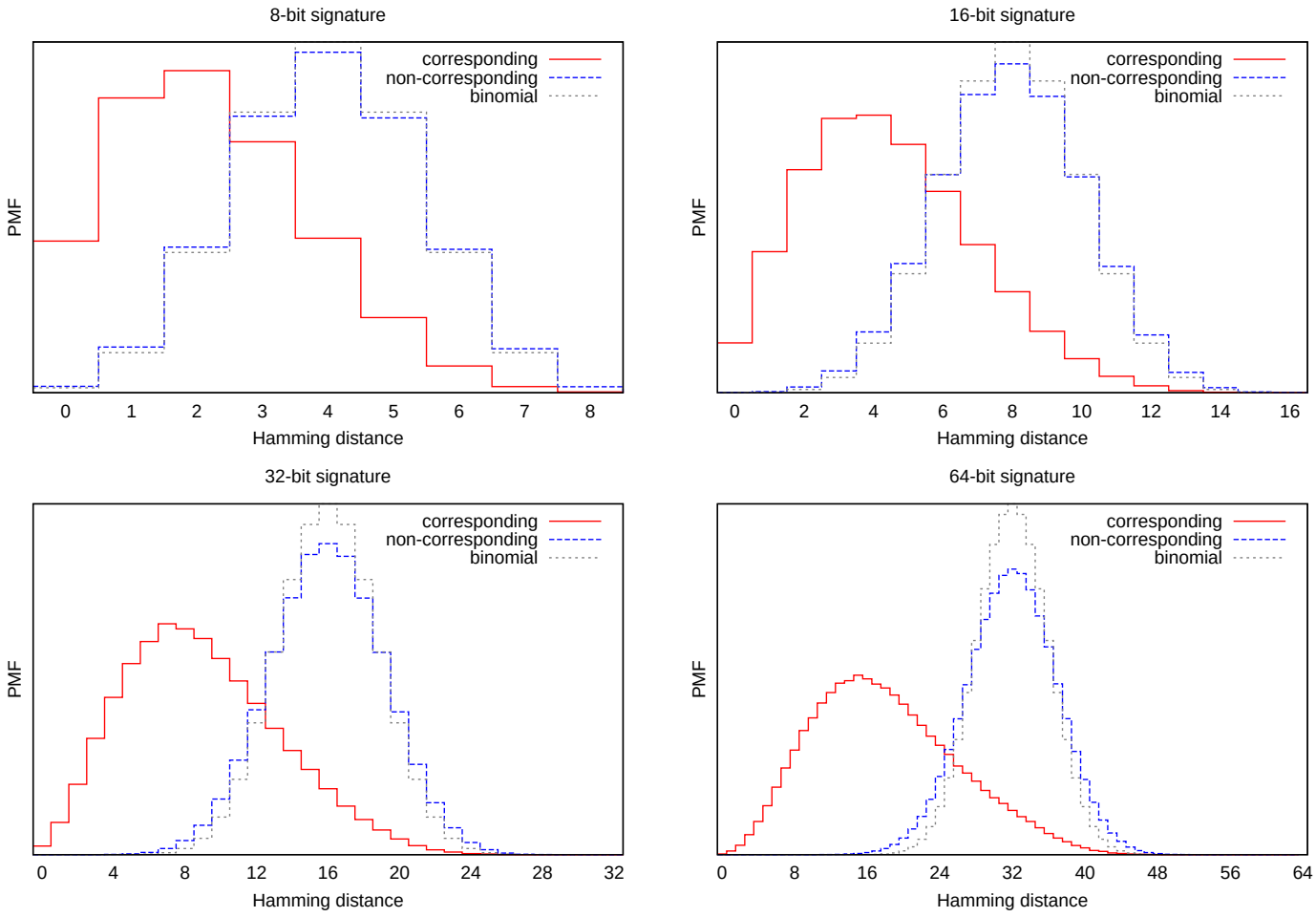


Figure 6: Holidays dataset: empirical probability mass function of the Hamming distances for corresponding and non-corresponding points, for several binary signature sizes. The corresponding points have been geometrically verified. The binomial distribution $\mathcal{B}(d_b, 0.5)$ is a theoretical model obtained by assuming the bits are uniform and independent.

exploit weak, i.e., partial, geometrical information without explicitly estimating a transformation mapping the points from one image to another. The method is integrated into the inverted file and can efficiently be applied to all images. Our weak geometric consistency constraints refine the voting score and make the description more discriminant.

Note that the *re-ranking* step can still be applied on a short-list to estimate the full geometric transformation. It is complementary to the weak consistency constraints and further improves the results (see Section 7.5).

4.1 Analysis of weak geometric information

In order to obtain orientation and scale invariance, region of interest detectors extract the dominant orientation of the region [1] and its characteristic scale [20]. This extraction is performed independently for each interest point. When an image undergoes a rotation or scale change, these quantities are consistently modified for all points, see Fig 7 for an illustration in the case of image rotations. It shows the difference in dominant orientations for pairs of matching regions. One can observe that only the incorrect matches are not consistent with the global image rotation.

Similarly, the characteristic scales of interest points are consistently scaled between two images of the same scene or object, as shown on Fig. 8.

4.2 Weak geometrical consistency

The key idea of our method is to verify the consistency of the angle and scale differences of the matching descriptors. We build upon and extend the BOF formalism of (1) by using *several* scores s_j per image. For a given image j, the entity s_j then represents the histogram of the angle and scale differences, computed from the characteristic angle and scale of the interest regions of corresponding descriptors. Although these two parameters are not sufficient to map the points from one image to another, they can be used to improve the image ranking. The update step of (1) is modified:

$$s_j(\delta_a, \delta_s) := s_j(\delta_a, \delta_s) + f(x_{i,j}, y_{i'}), \tag{13}$$

where δ_a and δ_s are the quantized angle and log-scale differences between the interest regions. The image score then becomes

$$s_j^* = g\left(\max_{(\delta_a,\delta_s)} s_j(\delta_a, \delta_s)\right). \tag{14}$$

The motivation behind the scores of (14) is to use angle and scale information to reduce the scores of images whose points are not transformed by consistent angles and scales. Conversely, a set of points consistently transformed will accumulate its votes in the same histogram bin, resulting in a high score.

Experimentally, the quantities δ_a and δ_s have the desirable property of being largely independent: computing separate histograms for angle and scale is as precise as computing the full 2D histogram of (13). Two histograms s_j^a and s_j^s are separately updated by

$$\begin{aligned} s_j^a(\delta_a) &:= s_j^a(\delta_a) + f(x_{i,j}, y_{i'}), \\ s_j^s(\delta_s) &:= s_j^s(\delta_s) + f(x_{i,j}, y_{i'}). \end{aligned} \tag{15}$$

The two histograms can be thought as marginal probabilities of the 2D histogram. Therefore, the final score

$$s_j^* = g\left(\min\left(\max_{\delta_a} s_j^a(\delta_a),\ \max_{\delta_s} s_j^s(\delta_s)\right)\right) \tag{16}$$

is a reasonable estimate of the maximum of (14). This approximation will be used in the following as it significantly reduces the memory requirements. In practice, the histograms are smoothed by a moving average to reduce the angle and log-scale quantization artifacts. Note that a more complex model (including translations) model could theoretically be included in WGC. However, for a large number of images, the number of parameters should be kept below 2, otherwise the memory and CPU costs of obtaining the scores would not be tractable.

4.3 Injecting a priori knowledge

Fig. 9(a) shows that the repartition of angle differences δ_a between matched descriptors is different for corresponding and non-corresponding point pairs. The shallow peaks on multiples of $\pi/2$ for non-corresponding points are due to the higher frequency of horizontal and vertical gradients in photos. The probability mass function of angle differences for corresponding points follows a highly non-uniform repartition. This is due to the human tendency to shoot either in "portrait" or "landscape" mode. A similar bias is observed for δ_s: image pairs with the same scale ($\delta_s = 0$) are more frequent.

The orientation and scale priors are used to weight the entries of our histograms before extracting their maxima. We have designed two different orientation priors (Fig. 9(b)): "same orientation" for image datasets known to be shot with the same orientation and "$\pm\pi/2$ rotation" for sets including non-straightened shots. On average, using priors improves the performance, as shown in the experimental section. Note however that images that underwent rare modification of orientation or scale are less likely to be correctly ranked when using transformation priors.

5 Descriptor quantization

In this section, we first analyze the complexity of querying the inverted file and introduce a cost factor representing the dictionary suboptimality. We then propose a strategy that increases the efficiency of the assignment of descriptors to visual words. Finally, we propose a multiple assignment strategy that improves the search accuracy at the cost of an increased query time.

5.1 Codebook construction and complexity

In contrast to the hierarchical method of [5] and to the method of [9], we use an exact brute-force k-means algorithm to generate the visual vocabulary. This is computationally expensive, but as this step is performed *off-line*, it has no impact at search time. Compared to [5], an exact k-means algorithm generates more balanced clusters, i.e., the lists in the inverted file have roughly the same length. This

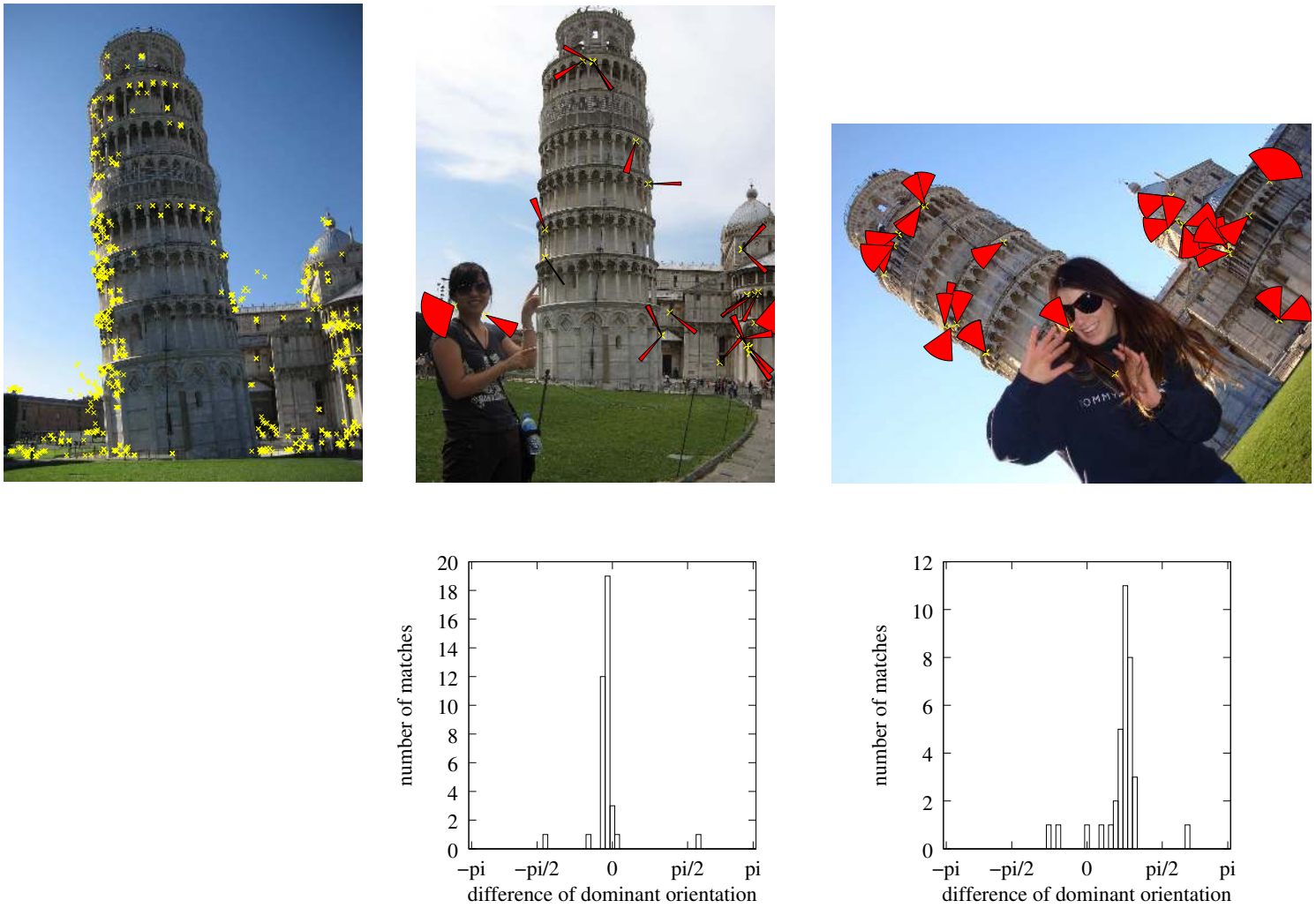


Figure 7: Orientation consistency. *Top-left:* Query image and its interest points. *Top-right:* two images of the same location viewed under different image rotations. The slices on each matched interest point show the difference in orientation between the interest point and the matching point on the query image. Matches are obtained with our HE method. *Bottom-right:* Histogram of the differences between the *dominant orientations* of matching points. The peak clearly corresponds to the global angle variation.

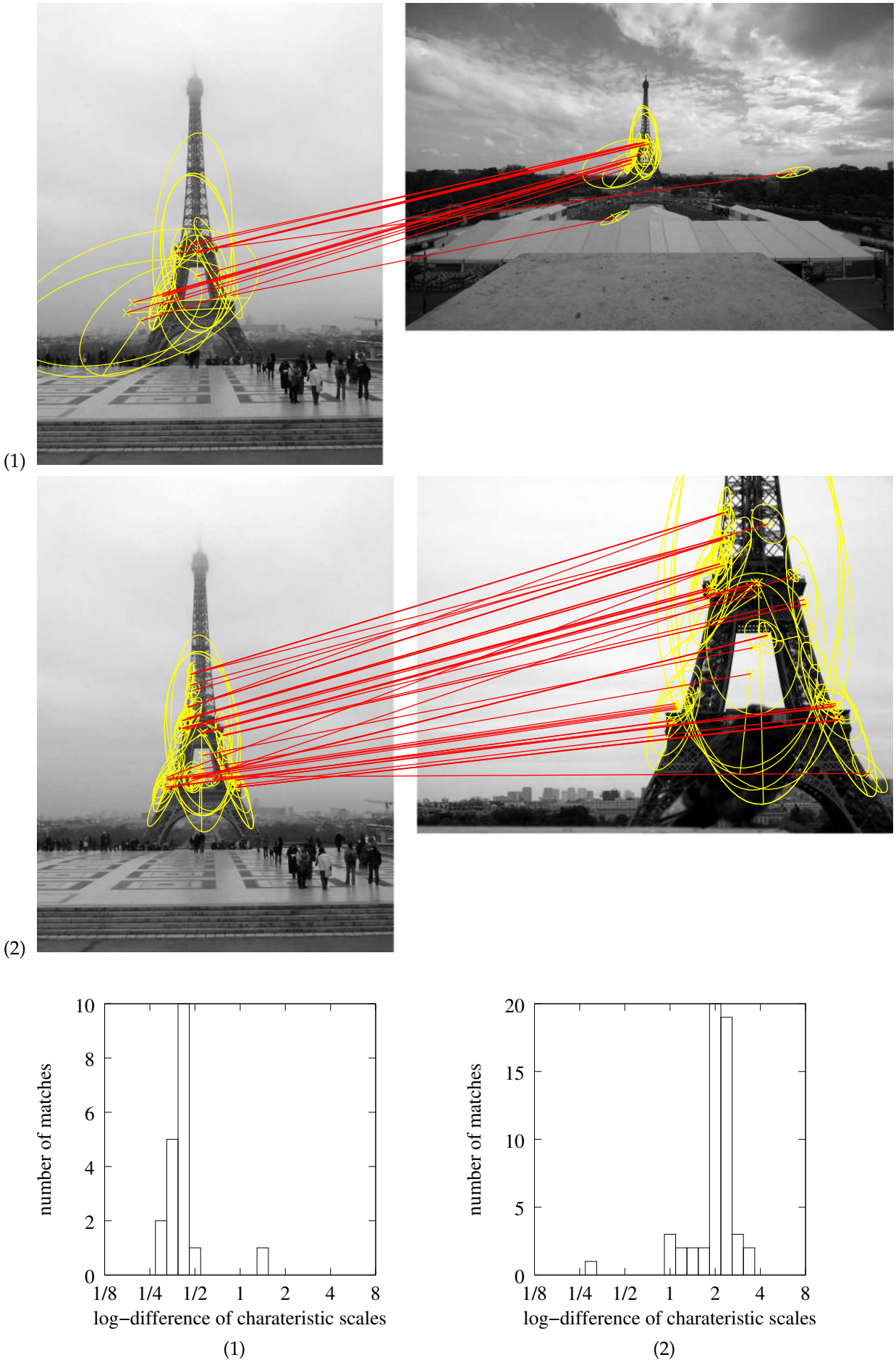


Figure 8: Scale consistency for two pairs of matching images. *Top two rows:* The matched interest point regions. *Bottom:* The corresponding histograms of log-scale differences between the characteristic scales of matched points. The peak clearly corresponds to the scale change between images.

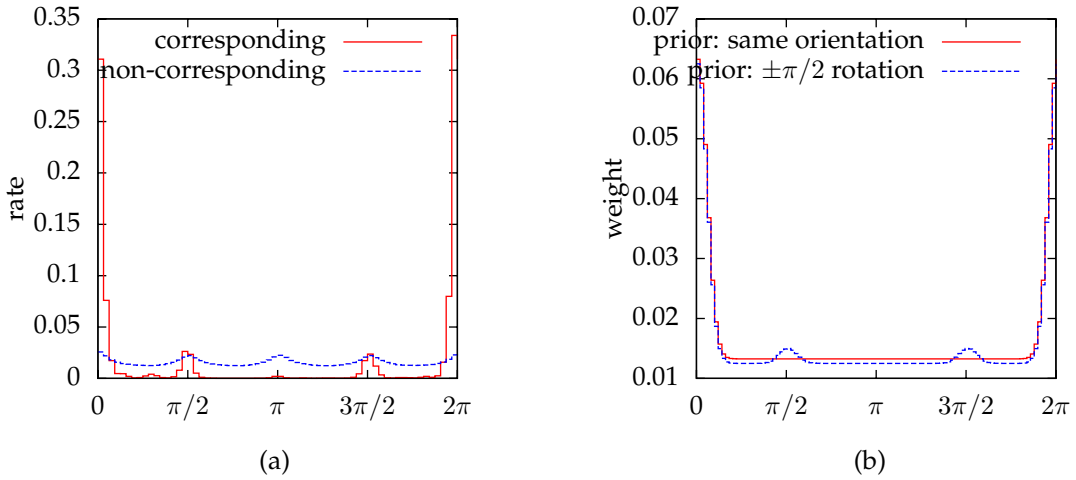


Figure 9: (a) Histogram of δ_a values accumulated over all query images of the Holidays dataset. Corresponding pairs are geometrically verified matching points between corresponding images. Non-corresponding pairs are HE-filtered point matches with non-corresponding images (from the Kentucky dataset). (b) Weighting function applied to the scores s_j^a.

results in a better efficiency when querying the inverted file, as the expected computing cost C associated with a single query descriptor is

$$C = n_d \sum_{i=1}^{k} p_i^2, \tag{17}$$

where n_d is the number of descriptors stored in the inverted file, and p_i denotes the probability that a given SIFT descriptor is assigned to the i^{th} visual word. The minimum of C is obtained when $p_i = 1/k$ for all visual words, i.e., when the inverted lists are of equal length. In that case, for a single input descriptor, the expected number of entries analyzed is equal to n_d/k.

The imbalance factor of a quantizer (visual vocabulary) is defined as

$$u = k \sum_{i=1}^{k} p_i^2, \tag{18}$$

which is ratio of the number of visited entries in the inverted file over the one of an optimal quantizer (minimal C). In other terms, for a given vocabulary size k, the imbalance factor is a measure of the query cost associated with a given visual word distribution.

Using k-means, the imbalance factors for vocabulary sizes of $k = 20000$ and $k = 200000$ are equal to 1.21 and 1.34, respectively. We have computed u from (18) by measuring the empirical probabilities p_i on one million images. Hierarchical clustering approaches, as proposed in [5, 11] leads to higher values: [11] reports factors between 4 and 5 for the hierarchical clustering.

5.2 Approximate visual word assignment

In order to efficiently assign descriptors to visual words with large vocabularies (e.g., for $k = 200000$ in our experiments), we use an approximate assignment scheme. It relies on a layered graph structure (Fig. 10) constructed as follows:

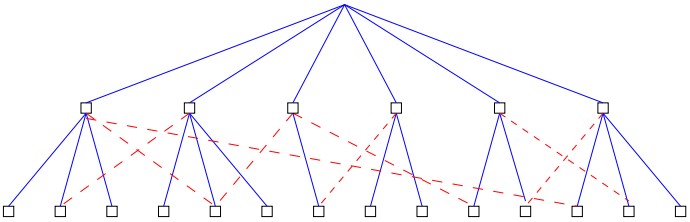


Figure 10: Two-layer graph structure used for the approximate assignment of visual words. *Plain:* The original connections between the two layers, as generated by the clustering. *Dashed:* Additional connections learned on an independent dataset.

Clustering: We compute a full k-means clustering on the training set to obtain a vocabulary of size k. The resulting centroids are the second-layer nodes of our structure.

Tree construction: a k-means clustering is performed *on the visual words*, producing k' centroids. These centroids form the first layer of our hierarchical structure. Each visual word of the original codebook is a leaf in the second layer, and is connected with its closest centroid in the first level.

Compared to the *top-down* approach of [5], the *bottom-up* construction of the tree is clearly more costly, as it requires to perform a k-means clustering for a large vocabulary. However, since this construction is performed offline, its efficiency is not critical. Using standard centroids as tree leaves preserves the k-means Voronoi cells, which minimize the reconstruction error between a descriptor and its visual word.

Graph construction: At this point, the tree structure can already be used to assign descriptors to visual words as in [5]. However, the nearest centroid of a descriptor in the second layer may not be connected to the nearest one in the first layer. In this case an assignment based on the tree structure does not find the visual word closest to the descriptor. The greedy N-best paths search strategy [12] addresses this

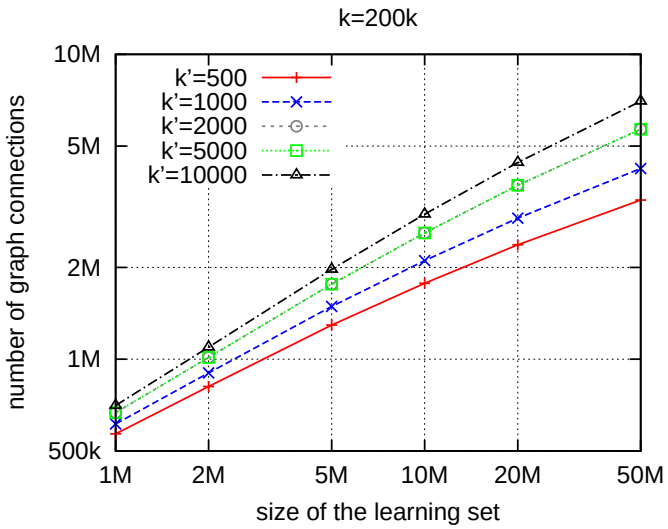


Figure 11: Number of connections in the graph structure, for several coarse quantizer sizes k', as the function of the size n' of the learning set. Note that the number of connections in the original tree structure is $k = 200$k.

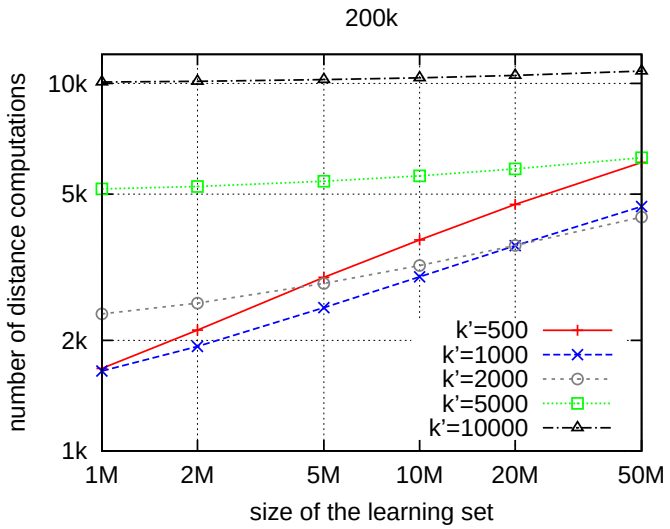


Figure 12: Graph-structured quantizer: number of vector distance computations per query. To be compared to that of a linear search, 200k for the visual vocabulary considered here.

issue by keeping several nodes in each level and by exploring their children, which may be of interest when using a large number of layers and a small branching factor.

Here, we complete the graph structure by connecting any first-layer node and leave that are the nearest neighbors of a descriptor. The connections are learned on a large dataset, by quantizing each descriptor with both quantizers, and by connecting the nodes of the resulting visual words. The number of connections is controlled by the size n' of the learning set. Fig. 11 shows the total number of connections generated between the first graph layer and the leaves for a vocabulary size of $k = 200$k for varying values of k' and n'.

To assign a descriptor to a visual word, we first search for the nearest neighbor in the first layer. We then search for the nearest neighbor in the leaves connected to it. Fig. 12 shows the average number of distance computations performed for a single query, as a function of the learning set size. This cost takes into account the two layers of the graph structure. For the proposed parameters, the number of distance computations is typically divided by 40 for a learning set of $n' = 50$M descriptors. The advantage of this method is that the structure is computed off-line and takes into account the statistics of the data. If the training set used for learning the connections is large enough, only connections that occur with a low probability are missed. This method is evaluated in the experimental subsection 7.3.

5.3 Multiple assignment

At query time it is also possible to assign a descriptor to not only one but *several* nearest visual words (using approximate visual word assignment for large k). Our strategy is similar to the multiple descriptor assignment proposed in [10] or the soft quantization method proposed in [13]. The method we propose hereafter is slightly different from [10] in that

- we perform multiple assignment for the query only, not for the images in the database. Therefore, the inverted file's memory usage is unchanged.
- the distance d_0 to the nearest centroid is used to filter centroids for which the distance to the descriptor is above αd_0 (typically, $\alpha = 1.2$). This criterion removes improbable matches and reduces the number of cells to explore.

Given these criteria, we assign on average each descriptor to 4 visual words, if the 10 nearest neighbors are considered. Hence, the query time is approximately multiplied by this value. This loss of efficiency is rewarded by a significantly higher accuracy, as shown in the experimental section 7.

6 Complexity

Both HE and WGC are integrated in the inverted file. This structure is usually implemented as an array that associates a list of entries with each visual word. Each entry contains a database image identifier and the number of descriptors of the image assigned to the visual word. The tf-idf weights and the BOF vector norms can be stored separately. The search consists in iterating over the entries corresponding to the visual words in the query image and in accumulating the scores accordingly.

We use an alternative implementation that consists in storing one entry per descriptor in the inverted list corresponding to the assigned visual word instead of one entry per image. This is required by HE and WGC, because additional information is stored per local descriptor. In our

Table 2: Inverted file memory usage.

		WGC	HE	WGC+HE
image id	21 bits	x	x	x
orientation	6 bits	x		x
log-scale	5 bits	x		x
binary signature	64 bits		x	x
total memory usage per entry:		4 bytes	11 bytes	12 bytes

Table 3: Query time per image for a one million-image dataset. Timings were measured in batch mode on a quad-core.

	$k = 20$k	$k = 200$k
compute descriptors	0.88 s	
quantization + binary signature	0.36 s	0.60 s
search, baseline BOF	2.74 s	0.62 s
search, WGC	10.19 s	2.11 s
search, HE	1.16 s	0.20 s
search, HE+WGC	1.82 s	0.65 s

experiments, the overall memory usage was not noticeably changed by this implementation.

HE impact on the complexity: For each inverted file entry, we compute the Hamming distance between the signature of the query descriptor and that of the database descriptor. This is done efficiently with a binary `xor` operation followed by a bit weight counter, which is efficiently implemented by combining 8-bit table lookups. Entries with a distance above h_t are rejected, which avoids the update of image scores for these entries. This is the case for most of the entries, as shown in Fig. 4.

WGC impact on the complexity: WGC modifies the score update by applying (15) instead of (1). Hence, two bins are updated, instead of one for a standard inverted file. With the tested parameters, see Table 2, the memory usage of the histogram scores is 127 (one per possible quantized difference, i.e., from -63 to +63) floating point values per image, which is small compared with the inverted lists.

Runtime: All experiments were carried out on 2.6 GHz quad-core computers. As the new inverted file contains more information, we carefully designed the size of the entries to fit into 12 bytes per point, as shown in Table 2.

Table 3 summarizes the average query time for a one million image dataset. We can observe that the baseline BOF approach is significantly faster for a larger visual vocabulary, i.e., for $k = 200$k. This is due to the high rate of zero components in the case of large visual vocabularies. Interestingly, HE reduces the inverted file query time compared to the baseline BOF approach for both values of k. This is due to the fact that the Hamming distance computation and thresholding is cheaper than updating the scores. WGC is costly, mostly because the histograms do not fit in cache memory and their memory access pattern is almost random. Overall, the search time of HE + WGC is comparable to the inverted file baseline BOF. Note that for $k = 200$k visual words, we use the approximate nearest neighbor search (section 5), i.e., the assignment is not ten times slower than for $k = 20$k, but increases by less than 2.

7 Experiments

We perform our experiments on three annotated datasets: our own *Holidays* dataset [21], the Oxford5k dataset and the University of Kentucky object recognition benchmark [5]. To evaluate large scale image search we also introduce a distractor dataset downloaded from Flickr. For evaluation we use mean average precision (mAP), as in [9], i.e., for each query image we obtain a precision/recall curve, compute its average precision and then take the mean value over the set of queries (it coincides with the area under the average precision curve). Descriptors are obtained by the Hessian-Affine detector and the SIFT descriptor, using the software of [17] with the default parameters. Clustering is performed with k-means on the independent Flickr60k dataset. The number of clusters is specified for each experiment.

Table 4: Datasets used in our experiments.

Dataset	#images	#queries	#descriptors
Holidays	1,491	500	4,455,091
Oxford5k	5,062	55	15,886,585
Kentucky	10,200	10,200	19,415,079
Flickr60k	67,714	N/A	140,211,550
Flickr1M	1,000,000	N/A	2,072,739,475

7.1 Datasets

In the following we present the different datasets used in our experiments, see Table 4 for an overview.

Holidays. We have collected a dataset which mainly contains personal holiday photos [21], but also images taken on purpose to test the robustness to various transformations: rotations, viewpoint and illumination changes, blurring, etc. The dataset includes a large variety of scene types (natural, man-made, water and fire effects, etc) and images are of high resolution. The dataset contains 500 image groups, each of which represents a distinct scene. The first image of each group is the query image and the correct retrieval results are the other images of the group. The dataset is available at [16]. It corresponds to a usage scenario in a personal photo management tool.

Oxford5k. The Oxford dataset [9] represents images of Oxford buildings. There are 55 query images corresponding to 11 distinct buildings. All the queries are defined by a rectangle delimiting the building and are in "upright" orientation. For each building, the 5062 images are annotated as relevant (good+OK), not relevant (bad), and should not be taken into account when measuring the accuracy (junk),

because they only contain a partial view (less than 25%) of the building. The usage scenario is a filter that re-ranks image results returned by a textual search.

Kentucky. This object recognition benchmark [5] contains 2550 different objects or scenes. Each one is represented by four images taken from four different viewpoints. For this dataset only, we give both the mAP and the measure of accuracy proposed by the authors, denoted by KS (Kentucky Score): it is the average number of relevant images ranked in top four positions when querying the dataset.

Flickr60k and **Flickr1M**. We have retrieved arbitrary images from Flickr and built two distinct sets: Flickr60k is used to learn the quantization centroids and the HE parameters (median values). For these tasks we have used respectively 5M and 140M randomly selected descriptors. They were also used to learn the graph structure of the approximate descriptor assignment introduced in Section 5.2 (for the 200K vocabulary only). Flickr1M are distracting images for large scale image search. Compared to *Holidays*, the Flickr datasets are slightly biased, because they include low-resolution images and more photos of humans.

For all our experiments, we have used a visual vocabulary learned on Flickr60k. Compared with learning the vocabulary on the query set, this choice is more representative of the behavior of the search in very large image datasets, for which 1) query descriptors represent a negligible part of the total number of descriptors, and 2) the number of visual words represents a negligible fraction of the total number of descriptors.

7.2 Evaluation of HE and WGC

Table 5 compares the proposed methods to the standard BOF baseline. One can observe that both HE and WGC result in significant improvements in terms of mAP. Furthermore, the combination of the two further increases the quality. Note that these results are obtained without spatial verification and query expansion. The parameters and variants have the following impact on the accuracy.

1. **The vocabulary size** of 200k visual words is significantly better than the 20k vocabulary for the BOF baseline. However, with HE the results are very similar for both sizes, as HE filters most incorrect matches.

2. **The WGC prior** significantly improves the performance for the Oxford dataset, as the images of this dataset are all upright. The prior penalizes matches that correspond to strong rotations. For the Kentucky benchmark, it is not useful, probably because many images have arbitrary angles.

3. **Weighting the Hamming distances** as proposed in the subsection 3.3 increases the mAP by 1% to 4% (absolute percentage points), depending on the dataset and the other parameters.

4. **The MA** of descriptors to visual words proposed in the subsection 5.3 significantly improves the mAP as well, providing on average an improvement of 4%. Interestingly, MA does not improve the performance of the standard BOF method.

5. **The impact of the threshold** h_t is shown in Fig. 13. One can observe a consistent behavior for all the datasets. The maximum mAP score is reached for h_t ranging from 20 to 26, depending on the vocabulary size and the dataset. Unsurprisingly, larger dictionary should be associated with a slightly lower value of h_t. Finally, weighting the distance is beneficial for all threshold values. The gain is especially important for higher values of the threshold. Note, however that the weights are not ultimately optimal, as the mAP is not a monotonously increasing function of the threshold.

The proposed methods and variants are complementary to each other. Except for the Kentucky benchmark, for which the WGC is not useful, the best accuracy is obtained by using all of them together. Note that the size of the vocabulary does not significantly influence the accuracy of the best method, but does results in a speed-up. Compared to the standard BOF, our method increases the mAP by 10% on the Kentucky recognition benchmark, by 23% on the Oxford dataset and by 24% on the INRIA Holidays dataset.

7.3 Graph-structured quantizer

Fig. 14 compares three ANN search algorithms:

- the baseline is an optimized implementation of the exhaustive linear search based on blocked matrix multiplications. This is faster than the baseline implementation used in [22] when several computations can be done in parallel. The performance of the other algorithms are measured as speedups over this baseline;

- the state-of-the-art FLANN algorithm [22]. We use the source code provided by the authors, which automatically tunes the parameters by cross-validation. We used various settings of the target precision to obtain different speed/accuracy operating points;

- our graph-structured ANN search algorithm proposed in Section 5.2. The method requires an independent learning set to learn the graph connections. The trade-off between speed and accuracy is obtained by varying the sizes of 1) the coarse quantizer (parameter k') and of 2) the learning set (denoted by n' in the figure).

One can observe that there is a trade-off between accuracy and speedup. For the 200k set, FLANN is better when a limited probability of correct assignment is sufficient, while our graph-structured algorithm offers a higher speedup for smaller sets, or when high nearest neighbor search accuracy is required. The performance strongly depends on the set size: for both algorithms, the speedup is limited for the small set, an observation already made in [22]. Our algorithm is significantly better than FLANN on the 200k vector set. However, the learning stage of FLANN is significantly faster than that of our graph-structured algorithm, as our method requires 1) to perform

Table 5: Results for the 3 datasets and for the different methods and variants: HE (see section 3) and distance weighting (3.3), WGC (section 4) without or with prior (4.3) and MA (5.3). For HE, the threshold h_t is set to 24 for all experiments.

	Kentucky				Oxford		Holidays	
	KS		mAP		mAP		mAP	
$k =$	20k	200k	20k	200k	20k	200k	20k	200k
BOF	2.88	2.95	0.752	0.771	0.338	0.384	0.469	0.572
HE	3.26	3.20	0.843	0.826	0.497	0.489	0.707	0.723
HE+weights	3.30	3.24	0.852	0.834	0.517	0.507	0.745	0.745
HE+weights + MA	**3.42**	3.38	**0.878**	0.868	0.549	0.561	0.735	0.775
WGC, no prior	2.95	2.93	0.771	0.764	0.391	0.404	0.600	0.612
WGC	2.93	3.00	0.768	0.781	0.445	0.462	0.647	0.688
HE+WGC+weights	3.27	3.23	0.845	0.834	0.562	0.545	0.770	0.761
HE+WGC+weights+MA	3.38	3.35	0.870	0.863	0.605	**0.615**	**0.813**	0.804

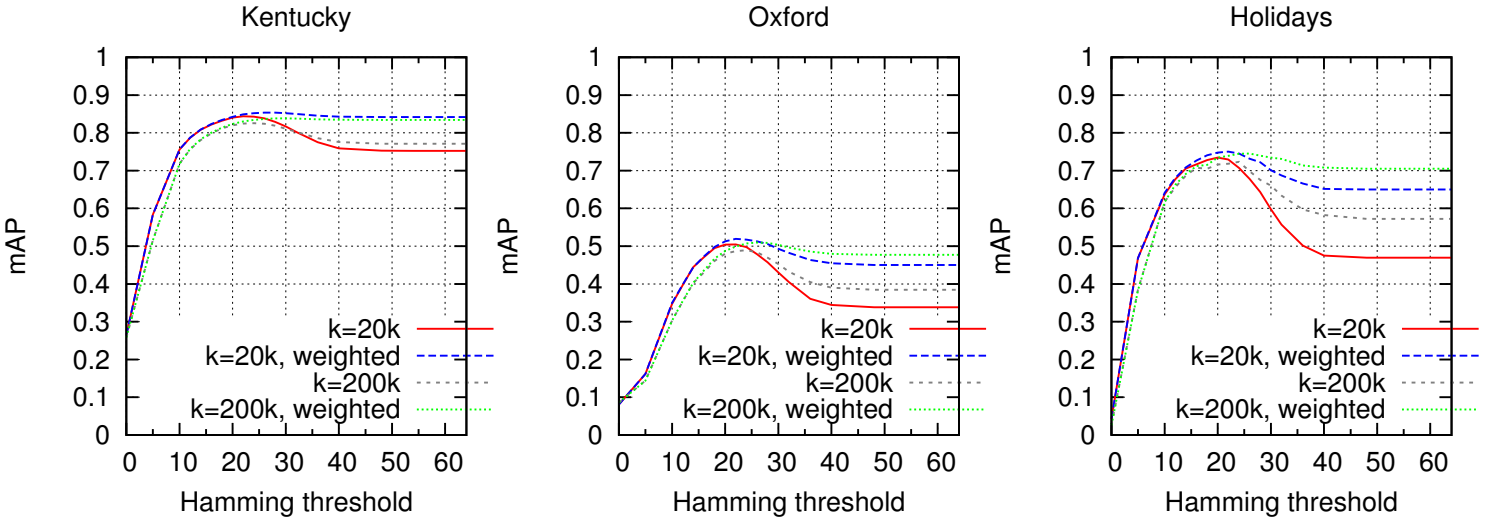


Figure 13: Hamming threshold h_t and distance weighting: impact on the search accuracy (SA, without WGC, without spatial verification)

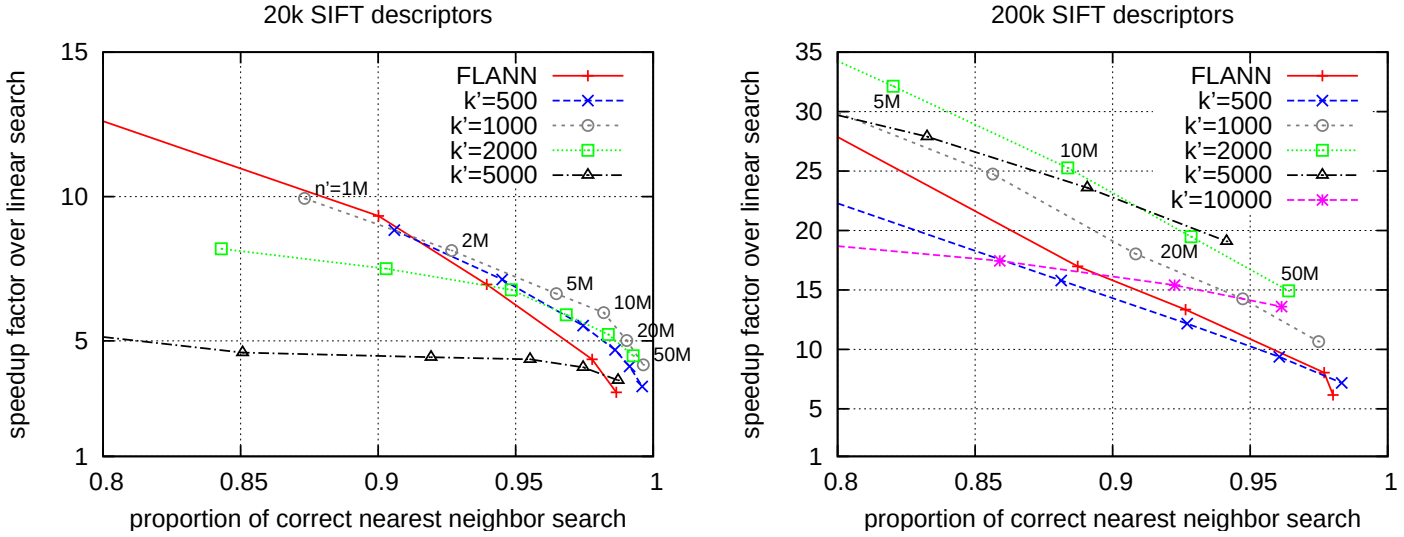


Figure 14: Speedup, over exhaustive distance computation, as a function of the obtained precision for our graph-structured quantizer and for the FLANN algorithm [22].

k-means clustering on the vector set and 2) to find the nearest neighbors of the two graph layers for a large training set of vectors.

Except for the evaluation performed in Figure 14, all our experimental results with the 200k vocabulary have been generated using $k' = 10000$ and a very large learning set. In this case, the probability of assigning a descriptor to its nearest centroid is 96.09% and grows to 99.87% when using our graph-structured quantizer jointly with MA (with the parameters given in subsection 5.3). In this setup, the impact on the image retrieval quality is negligible even in the single-assignment case. For example, the mAP obtained for the Holidays dataset with weighted HE is 0.74566 with approximate assignment against 0.74454 for exact nearest centroid assignment. The difference of 0.1% is not statistically significant.

7.4 Large scale experiments

Fig. 15 shows an evaluation of the different approaches for large datasets, i.e., we combined each dataset with a varying number of "distractor" images from the 1M Flickr dataset. For HE we have used the entropic weighting introduced in section 3.3 and set, again, the threshold h_t to 24. For WGC we used priors. We clearly see that the gain obtained with WGC + HE is very significant. For all the datasets, the mAP is better with our method on about one million images (evaluation dataset+Flickr) than the standard BOF without any distractor. This reflects the very good behavior of our scheme on a large scale.

Results for various queries are presented in Fig. 16. One can observe that the scenes returned are taken from very different viewpoints and orientations. The last three rows show that some images from the *Flickr1M* dataset (marked as FFP, *false* false positive) are actually relevant to the query image. They are counted as false positives and artificially decrease the results in terms of mAP given in Fig. 15. Fig. 17 shows for examples for which HE and WGC improve the quality of the ranking significantly.

7.5 Re-ranking and query expansion

In this subsection, we evaluated two successful state-of-the-art post-processing methods, re-ranking [1, 9] and query expansion [23], jointly with our approach. They are applied only to a fraction of the images due to complexity reasons. Therefore, the quality of the final results depends significantly on the results of the initial search system.

Geometrical re-ranking verifies the global geometrical consistency between matches [1, 9] for a short-list of database images returned by the image search system. Here we implement the approach of [1] and apply it to a short-list of 200 images.

We first obtain a set of matches, i.e., each descriptor of the query image is matched to the 10 closest ones in all the short-list images. We then estimate an affine 2D transformation in two steps. First, a Hough scheme estimates a transformation with 4 degrees of freedom. Each pair of matching regions generates a set of parameters that "vote" in a 4D histogram. In a second step, the sets of matches from the largest bins are used to estimate a finer 2D affine transform. The images for which the geometrical estimation succeeds are returned in first positions and ranked with a score based on the number of inliers. The images for which the estimation failed are appended to the geometrically matched ones, with their order unchanged.

Fig. 15 shows the results obtained with a short-list of 200 images. The further improvement confirms that this stage is complementary to WGC.

Table 6: Evaluation of the two methods for query expansion when combined with HE, WGC and MA. The Oxford building dataset is combined with distractors from Flickr1M. The vocabulary size is 20k.

	Oxford+				Holidays
re-ranking method:	0	10k	100k	1M	
geometric verification	0.667	0.652	0.591	0.486	0.848
TCE	0.757	0.735	0.674	0.582	0.827
AQE	0.747	0.736	0.687	0.572	0.842

Query expansion. Images with a large number of geometrically consistent matches are reliable. Therefore, they can be re-used as new queries that "expand" the original query [23]. The results of these "expanded" queries are considered relevant to the initial query. We adapted two query expansion methods from [23]:

- *Transitive closure expansion (TCE)* considers the tree of images with the initial query being its root. The children of a node are images that reliably match with it. TCE consists of a breadth-first scan of the tree, where nodes are returned as results in the order they are visited. The number of expansions is limited to 20 to avoid drift.
- *Additive query expansion (AQE).* The interest points of reliable results to the initial query are geometrically remapped to this image. The resulting set of points is used to perform a second query. The returned images are appended to results of the initial query. Only one re-querying is performed. This method is similar to the average query expansion of [23].

Table 6 shows results of the two query expansion techniques. In contrast to [23], our AQE method is not necessarily more accurate than the TCE method. We can observe a performance gain of around 10% on the Oxford dataset. Unsurprisingly, query expansion does not improve the results on the Holidays dataset, which contains only a few images of the same object.

Note that both re-ranking and query expansion are quite slow. Per query, re-ranking takes 18 seconds and query expansion costs 78 seconds for TCE and 48 seconds for AQE (to be compared with the timings of table 3). The memory usage does not change: these methods are only performed on a fraction of the dataset.

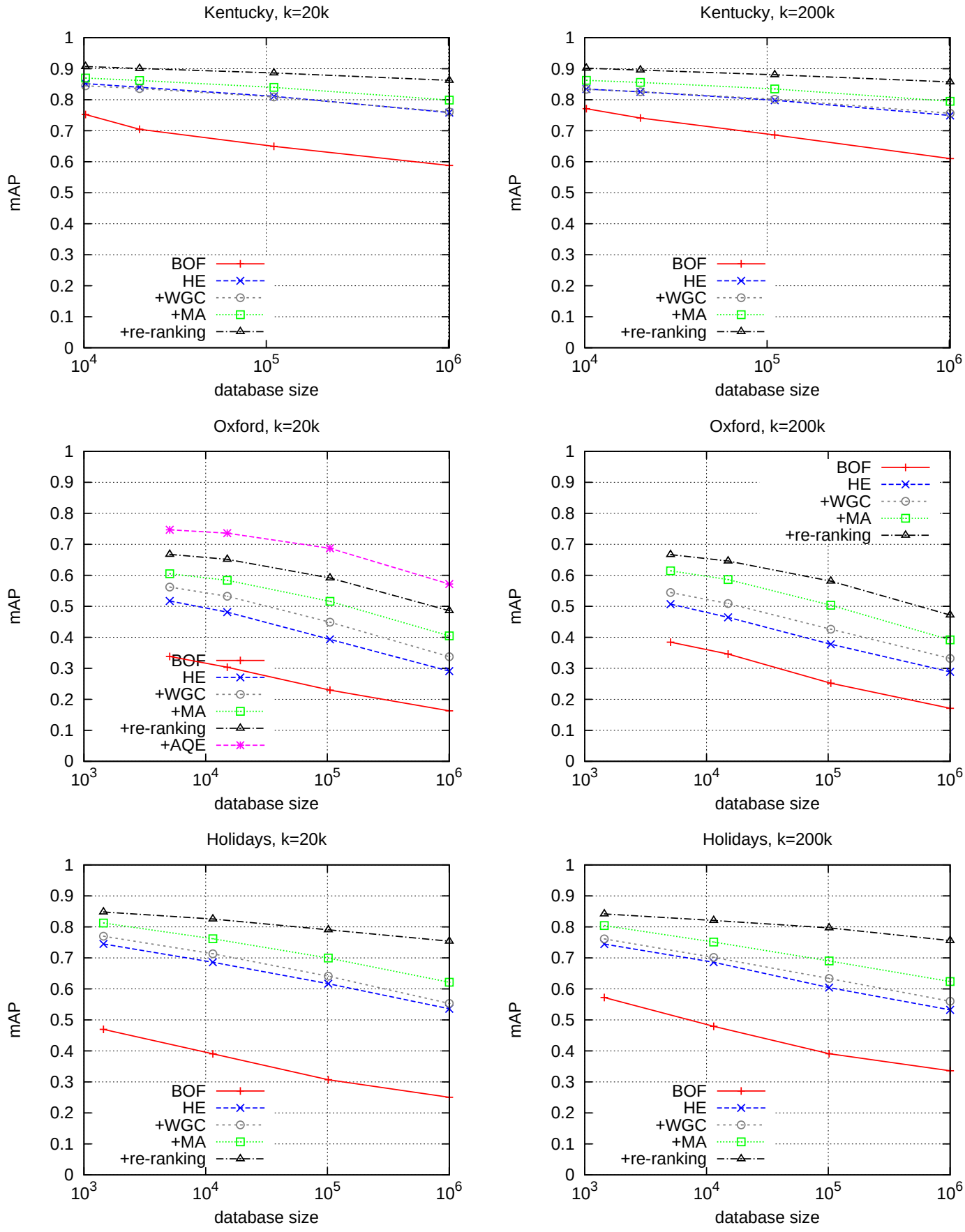


Figure 15: Performance of the image search as a function of the dataset size for various methods. The three reference datasets (*Kentucky*, *Oxford* and *Holidays*) have been merged with a varying number of distractors from *Flickr1M*.

7.6 Comparison with the state-of-the-art

In this subsection we compare to the state-of-the art on the Oxford and Kentucky datasets. The Holidays dataset has been introduced in the preliminary version of this paper [21] and results with HE+WCG have been improved here by adding a weighting, MA and query expansion. Table 7 compares our results with state-of-the-art methods. All the results presented have been obtained for a vocabulary learned on an independent dataset.

Oxford: Without post-processing, our best result on the Oxford Building dataset is 0.615, which is significantly better than the state-of-the-art [13] of 0.493 in a comparable setup, i.e., the visual vocabulary is learned on a set of Paris images. On Oxford combined with 100,000 distractors we obtain better results before re-ranking than [13] on a similar setup referred to by D1+D2 in their paper: 0.516 *vs* 0.343.

With geometrical re-ranking, our results are also significantly better on the Oxford dataset (respectively Oxford with 100,000 distractors) than [13]: 0.667 *vs* 0.598 (respectively 0.591 *vs* 0.480). Finally, when using query expansion, [13] reports 0.718 and 0.605 without and with 100K distractors, respectively. In a similar setup, we obtain 0.747 and 0.687, respectively.

In all cases, with or without post-processing, the gain due to our method is more significant for a large dataset, here 100k. This is probably due to the precision of our descriptor matching, i.e., the larger the number of descriptors the more important is the matching based on the inverted file. Note that a comparison with one million descriptors is not possible, as no results have been reported in the literature.

Kentucky: The comparison is performed with initial results obtained by the authors of the dataset [5] and our previous work [10], which to our knowledge is the state-of-the-art for this benchmark. In a similar setup, our new method obtains a lower mAP value on the benchmark itself. However, on a large scale we obtain a better mAP value of 3.10 (against 2.93), probably because an approximate strategy was required at this scale in the method of [10].

8 Conclusion

This article has introduced several ways of improving a standard bag-of-features representation. The first one is based on a Hamming embedding which provides binary signatures that refine visual words. It results in a similarity measure for descriptors assigned to the same visual word. The second is a method that enforces geometrical consistency constraints and uses a priori knowledge on the rotation and scaling transformations. The constraints are integrated within the inverted file and are applied to all the database images. Both these methods improve the performance significantly, especially for large datasets. Interestingly, these modifications do not result in an increase of the runtime. We have then proposed a graph-structured quantizer to improve the efficiency of the assignment of SIFT descriptors to visual words. This quantizer is shown to be competitive compared to those of the state of the art when high assignment accuracy is required.

acknowledgements

We would like to acknowledge the ANR projects GAIA and RAFFUT as well as the search engine project QUAERO for their financial support.

References

[1] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[2] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *International journal of computer vision*, vol. 60, no. 1, pp. 63–86, 2004.

[3] J. Matas, O. Chum, U. Martin, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *British machine vision conference*, 2002, pp. 384–393.

[4] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *International conference on computer vision*, 2003, pp. 1470–1477.

[5] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *Conference on computer vision and pattern recognition*, 2006, pp. 2161–2168.

[6] D. Omercevic, O. Drbohlav, and A. Leonardis, "High-dimensional feature matching: employing the concept of meaningful nearest neighbors," in *International conference on computer vision*, 2007.

[7] M. Datar, N. Immorlica, P. Indyk, and V. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Symposium on computational geometry*, 2004, pp. 253–262.

[8] A. Andoni, M. Datar, N. Immorlica, P. Indyk, and V. Mirrokni, *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice*. MIT Press, Mar 2006.

[9] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Conference on computer vision and pattern recognition*, 2007.

[10] H. Jégou, H. Harzallah, and C. Schmid, "A contextual dissimilarity measure for accurate and efficient image search," in *Conference on computer vision and pattern recognition*, 2007.

[11] F. Fraundorfer, H. Stewénius, and D. Nistér, "A binning scheme for fast hard drive based image search," in *Conference on computer vision and pattern recognition*, Jun 2007.

Table 7: Comparison of our best results with the state-of-the-art, using the same accuracy measures: mAP for Oxford and KS for the Kentucky benchmark.

dataset	Oxford		Kentucky	
distractors	0	100K	0	1M
soft assignment [13]	0.493	0.343		
ours	0.615	0.516		
soft + geometrical re-ranking [13]	0.598	0.480		
ours + geometrical re-ranking	0.667	0.591		
soft + query expansion [13]	0.718	0.605		
ours + query expansion	0.747	0.687		
hierarchical vocabulary [5]			3.19	
CDM [10]			3.61	2.93
ours			3.42	3.10
ours + geometrical re-ranking			3.55	3.40

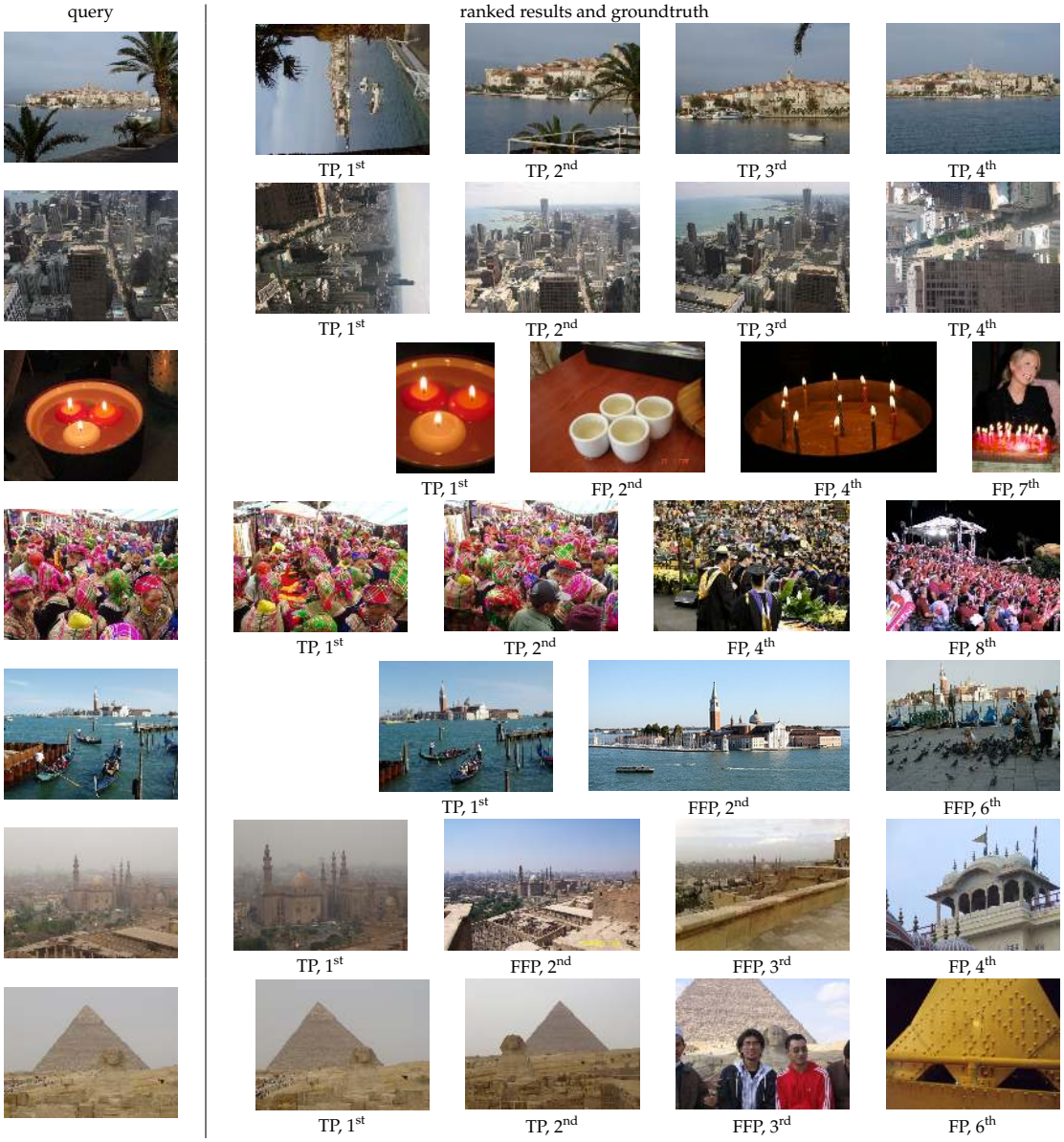


Figure 16: Queries from the *Holidays* dataset and some corresponding results for *Holidays*+1M distractors from Flickr1M. True positives are marked by TP and false positives by FP. As the *Holidays* dataset includes pictures of popular tourist attractions, matches were also found in the distractor dataset. They count as false positives and are marked by FFP (false false positive).

Figure 17: Two queries from the *Holidays* dataset and the ranks obtained with different methods (BOF, WGC, HE, HE+WGC, re-ranked) for two true positives. The database is Holidays + 1M. Note that in the first row the "easiest" true positive is not shown.

[12] G. Schindler, M. Brown, and R. Szeliski, "City-scale location recognition," in *Conference on computer vision and pattern recognition*, Jun 2007.

[13] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *Conference on computer vision and pattern recognition*, 2008.

[14] A. Torralba, R. Fergus, and Y. Weiss, "Small codes and large databases for recognition," in *Conference on computer vision and pattern recognition*, 2008.

[15] A. Oliva and A. Torralba, "Modeling the shape of the scene: a holistic representation of the spatial envelope," *International journal of computer vision*, vol. 42, no. 3, pp. 145–175, 2001.

[16] H. Jégou and M. Douze, "INRIA Holidays dataset," http://lear.inrialpes.fr/people/jegou/data.php, 2008.

[17] K. Mikolajczyk, "Binaries for affine covariant region descriptors," http://www.robots.ox.ac.uk/ vgg/research/affine/, 2007.

[18] M. Douze, H. Jégou, H. Singh, L. Amsaleg, and C. Schmid, "Evaluation of gist descriptors for web-scale image search," in *Conference on image and video retrieval*, july 2009.

[19] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Advances in neural information processing systems*, 2009.

[20] T. Lindeberg, "Feature detection with automatic scale selection," *International journal of computer vision*, vol. 30, no. 2, pp. 77–116, 1998.

[21] H. Jégou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *European conference on computer vision*, Oct 2008.

[22] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *International conference on computer vision and applications*, 2009.

[23] 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 *International conference on computer vision*, 2007.