Int J Multimed Info Retr (2013) 2:243–259
DOI 10.1007/s13735-013-0040-x

REGULAR PAPER

Bundle min-Hashing

Speeded-up object retrieval

Stefan Romberg · Rainer Lienhart

Received: 19 June 2013 / Revised: 23 July 2013 / Accepted: 5 August 2013 / Published online: 4 September 2013

Abstract We present a feature bundling technique based on min-Hashing. Individual local features are aggregated with features from their spatial neighborhood into bundles. These bundles carry more visual information than single visual words. The recognition of logos in novel images is then performed by querying a database of reference images. We further present a WGC-constrained RANSAC and a technique that boosts recall for object retrieval by synthesizing images from the original query image or reference images. We demonstrate the benefits of these techniques for both small object retrieval and logo recognition. Our logo recognition system clearly outperforms the current state-of-the-art with a recall of 83 % at a precision of 99 %.

Keywords Feature bundling · Object retrieval · Min-hash · RANSAC · Query expansion · Logo recognition

1 Introduction

In computer vision, the bag-of-visual words approach has been very popular in the last decade. It describes an image by multiple local features; high-dimensional descriptor vectors are quantized to single numbers—the visual words. An image is modeled as an unordered collection of word occurrences, commonly known as bag-of-words. The benefits of this description are an enormous data reduction compared to the original descriptors, a fixed-size image description, robustness to occlusion and viewpoint changes, and eventually simplicity and small computational complexity.

It has been observed several times that the retrieval performance of bag-of-words based methods improves much more by reducing the number of mismatching visual words than by reducing quantization artifacts. Examples are large vocabularies or Hamming Embedding [8]. In other words, the precision of the visual description seems to be more important than its recall, because low recall may be recovered by doing a second retrieval round, i.e. by query expansion.

Inspired by this observation we present a feature bundling technique [22] that builds on visual words. It does not describe each visual word individually but rather aggregates spatial neighboring visual words into feature bundles. An efficient search technique for such bundles based on min-hashing (mH) allows for similarity search without requiring exact matches. Compared to individual visual words such bundles carry more information, i.e. fewer false positives are retrieved. Thus, the returned result set is much smaller and cleaner. Our logo recognition framework [23] exploits this bundle representation to retrieve approximately 100 times fewer images than bag-of-words while having equal performance in terms of mAP. This technique turns the multi-class recognition of brand logos into simple lookups in hash tables.

This work is an extension of our work in [22,23]. We summarize our contributions as follows:

- We discuss and evaluate a new retrieval technique based on feature bundles and extensively compare its performance to existing approaches.
- A RANSAC variant for fast spatial re-ranking is described that yields superior results compared to existing approaches by exploiting a weak-geometric constraint to speed up the computation.

S. Romberg (✉) · R. Lienhart
Multimedia Computing and Computer Vision Lab,
Augsburg University, Universitätsstr. 6a, 86159 Augsburg, Germany
e-mail: stefan.romberg@informatik.uni-augsburg.de

R. Lienhart
e-mail: lienhart@informatik.uni-augsburg.de

- We demonstrate that recall of a system targeting high precision for object retrieval can be increased easily by exploiting synthetically generated images with query expansion and database augmentation.
- Eventually, the former techniques then are exploited for scalable logo recognition and combined in a system that significantly outperforms the current state-of-the-art.

2 Related work

We present related work suited to image and object retrieval and also briefly highlight the related work relevant in the context of min-Hashing.

Visual words and bundling In contrast to this work most approaches to feature bundling are indeed post-retrieval verification steps where the internal geometry of a bundle is used to discard false correspondences. An early approach in this spirit exploited the number of matching neighboring features to discriminate true feature matches from random matches [25]. Later, it was proposed to bundle multiple SIFT features that lie in the same MSER region into a single description [26]. However, this work uses individual visual words for retrieving candidate images, the bundles are only used together with a weak geometric similarity criterion for post-retrieval verification. In [4], the most informative projections that map the visual words from the 2-D space into feature histograms (termed "spatial bag-of-words") are learned. A similar approach, yet more unbiased to certain image layouts, splits the original feature histograms by random projections into multiple smaller "mini bag-of-features" [10]. Separate lookups and an aggregating scoring are used to find the most similar images in an image database. In [28], descriptive visual phrases are mined by analyzing the local neighborhood of local feature yielding a more discriminative visual description than single visual words.

Logo retrieval There has been much previous work on logo retrieval varying from recognition of brand logos on scanned documents to the search in real-world images. We focus on the latter case and retrieval schemes only. In [3], correspondences between SIFT descriptors of video frames and reference image are determined in order to detect whether a logo is present. The logo is then further localized by estimating the center of all the matches. In [13], feature triples are derived from a multi-scale Delaunay triangulation of interest points that yields a highly distinctive signature. A similar approach bundles triples of visual words including their spatial layout into visual signatures that are then subsequently indexed by a cascaded index making efficient testing of images for the presence of pairs and triples feasible [24]. In [12], the authors propose an adaptive RANSAC thresholding mechanism that controls the number of false positives and show that this improves the post-retrieval verification for many queries. In [7], the authors combine three types of local features to capture gradient distribution, shape and patch appearance and adaptively weight their combination during retrieval. Another approach based on feature bundling uses a regular grid to bundle local features that reside in the same grid cell [11]. Similar to Partition min-Hashing [15], each grid cell is described by a bag-of-words but the lookup for matching grid-cells is done by bag-of-words retrieval followed by a branch-and-bound object localization. In [16] feature selection is performed to determine consistent visual words. This allows to reduce the number of visual words used to query the inverted index with no or little loss of accuracy.

Min-Hashing (mH) Min-Hashing is a locality-sensitive hashing technique that is suitable for approximate similarity search of sparse sets. Originally developed for detection of duplicate text documents, it was adopted for near-duplicate image detection and extended to the approximation of weighted set overlap as well as histogram intersection [6]. Here, an image is modeled as a sparse set of visual word occurrences. Min-Hashing then allows to perform a nearest-neighbor search among all such sparse sets within an image database. This approach is described more extensively in Sect. 3.1.

Geometric min-Hashing (GmH) A conceptually similar approach to ours is Geometric min-Hashing [5]. However, its statistical preconditions for the hashing of sparse sets are totally different to our setting. There are two major differences: (1) GmH samples several central features by min-Hash functions from all over the image. Thus, neither all nor even most features are guaranteed to be included in the image description. (2) Given a central feature (randomly drawn by a hash function) the local neighborhood of such feature is described by a single sketch. This makes GmH very memory efficient, but not suitable for generic image retrieval because of low recall. Consequently, the authors use it to quickly retrieve images from a large database in order to build initial clusters of highly similar images. These clusters are used as "seeds"; each of the contained image is used as query for a traditional image search to find more cluster members that could not be retrieved by GmH.

Partition min-Hashing (PmH) In [15], a scheme is introduced that divides the image into partitions. Unlike for normal min-hashing, min-Hashes and sketches are computed for each partition independently. The search then proceeds by determining the sketch collisions for each of the partitions. This scheme is conceptually similar to a sliding window search as partitions may overlap and are processed step by step. The authors show that PmH is significantly faster

than mH and has identical collision probabilities for sketches as mH in the worst case, but theoretically better recall and precision if the duplicate image region only covers a small area.

3 Bundle min-Hashing

We build our bundling technique on min-Hashing mainly for two reasons: (1) feature bundles can be naturally represented as sparse sets and (2) min-Hashing does not imply a strict ordering or a hard matching criterion. This requirement is not met by local feature bundles. Due to image noise, viewpoint and lighting changes, the individual local features, their detection, and their quantization are unstable and vary across images. Even among two very similar images, it is extremely unlikely that they share identical bundles. We therefore utilize the min-Hashing scheme as a robust description of local feature bundles because it allows to search for similar (not identical) bundles.

The proposed bundling technique is an efficient approximate search method for similar images with higher memory requirements than pure near-duplicate search methods, but similar to that of bag-of-words. Its performance is close to bag-of-words, but at a much lower response ratio and much higher precision.

3.1 Min-Hashing

Min-Hashing is a locality-sensitive hashing technique that allows for approximate similarity search of sparse sets. It models an image as a sparse set of visual word occurrences. As the average number of visual words per image is much smaller than the vocabulary size for large vocabularies, the resulting feature histograms are sparse and are converted to binary histograms (i.e. sets representing whether a visual word is present at least once).

If one were able to do a linear search over all sets in a database, he might define a threshold on the overlap $ovr(I_1, I_2)$ between two such sets I_1 and I_2. This is equivalent to a threshold on the Jaccard similarity and determines whether these two sets are considered "identical" or matching. However, as the linear search over a database is infeasible in practice the min-Hashing scheme provides an efficient way to index these sets based on this overlap criterion.

Given the set $I = \{v_1, \ldots, v_l\}$ of l visual words of an image I, the min-Hash function is defined as

$$mh(I) = \underset{v_i \in I}{\operatorname{argmin}}\, h(v_i) \tag{1}$$

where h is a hash function that maps each visual word v_i *deterministically* to a random value from a uniform distribution. Thus, the min-Hash mh itself is a visual word, namely that word that yields the minimum hash value (hence the name min-Hash). The probability that a min-Hash function mh will have the same value for two different sets I_1 and I_2 is equal to the set overlap:

$$P(mh(I_1) = mh(I_2)) = ovr(I_1, I_2) = \frac{|I_1 \cap I_2|}{|I_1 \cup I_2|} \tag{2}$$

Note that an individual min-Hash value not only represents a randomly drawn word that is part of the set, but each min-Hash also implicitly "describes" the words that are *not* present and would have generated a smaller hash—because otherwise it would have been a different min-Hash value.

The approximate search for similar sets is then performed by finding sets that share min-Hashes. As single min-Hashes yield true matches as well as many false positives or random collisions, multiple min-Hashes are grouped into k-tuples, called *sketches*. This aggregation increases precision drastically at the cost of recall. To improve recall, this process is repeated n times. Thus, independently drawn min-Hashes are grouped into n tuples of length k. The probability that two different sets have at least one of these n sketches in common is then given by

$$P(\text{collision}) = 1 - (1 - ovr(I_1, I_2)^k)^n \tag{3}$$

This probability depends on the set overlap. In practice the overlap between non-near-duplicate images that still show the same object is small. In fact, the average overlap for a large number of partial near-duplicate images was reported to be 0.019 in [15]. This clearly shows that for applications which target the retrieval of partial-near-duplicates, e.g. visually similar objects rather than full-near-duplicates, the most important part of that probability function is the behavior close to 0.

The indexing of sets and the approximate search are performed as follows: to index sets their corresponding sketches are inserted into hash-tables (by hashing the sketches itself into hash keys), which turn the (exact) search for a part of the set (the sketch) into simple lookups. To retrieve the sets similar to a query set, one simply computes the corresponding sketches and searches for the sets in the database that have one or more sketches in common with the query. A lookup of each query sketch determines whether this sketch is present in the hash table, which we denote as "collision" in the following. The lookups can be done efficiently in constant time as hash tables offer in amortized $\mathcal{O}(1)$. If there is a query sketch of size k that collides with a sketch in the hash table, then the similarity of their originating sets is surely >0, because at least k of the min-Hash functions agreed. To avoid collisions resulting from unrelated min-Hash functions, the sketches are put into separate hash tables: the kth sketch is inserted into the kth hash table.

3.2 Bundle min-Hashing

The idea of our bundling technique is simple: we describe the neighborhoods around local features by bundles which aggregate the visual word labels of the corresponding visual features. The bundling starts by selecting *central features*, i.e. all features in an image with a sufficient number of local features in their neighborhood. Analogous to the feature histogram of a full image, the small neighborhood surrounding each central feature represents a "micro-bag-of-words". Such a bag-of-words vector will be extremely sparse because only a fraction of all features in the image is present in that particular neighborhood. Since the features of a bundle are spatially close to each other, they are likely to describe the same object or region of interest.

More specifically, given a feature $\mathbf{x}_i$ its corresponding feature bundle $b(\mathbf{x}_i)$ is defined as the set of spatially close features for a given feature $\mathbf{x}_i$:

$$b(\mathbf{x_i}) = \{\mathbf{x}_j | \mathbf{x}_j \in N(\mathbf{x}_i)\} \tag{4}$$

where $N(\mathbf{x}_i)$ is the *neighborhood* of feature $\mathbf{x}_i$ which is described at the end of this section. We further assume that for all features $\mathbf{x}_i$ in an image the descriptor vectors have been quantized to the corresponding visual words $v_i = q(\mathbf{x}_i)$.

The bundle $b(\mathbf{x}_i)$ is then represented by the corresponding set of visual words of all features included in that bundle:

$$W_i(b(\mathbf{x}_i)) = \{q(\mathbf{x}_j) \mid \mathbf{x}_j \in b(\mathbf{x}_i)\}. \tag{5}$$

The set W_i is then indexed by regular min-Hashing.

In extensive experiments we observed the following: first, sketches of size 2 perform best compared to sketches of size 3. Second, we found that the performance increases drastically if the first sketch element is not determined by min-Hashing but rather set to the visual word of the central feature itself. That is, for each bundle the nth sketch is given as 2-tuple

$$(v_i,\ mh_n(W_i(b(\mathbf{x}_i)))) \tag{6}$$

where v_i denotes the visual word label of the central feature and mh_n denotes the min-Hash returned by the nth min-Hash function from the set of all visual words W_i present in bundle $b(\mathbf{x_i})$. The full process is illustrated in Fig. 1.

The major advantage can be seen when comparing the collision probabilities of a single min-Hash and sketches of size 2 (see Fig. 2). With our approach two bundles (the central feature plus a single min-Hash) with an overlap of only 0.2 have a 50 % chance that one of 4 sketches collide. This means, while there are multiple feature bundles that need to be described, each with several sketches, only very few sketches are needed per bundle to achieve a high probability to retrieve similar sets. This keeps the memory requirements for the indexing low. Further redundancy is added as images that contain multiple bundles which may overlap. If some bundles do not match (collide) across images, there is the chance that other bundles in the same images collide.

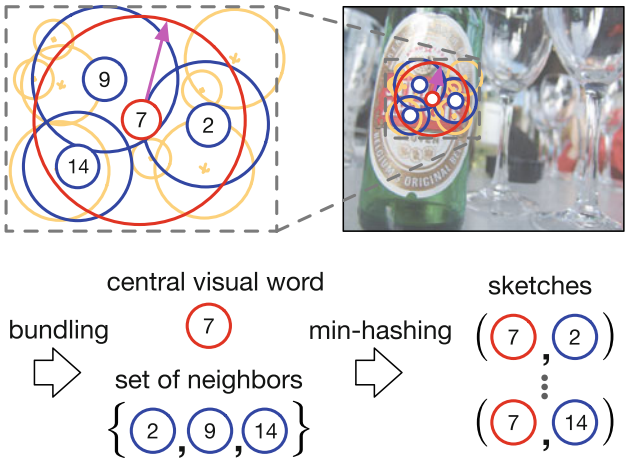

Fig. 1 *Bundle min-Hashing* the neighborhood around a local feature, the *central feature* (*red*), is described by a feature bundle. Features that are too far away or on scales too different from that of the central feature are ignored during the bundling (*yellow*). The features included in such a bundle (*blue*) are represented as a set of visual word occurrences and indexed by min-Hashing (see Sect. 3.2)

Bundling strategy The bundling strategy $N(\mathbf{x}_i)$ we use is based on the intuition that features which are spatially close to each other are likely to describe the same object. That is, given a central feature we bundle it with its direct spatial neighbors. We require that at least two other features are present in its neighborhood and that these must be on a similar scale. This is in line with the observation that true feature correspondences are often at the same scale [8]. Thus, each feature that is closer to a given central feature $\mathbf{x}_i$ than a given cut-off radius $r_{\max}$ is included in the respective bundle $b(\mathbf{x}_i)$: the radius $r_{\max}$ is chosen relative to the scale (patch size) of the central feature s_i. The minimum and maximum scales $s_{\min}$ and $s_{\max}$ control the scale band considered for determining the neighbors relative to the scale of the central feature. Figure 1 shows the bundling criterion for $s_{\min} = 0.5$, $s_{\max} = 2.0$ and $r_{\max} = 1.0$ (red circle = radius of the central feature itself).

These criteria eventually make the bundling process ignore features that have no spatial neighbors in a reasonable distance. This effectively decreases the number of bundles below the number of local features in an image such that fewer sketches need to be stored resulting in a smaller total memory consumption.

Bundling implementation The features within a certain distance to a central feature are efficiently determined by orthogonal range search techniques like kd-trees or range trees which allow sub-linear search.

Min-Hash functions The min-Hashes can be computed by generating multiple random permutations on the range of all

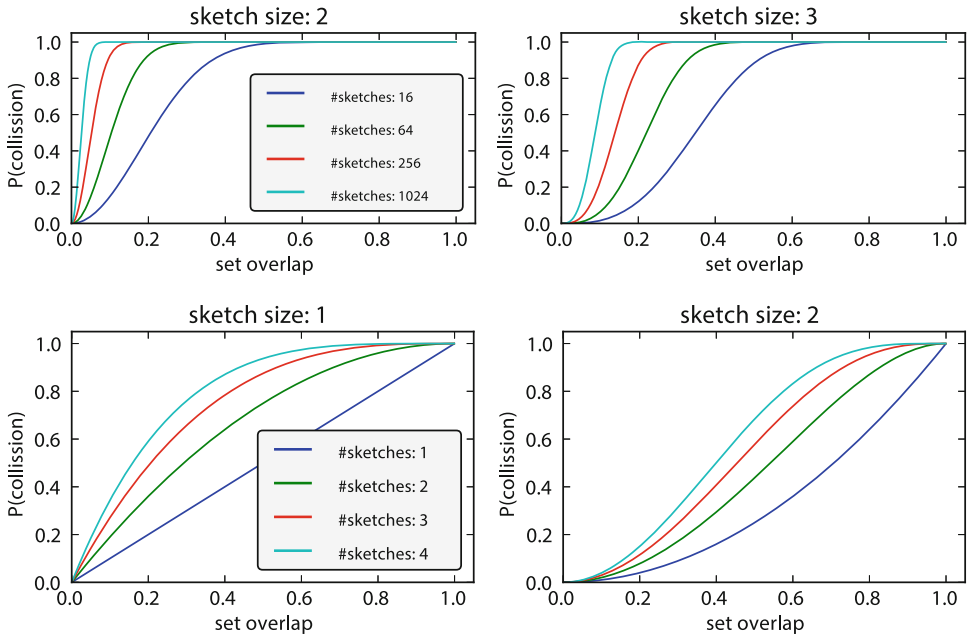

Fig. 2 *Upper row* collision probabilities in regular min-Hash with sketches of size 2 (*left*) and 3 (*right*). *Lower row* collision probabilities given the set overlap between bundles. *Left* single min-Hash (as used by bundle min-Hashing). *Right* sketches of size 2

visual words and storing them in a lookup table. Given such permutation the hash for each visual word in a set would be obtained by simple lookup. However, for large vocabularies and multiple hash functions this lookup table is larger than CPU caches and lookups get slow. Therefore, we use randomizing hash functions instead of precomputed permutation tables to compute the hashes. These hash functions return a uniformly drawn random value deterministically determined by the given visual word and a seed that is kept fixed. This implementation is both substantially more memory efficient and faster than lookup tables. There is no guarantee that all visual words will produce different values but this approximation seems "good enough" while being much faster than a lookup table.

Adjustable search The bundle representation by multiple sketches has an advantageous side-effect: it facilitates a search tunable from high precision to high recall *without* post-retrieval steps or redundant indexing. Once bundles have been indexed with k sketches per bundle, the strictness of the search may be changed by varying the number of sketches *at query time* from $1 \ldots k$. As the sketch collision probability is proportional to the set overlap, bundles that have a high overlap with the query will be retrieved earlier than bundles with smaller overlap. Thus, by varying the number of query sketches one can adjust the strictness of the search (see Table 1: mean precision mP and mean recall mR change with varying #sketches). As the ith sketch was inserted into the ith hash table, querying sketches from $1 \ldots i$ will yield only bundles where the corresponding sketches and hash functions in tables $1 \ldots i$ agreed at least once.

3.3 Ranking and filtering

Once the images which share similar bundles with the query are determined, they may be ranked by their similarity to the query. One possibility is to compute a similarity based on the number of matching bundles between these images. However, a ranking based on the cosine similarity between the full bag-of-words histogram of the query image and the retrieved images performs significantly better than a ranking based on the sketch collision counts, as it is difficult to derive a good measure for image similarity based on a few collisions only. Thus, in our experiments we rank all retrieval results by the cosine similarity between the bag-of-words histograms describing the full images [25].

In other words, the retrieval by feature bundles is effectively a filtering step: the bundles are used to quickly fetch a small set of images that are very likely relevant. These images are then ranked by the cosine similarity between bag-of-words histograms obtained with a vocabulary of 1M words (see Sect. 3.5.3). We also address the problem of visual word burstiness by taking the square root of each tf-idf histogram entry as proposed in [9]. This is important for logo recognition as logos often consist of text and text-like elements which are known to be prone to yield repeated visual words ("visual words bursts"). The small response ratio of the retrieval with bundles is a major benefit: small result sets of high precision can be processed quickly even with sophisticated re-ranking methods. Query expansion may then be used to do a second retrieval.

Table 1 Comparison of bag-of-words retrieval with bundle min-Hashing

#Sketches	$s_{\min}$	$s_{\max}$	$r_{\max}$	Voc.	mAP	AvgTop4	mP	mR	RR	⌀# bundles	Rel. storage (%)
Bag-of-words, tf-idf-sqrt weighting				200K	0.510	2.88	0.010	*0.952*	0.912	2,468.1 words	100
Bag-of-words, tf-idf-sqrt weighting				500K	0.545	3.06	0.010	0.932	0.845	2,468.1 words	100
Bag-of-words, tf-idf-sqrt weighting				1M	0.545	**3.16**	0.011	0.911	0.763	2,468.1 words	100
4	0.5	2.0	1.0	200K	*0.554*	3.14	0.243	0.639	0.025	1,640.9	266
3	0.5	2.0	1.0	200K	0.545	3.13	0.269	0.623	0.022	1,640.9	199
2	0.5	2.0	1.0	200K	0.527	3.09	0.312	0.592	0.018	1,640.9	133
1	0.5	2.0	1.0	200K	0.478	3.04	**0.401**	0.520	**0.012**	1,640.9	**66**

The upper part shows the scores of three different bag-of-words retrieval runs. The lower part shows the bundle configuration that resulted in the highest mAP for 1, 2, 3 and 4 sketches per bundle. The columns $s_{\min}$, $s_{\max}$, $r_{\max}$ and $Voc.$ denote the bundling parameters (as described in Sect. 3.2) and the vocabulary size. The scores follow in the order of mAP, average top 4 score, mean precision, mean recall and response ratio. The column ⌀# bundles denotes the average number of bundles per image. The last column estimates the memory consumption and shows the number of hash table entries (sketches) relative to the number of visual words per image
Bold values denote the best score

3.4 Influence of parameters

Several parameters affect the behavior of Bundle Min-Hashing. In this section we describe common trade-offs.

Visual vocabulary The quantization of high-dimensional descriptors to discrete visual words is a lossy quantization. For near-duplicate retrieval and also for the retrieval of similar objects it has been shown that large vocabularies are beneficial despite the larger quantization error. This suggests that for retrieval it is more important to suppress false correspondences than obtaining a large number of tentative correspondences.

Sketch size The number of min-Hashes that are aggregated into k-tuples directly control the collision probability. With a larger sketch size the collision probability of random collision not only decreases drastically but also leads to low recall. In practice mostly sketches of size 2 and 3 are used [5,6,15] as larger sizes have impracticable low recall and a single min-hash just represents a single word. For Bundle min-Hashing we use sketches of size 2 but set the first component to the central visual words.

Number of sketches In contrast to the sketch size, increasing the number of sketches increases the collision probability. In other works, a few dozen up to a few thousand sketches are used depending on the representation. In our work, we compute multiple bundles for multiple features in the image and therefore we want to minimize the memory needed to store them. We observed that 2 sketches are sufficient for reasonable performance and 3 or 4 sketches slightly improve it further.

Locality The features that are eventually bundled into a single description are sampled from a region which size depends on the central features. Thus, the region size implicitly influences the number of features and therefore the "noise ratio" when min-Hashes are computed for that region. Intuitively, features close to each other (e.g. with overlap) are correlated, while features lying far from each other may be treated as approximately independent. In the same manner the used interest point detector also has a major influence. Detectors that fire on blob-like regions usually yield more distributed interest points than a corner detector that fires on every peak of corner-like structures.

3.5 Experiments

3.5.1 Datasets

As we focus on the retrieval of small objects we test and optimize our approach on the FlickrLogos-32 [24] dataset. Then, we further demonstrate its performance on the datasets UkBench [19] and Oxford [20] as well.

FlickrLogos-32 The dataset we chose to evaluate our logo retrieval approach is FlickrLogos-32. It consists of 32 classes of brand logos [24]. Compared to other retrieval-centric datasets, e.g. Oxford, images of a similar class in FlickrLogos-32 share much smaller visually similar regions: the average object size of the 55 query images (annotated in the ground truth) of the Oxford dataset is 38 % of the total area of the image (median 28 %) while the average object size in the test set of the FlickrLogos dataset is only 9 % (median 5 %). As the retrieval of the Oxford buildings is sometimes coined "object retrieval", the retrieval task on the FlickrLogos dataset can be considered as "small object retrieval".

The dataset is split into three disjoint subsets. For each logo class, we have 10 training images, 30 validation images, and 30 test images—each containing at least one instance

of the respective logo. Both validation and test set further contain 3,000 logo-free images. Our evaluation protocol is as follows: all images in the training and validation sets (4,280 images), including those that do not contain any logo are indexed by the respective method. The 960 images in the test set which do show a logo (given by the ground truth) are used as queries to find the most similar images from the training and validation sets. The respective retrieval results are re-ranked by the cosine similarity (see Sect. 3.3).

This logo dataset targets the evaluation of small object retrieval and classification since it features logos that can be considered as rigid objects with an approximately planar surface. The difficulty arises from the great variance of object sizes, from tiny logos in the background to image-filling views. Other challenges are perspective and eventually multi-class recognition.

UkBench Traditionally, min-Hashing has mostly been used for near-duplicate retrieval task. In that spirit, we also evaluate our approach on the UkBench dataset [19] that features 10,200 images arranged in groups of 4 images showing the same object, scene or CD cover.

Oxford The dataset of Oxford buildings [20] is one of the most well-known datasets for evaluating image retrieval. This dataset contains 5,063 images of 11 buildings from Oxford as well as various distractor images. It is known for its difficulty to discriminate very similar building facades from each other.

3.5.2 Visual features

For our experiments we used SIFT descriptors computed from interest points found by the Difference-of-Gaussian (DoG) detector. For evaluation of warping (Sect. 5) as well as our logo recognition system (Sect. 6) we used the improved RootSIFT descriptors. In all other cases—unlike mentioned—we used the traditional SIFT descriptors in order to make our scores comparable to those in the literature.

We used an interest point detector that yields circular feature patches but the bundling scheme may also be used with features from affine covariant regions. When determining the features to be included in a bundle one simply has to take account of the elliptic regions.

The bundling parameters we show are tuned for a particular detector (DoG) and therefore for its detection characteristics. It is likely that bundling parameters need to be specifically adapted to a certain interest point detectors as each detector varies in the number of detections, the distribution of interest points (e.g. blob-like, corner-like) and the behavior of the non-maximum-suppression. Different detectors will yield a different number of features in a predefined neighborhood. Therefore, one has to adjust the bundling parameter to the sparsity of the neighborhoods depending on the interest point detector.

For clustering the descriptor to obtain visual words we use approximate k-means which employs the same k-means iterations as standard k-means but replaces the exact distance computations by approximated ones. We use a forest of 8 randomized kd-trees to index the visual word centers [18]. This kd-forest allows to perform approximate nearest neighbor search to find the nearest cluster for a descriptor both during clustering as well as when quantizing descriptors to visual words. The vocabulary and IDF weights have been computed on the training and validation sets of FlickrLogos-32 only.

3.5.3 Evaluation

As a retrieval system should have both high precision and high recall, we measure the retrieval performance by mean average precision (mAP) which describes the area under the precision–recall curve. It characterizes both aspects; a system will only gain a high mAP if both precision and recall are high. Here, the AP is computed as $\mathrm{AP} = \sum_{i=1}^{N} \frac{1}{2}(P_i + P_{i-1}) \cdot (R_i - R_{i-1})$ with $R_0 = 0$, $P_0 = 1$ where P_i and R_i denote precision/recall at the ith position in the retrieved list.

The response ratio (RR) is used to measure the efficiency of the retrieval. It describes the number of retrieved images in relation to the database size. The higher the response ratio the more images in the result list, which is usually post-processed or verified by computationally expensive methods. A low response ratio will thus increase the overall efficiency of the search. The precision among the top-ranked images is measured by the average top 4 score (Top4) defined as average number of correctly retrieved images among the top 4 results. A perfect retrieval would yield a score of 4.0.

Bag-of-words First we compare the performance of various approaches based purely on the cosine similarity between bag-of-words on the FlickrLogos-32 dataset. Thus, we evaluate the retrieval performance of a plain bag-of-words search with varying vocabularies and varying patch sizes of the descriptors. We are especially interested in the impact of extremely large visual vocabularies on the performance. Thus, we vary the vocabularies from 10,000 (10K) to 4,000,000 (4M) words.

The results are shown in Fig. 3. In [22] we have already shown that IDF-weighting is always beneficial in the bag-of-words framework, even for large vocabularies greater than 1 million words. Thus, tf-idf weighting was used in all cases. As found in prior works, large vocabularies show significantly better performance. The peak is consistently at 500K/1M words. The patch size that is described by a SIFT descriptor linearly depends on the scale and a magnification factor m. We further test how this magnifier changes the performance. The best performance is obtained with descriptors computed with $m = 3$ as in Lowe's work. In addition we compare the performance of bag-of-words based on standard

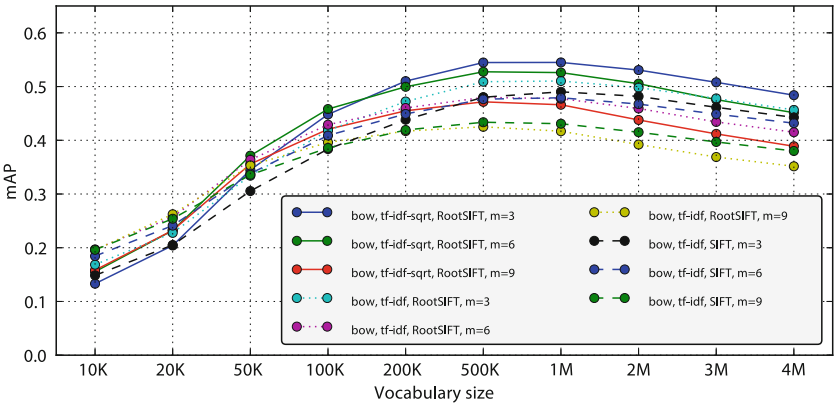

Fig. 3 Retrieval score (mAP) for several bag-of-words variants on the FlickrLogos-32 dataset

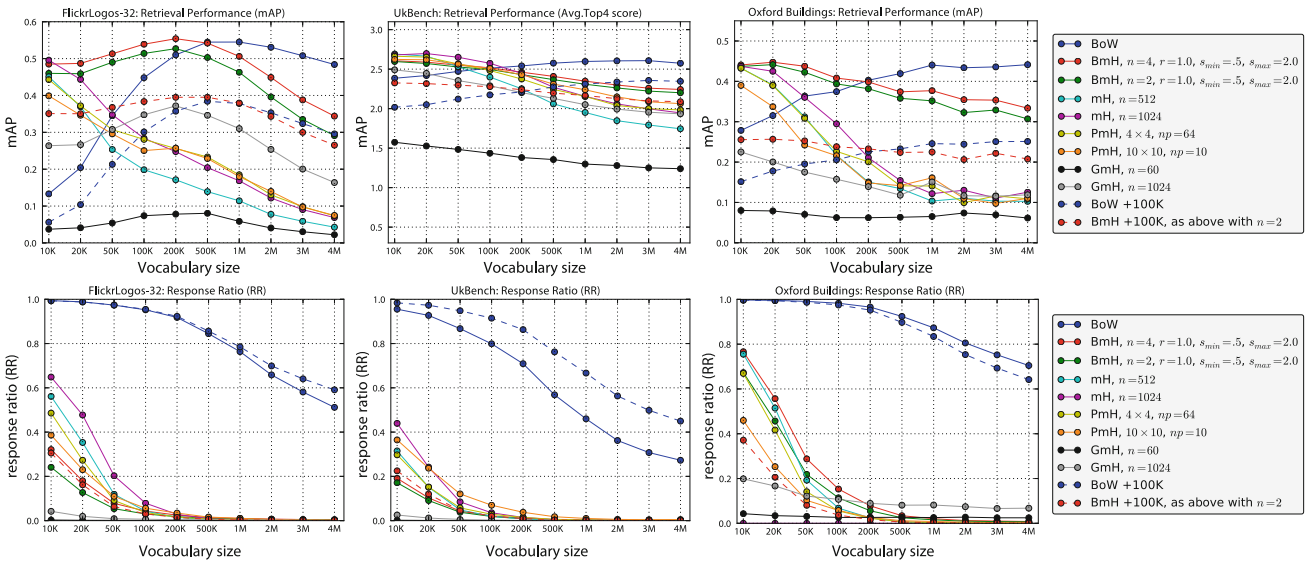

Fig. 4 From *left to right* retrieval results on FlickrLogos-32, UkBench and the Oxford buildings dataset. The *upper rows* show the mAP, the *lower row* the corresponding response ratio of each approach. The performance of Bundle min-Hashing (*BmH*) is on par with bag-of-words (*BoW*) and outperforms min-Hash (*mH*), Partition min-Hash (*PmH*) and Geometric min-Hashing (*GmH*). Its response ratio is an order of magnitude lower than bag-of-words and comparable to the latter

SIFT with that of the relatively new RootSIFT variant [2]. Clearly, the bag-of-words based on RootSIFT outperforms the SIFT-based bag-of-words. Finally, the burstiness measure proposed in [9] where the square root is taken for each element of the tf-idf-weighted histogram further improves the retrieval performance (denoted as "tf-idf-sqrt"in Fig. 3) as it down-weights repeating and thus less informative visual words ("bursts").

For further experiments on FlickrLogos-32, we therefore use visual words computed from RootSIFT descriptors and re-rank the results retrieved by feature bundles by the cosine similarity between bag-of-words histograms with square-rooted tf-idf weights. In order to make our results comparable to others in the literature we use regular SIFT descriptors for evaluating on UkBench and Oxford and omit the burstiness measure (plain tf-idf instead). In all cases the best-performing vocabulary of 1M words is used for re-ranking, disregarding which vocabulary was used when building the feature bundles.

Feature bundles We evaluate the performance of our bundling strategy with regards to mAP and response ratio and compare it to a retrieval with bag-of-words and tf-idf weighting, as described, e.g. in [25].

In order to find the best bundle configurations we have performed extensive evaluations on the parameters of the bundle configuration. Due to limited space, we cannot show a detailed evaluation for all parameters. Instead, we report the best-performing bundle configuration (with respect to mAP) in Table 1. Similar to bag-of-words the bundles profit from large vocabularies, but the peak is at $200K$–$500K$ words. Most important, the bundles roughly have equal performance as bag-of-words, but have an order of magnitude

lower response ratio (RR) as shown in Table 1 and also in Fig. 4.

Note that we re-rank the result lists determined by Bundle min-Hashing by the cosine similarity as given by the bag-of-words model. As the bundling is by definition only able to find correspondences between images that share visual words, the result set of the retrieval by feature bundles is a *true subset* of the result set obtained with bag-of-words retrieval. This clearly demonstrates the discriminative power of feature bundles for efficient filtering before more expensive post-retrieval steps are applied to the result set.

Min-Hash, Partition min-Hash, Geometric min-Hashing We extensively compare our approach to min-Hashing (mH) as well as Partition min-Hash (PmH) and Geometric min-Hashing (GmH) on three different datasets. These approaches are specifically meant for (partial) near-duplicate image search. It may seem unfair to compare these to approaches that have higher memory requirements for their image description and exploit it for a more accurate retrieval. However, this comparison shows how these methods may perform on those datasets when used with typical parameters. For all experiments the sketch size was set to 2; n denotes the number of sketches. In case of Partition min-Hash 4×4 and 10×10 denote 16 and 100 overlapping partitions whereas np denotes the number of sketches per partition. The overlap was set 50 % in all runs. For Geometric min-Hashing we follow the setup in [5].

As already mentioned, we re-rank each preliminary result set of all approaches by the cosine similarity (see Sect. 3.3). We would like to point out that min-Hash, Partition min-Hash as well as Geometric min-Hashing *significantly* benefit from this. Bundle min-Hashing benefits as well but the effect is less pronounced.

In Fig. 4, the results for the previously selected Bundle min-Hashing configuration are compared to the former approaches and bag-of-words. For retrieval of near-duplicate images there is little difference between most approaches. However, for object search on Oxford and on FlickrLogos-32 the differences are pronounced. Bag-of-words has high scores in every settings at the cost of a huge response ratio, i.e. a single query still retrieves 80 %+ of the whole database. min-Hash, Partition min-Hash and Geometric min-Hash do not suffer from this but from low recall. In contrast to, e.g., Geometric min-Hashing (with high precision and low recall) and bag-of-words (with low precision and high recall), Bundle min-Hashing seems an intermediate approach combining the best of both worlds: it has low response ratio, high precision and high mAP.

Speed We measured the wall time of a single-threaded C++ application for bundling, min-Hashing, insertion into hash tables and all I/O operations excluding feature computation and quantization. For instance, with the configuration as in Table 1 and 4 sketches, indexing of the FlickrLogos-32 dataset (4,280 images) takes about 96.8 s (≈23 ms per image) while processing the 960 queries takes about 13.4 s (≈14 ms per image).

Scalability We further test how the retrieval is affected once 100,000 distractor images (randomly chosen Flickr images) are added to the database (denoted as "+100K" in Fig. 4). For this scenario we used the more memory-conservative scheme with 2 sketches per bundle that performs almost as good as 4 sketches per bundle but requires only half the hash tables. As expected the mAP consistently drops for bag-of-words as well as Bundle min-Hashing but the latter seems less affected than bag-of-words and slightly outperforms it even while we used only 2 sketches per bundle. This suggests that higher-order descriptions are more distinctive and deteriorate slower than first-order descriptions like bag-of-words with growing database sizes and increasing noise.

4 Fast re-ranking: 1P-WGC-RANSAC

In order to ensure that the top retrieved images correctly show the query object we employ a spatial verification step on the list of retrieved images. The gold standard for this purpose is RANSAC. Our approach is based on a variant that uses single feature correspondences to estimate a transformation between two images [20]. The associated scale and dominant orientation of the two local features of each correspondence is used to estimate a similarity transform (4 degrees-of-freedom with translation, rotation and uniform scaling). The major benefit is that a single correspondence generates a hypothesis. Evaluating all these correspondences makes this procedure deterministic, fast and robust to small inlier ratios. The top 10 hypotheses with the highest score determined by the symmetric transfer error and truncated quadratic cost function [14] are kept for further refinement. If the top hypotheses have more than 15 inliers these are then refined by a local optimization (LO) step that estimates a fully projective transformation via least-median-of-squares.

While RANSAC is in general considered as slow and costly this is not entirely true. In fact we found that most of the time was spent for projective re-estimation. Moreover, while this refinement improves the visual quality of the estimated transformation it has little effect on the induced ranking. To illustrate this, Fig. 5 shows the top 10 hypotheses *without* projective re-estimation. The similarity of these hypotheses indicates that for re-ranking the re-estimation maybe omitted. Thus, we propose a new variant 1P-WGC-RANSAC *without* subsequent LO step that is faster than a non-WGC-constrained RANSAC and much faster than a variant estimating a fully projective transformation between images.

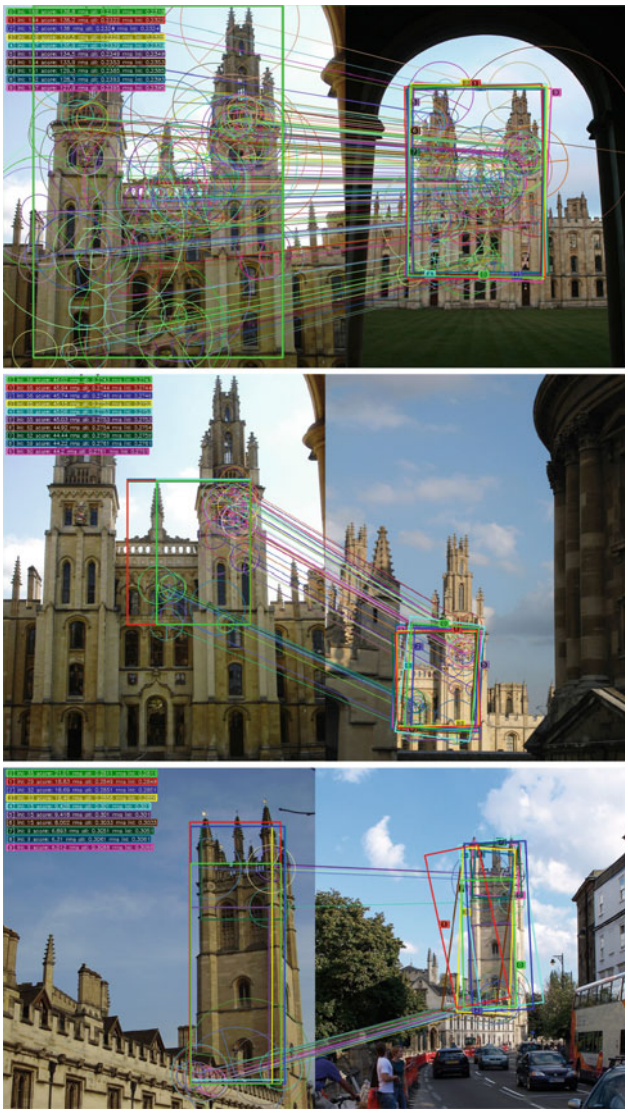

Fig. 5 Three examples of spatial re-ranking with 1P-WGC-RANSAC. The top 10 hypotheses are projected as *colored rectangles* into the images. *Top* an easy case; the top 10 hypotheses are almost identical and hard to distinguish. *Middle* a more challenging case. There is more variation but still the top 10 hypotheses are quite similar. *Bottom* a difficult case. The hypotheses show bigger variance, most likely because of the somewhat degenerate point correspondences at the *lower left corner* of the matched image regions

For 1P-WGC-RANSAC, a weak geometric consistency (WGC) constraint is imposed. Only correspondences from features with orientations and scales that are consistent with the estimated transformation may be scored as inliers. We found that this constraint has little impact on the quality of the re-ranking. The re-ranking is neither significantly better nor worse. However, it acts as a filter that can be employed *before* the inliers are determined. If a feature correspondence violates the WGC constraint it is directly treated as outlier. Thus, the error function within the RANSAC framework is speeded up as there is no need to compute the perspective mapping for these false correspondences. Here, we use the following constraint: scale change must be in [0.5, 2.0] and angles must differ less than 30°.

We compare our approach to that of Philbin et al. [20] and Arandjelovic et al. [2] on the Oxford5K dataset [20] following the common test protocol: the top 1,000 retrieval results per query are re-ranked with an early stop if 20 images in a row could not be verified successfully. Images are scored by the sum of the IDF weights of all inlier words and verified images are placed above unverified images in the result list. The results are shown in Table 2. Here, "SP" and "RANSAC" denote that spatial re-ranking was performed.

One can see that our implementation (using DoG-SIFT, magnifier of 9) yields slightly higher (1M words) or even significantly higher scores (100K words) than that of Philbin et al. [20] (using Hessian-affine SIFT). Quite surprisingly, the performance after re-ranking with the smaller vocabulary of 100K words is close to the one with 1M words. This demonstrates that the proposed scheme is able to deal with a small vocabulary, its less discriminative correspondences and small inlier ratios.

Similar on the FlickrLogos-32 dataset (see Table 3): the spatial verification of the top 200 images further improves the result as well. For both datasets the projective re-estimation does not improve the performance. It further refines the homography but is not able to discard additional false positives. Most likely a simple 4-dof geometric constraint for re-ranking is sufficient to filter out false positives. This underlines that re-ranking does not require to estimate fully affine/projective homographies and due to its speed 1P-WGC-RANSAC is beneficial for spatial verification.

To measure the time we performed all experiments on the same machine using 1 thread for execution of our C++ program and measured the wall time as median over 10 runs. In summary the WGC-constrained 1-point RANSAC without LO is about 30 % faster than without the WGC constraint, has slightly better performance for small vocabularies and is much faster than with LO refinement. Its throughput is extremely high (e.g. see ★ in Table 2: re-ranked 5,813 images $\approx$ 440 images/s $\approx$ 2.3 ms per image, single-threaded, including I/O) making it suitable for real-time applications.

5 Warping

While current local features are by design scale invariant and also somewhat robust to changes in lighting and image noise, it is well known that local features such as SIFT are particularly susceptible to changes in perspective. With increasing vocabulary size this effect gets more severe: descriptors computed from image patches that are actually identical but seen

Table 2 Comparison of spatial re-ranking results for the Oxford5K dataset following the protocol in [20]

Method	Voc	mAP	Time (s)
Philbin et al. [20], bow	100K	0.535	–
Philbin et al. [20], bow + SP	100K	0.597	–
Bow, tf-idf, SIFT	100K	0.571	–
1P-RANSAC, incl. LO	100K	0.678	160
1P-RANSAC, no LO	100K	0.680	72
1P-WGC-RANSAC, incl. LO	100K	0.693	115
1P-WGC-RANSAC, no LO	100K	0.692	53
Philbin et al. [20], bow	1M	0.618	–
Philbin et al. [20], bow + SP	1M	0.645	–
Arandjelovic et al. [2] SIFT, bow	1M	0.636	–
Arandjelovic et al. [2] SIFT, bow + SP	1M	0.672	–
Bow, tf-idf, SIFT	1M	0.647	–
1P-RANSAC, incl. LO	1M	0.712	54
1P-RANSAC, no LO	1M	0.711	15
1P-WGC-RANSAC, incl. LO	1M	0.704	50
1P-WGC-RANSAC, no LO	1M	0.703	12
Arandjelovic et al. [2] RootSIFT, bow	1M	0.683	–
Arandjelovic et al. [2] RootSIFT, bow + SP	1M	0.720	–
Bow, tf-idf, RootSIFT	1M	0.675	–
1P-RANSAC, incl. LO	1M	0.728	92
1P-RANSAC, no LO	1M	0.729	17
1P-WGC-RANSAC, incl. LO	1M	0.723	55
1P-WGC-RANSAC, no LO ★	1M	0.723	13

Table 3 FlickrLogos-32: spatial re-ranking results

Method	Voc.	mAP	Time (s)
Bow, tf-idf-sqrt	100K	0.448	–
1P-RANSAC, incl. LO	100K	0.513	953
1P-RANSAC, no LO	100K	0.513	387
1P-WGC-RANSAC, incl. LO	100K	0.510	731
1P-WGC-RANSAC, no LO	100K	0.510	325
Bow, tf-idf-sqrt	1M	0.545	–
1P-RANSAC, incl. LO	1M	0.565	510
1P-RANSAC, no LO	1M	0.565	153
1P-WGC-RANSAC, incl. LO	1M	0.568	447
1P-WGC-RANSAC, no LO	1M	0.568	111

from a different perspective are quantized to different—and therefore unrelated—visual words.

There exist several partial solutions to this problem. The most popular is query expansion (QE) where the top-ranked retrieved images are exploited to augment the original query. The augmented query is then re-issued in order to retrieve images that have not been found in the first round. Consequently, query expansion fails—and causes the results to be worse than without—if the top-retrieved images are false positives. This may happen if the query is actually challenging or only few true positives are contained in the database.

We propose a different method to overcome this problem, especially suited for small objects where it is crucial to find the few true matching visual words. It is a purely data-driven approach that synthesizes new images from existing images by applying transformations to the image itself, a process often called "warping". There are different ways to exploit image warping:

1. *Synthetic query expansion (SynQE)* Multiple versions of the query image may be synthesized simulating the query as it may be seen under different conditions and perspectives. Each image is then treated as an individual query; their corresponding result lists are then merged into a single list. This method is illustrated in the upper half of Fig. 6.

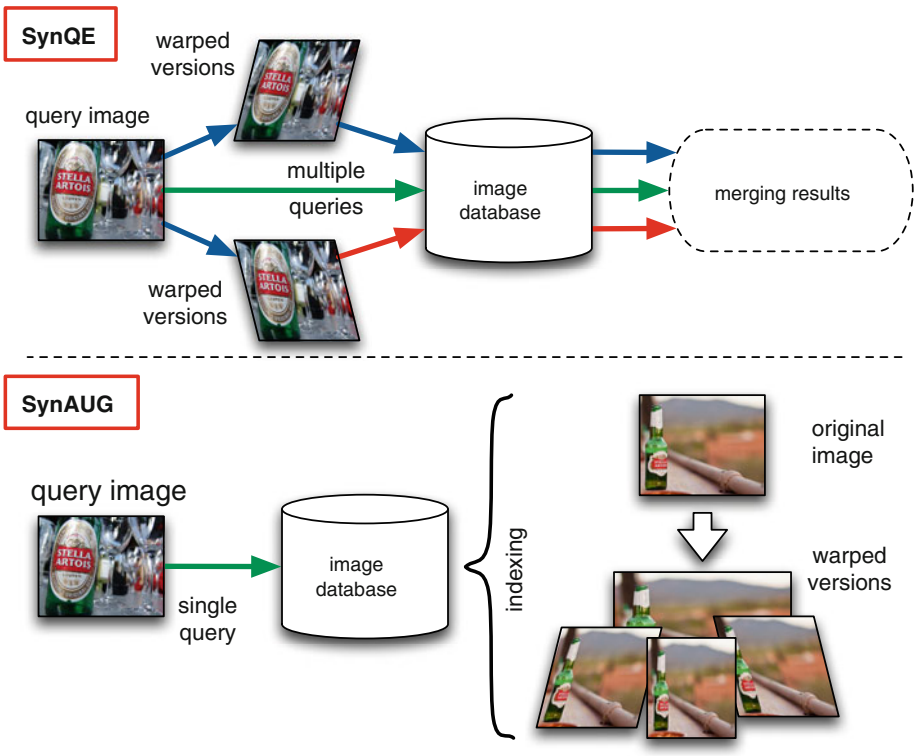

Fig. 6 *Top* synthetic query expansion. *Bottom* synthetic database augmentation

2. *Synthetic database augmentation (SynAUG)* The database is augmented by adding new generated images synthesized from each original database image. This is especially useful if it is desired that a query containing certain predefined objects—such as logos—should find the true results with high probability from a limited set of manually managed reference images. This is shown in the lower half of Fig. 6.
3. *SynQE + SynAUG* The combination of (1) and (2). This can be seen as counterpart to ASIFT [27] working with discrete visual words and an inverted index or another database instead of comparing raw descriptors between two images.

We choose the following simple transformations to synthesize new images: $S_x(\alpha)$, $S_y(\alpha)$, $S_x(\alpha)R(45°)S_x(\alpha)$ and $S_x(\alpha)R(-45°)S_x(\alpha)$. $S_x(\alpha)$ denotes the matrix for scaling by factor α in x-direction, $S_y(\alpha)$ analog in y-direction and $R(45°)$ denotes the matrix for rotation by 45°. The last two transformations are opposed shearings along x direction.[1] The inverse transformations of the former four are added as well, resulting in a total of eight transformations.

Intuitively, the synthesizing of images creates variations from a single image. The more variation is captured within the index the more likely the retrieval of an arbitrary query will succeed. To illustrate the effect of such warping on an individual local feature the used transformations are visualized in Fig. 7. The original patch described by a local feature is shown as black circle. Once images are warped with a scaling factor of $\alpha < 1$ the image is effectively down-scaled such that the actual described image region is larger than the original patch. Similar, if the images are up-scaled ($\alpha > 1$) the actual described image patch is smaller than the region described by the original local feature. The elliptic shape depends on the transformation and while obtained differently remind of affine covariant regions as obtained by Hessian-affine or Harris-affine feature detectors [17]. In fact, our technique effectively simulates a global affine transformation applied to the whole image while affine covariant detectors estimate an affine transformation per local feature.

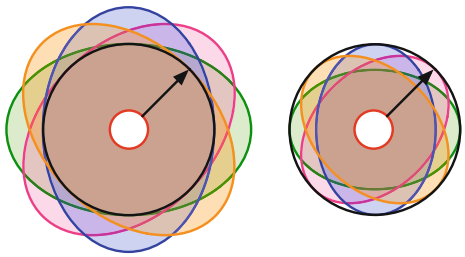

Fig. 7 Visualization of the patches described by visual words computed from warped images *back-projected* into the original image. The original patch is shown as *black circle*. *Left* $\alpha = 0.7$, *right* $\alpha^{-1} \approx 1.43$. Images to scale

In practice, the following issue needs to be addressed: we observed artifacts when computing local features from

[1] The two shearings along y-direction are equivalent.

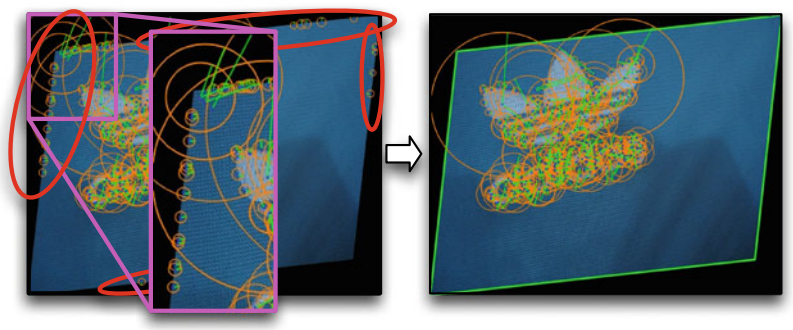

Fig. 8 *Left* local feature patches (*orange circles*) with their associated orientation (*green line*) as extracted from warped images. The detector often fires on the boundaries and introduces artifacts (*marked red*). *Right* features retained after eliminating those too close to the boundary (*green contour*)

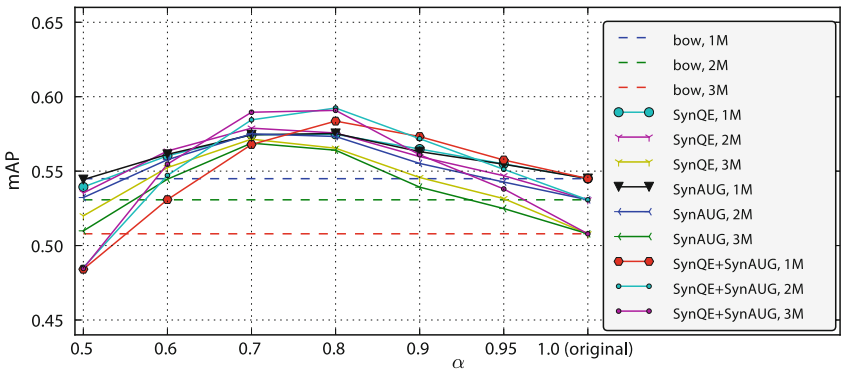

Fig. 9 *FlickrLogos-32* impact of synthetic query expansion and database augmentation on BoW retrieval performance

warped images. The feature detector often fires close to or directly on the boundaries of transformed images. Even while detections on edge-like structures are suppressed during feature detection, due to image noise these still occur on those boundaries. Moreover, placing the transformed images within an empty background (black) implicitly yields local contrast extrema between the image corners and that background. Examples of the resulting artifacts are visualized in Fig. 8.

To the best of our knowledge there is no clear way to avoid this issue directly during interest point detection. Thus, we discard features closer to the boundary of the transformed image than half of their radius in a separate step. Obviously, images that have been scaled in x- or y-direction only do not need post-processing.

For our synthetic query expansion and database augmentation scheme it is important and for the combination of both it is mandatory to discard such detections. The corresponding visual words do not carry useful information as they mostly describe black background. As a consequence, these lead to spurious false visual word correspondences between unrelated images which in turn deteriorate the retrieval. Once these detections are discarded the retrieval with features from warped images behaves as one would expect.

For SynQE multiple queries are issued to the index yielding multiple separate result lists. These are merged subsequently: images contained in multiple result lists get the maximum of each individual cosine similarity score as proposed in [1]. Similar for SynAUG: once a synthetic image is found it votes with its score for the original image and the maximum of all votes is taken as final similarity measure.

We test these techniques with a bag-of-words retrieval as described in Sect. 3.5.3 (RootSIFT, tf-idf-sqrt) and vocabularies of 1M, 2M and 3M words. The scaling parameter α is varied from 0.95 to 0.5 to test which group of transformations works best for simulating the perspective change in practice.

The corresponding results on the FlickrLogos dataset are shown in Fig. 9. Both SynQE and SynAUG improve the retrieval performance with a maximum at $\alpha = 0.7/0.8$. The combination of both, i.e. SynQE + SynAUG slightly increases the performance further. An even larger visual vocabulary of 2M words increases the performance dramatically over its baseline (11.6 %) but somewhat surprisingly only slightly above those of the vocabulary with 1M words.

The results on the Oxford5K dataset are shown in Fig. 10. Here, SynQE and SynAUG also improve retrieval performance though less pronounced. Consistently, SynAUG performs slightly better than SynQE. The performance of the vocabularies with 2M and 3M words increases dramatically over their baselines (bag-of-words without SynQE/SynAUG). While these do not perform better than the 1M vocabularies the most interesting behavior is that the performance of all vocabularies and methods seems to be "saturated" at around a mAP of 0.73. In other words, the actual choice of the vocabulary size has reduced impact. Thus, larger vocabularies that scale better to large image databases can be used with little loss of mAP. To summarize, the

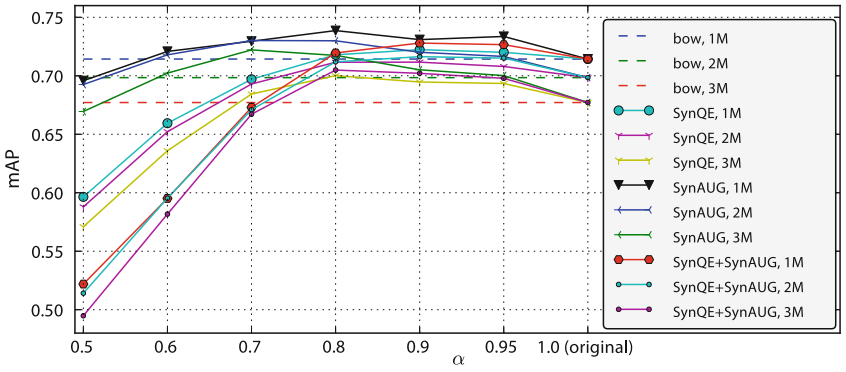

Fig. 10 *Oxford5K* impact of synthetic query expansion and database augmentation on bag-of-word retrieval performance

results on both datasets underline that discrete visual descriptions benefit from synthetic image generation—especially for small object retrieval such as logos.

6 Logo recognition

Now that we have discussed visual features, vocabularies, feature bundling, re-ranking and synthetic query expansion; we present our final logo recognition system:

Indexing The logo classes that our system should be able to detect are described by a set of images showing these logos in various poses. We refer to this set as *reference set* and use the images within the training and validation sets of the FlickrLogos-32 dataset for this purpose. Feature bundles are computed for each image in the reference set and inserted into the hash table associated with the information to which class a reference image belongs. Optionally, SynAUG is applied: artificially generated transformed versions of the original images are used to augment the reference set. In the following we refer to the transformation group with $\alpha = 0.7$ when referring to SynQE and SynAUG.

Testing An image is being tested for the presence of any of the logo classes by computing feature bundles and performing lookups in the hash table to determine the reference images that share the same bundles. The retrieved list of images is then re-ranked as described in Sect. 3.3. Optionally, SynQE may be applied: multiple transformed versions of the original query image are used to query the database multiple times or the database as described in Sect. 5. Afterwards the fast spatial re-ranking with 1P-WGC-RANSAC without projective refinement (see Sect. 4) is applied to the retrieved list. Finally an image is classified by a k-nn classifier: a logo of the class c is considered to be present if the majority of the top k retrieved images is of class c. In our experiments we chose $k = 5$.

Experimental setup We follow the evaluation protocol as in [24]: training and validation sets including non-logo images are indexed by the respective method. The whole test set including logo and logo-free images (3,960 images) is then used to compute the classification scores.

Table 4 FlickrLogos-32: logo recognition results

Method	Precision	Recall
Romberg et al. [24]	0.98	0.61
Revaud et al. [21]	≥0.98	0.73
Bag-of-words, 100K	0.988	0.674
Bag-of-words, 1M	0.991	0.784
Bag-of-words, 1M, SP	0.996	0.813
Bag-of-words, 1M, SP + SynQE	0.994	0.826
Bag-of-words, 1M, SP + SynAUG	0.996	0.825
BmH, 200K, collision count	0.688	0.411
BmH, 200K, CosSim	0.987	0.791
BmH, 1M, collision count	0.888	0.627
BmH, 1M, CosSim	0.991	0.803
BmH, 1M, CosSim + SP	0.996	0.818
BmH, 1M, SP only	0.996	0.809
BmH, 1M, CosSim + SP + SynQE	*0.999*	**0.832**
BmH, 1M, CosSim + SP + SynAUG	0.996	0.829

Bold values denote the best score

Results Table 4 shows the obtained results for various approaches. Revaud et al. [21] use a bag-of-words-based approach coupled with learned weights that down-weight visual words that appear across different classes. It can be seen that a bag-of-words-based search as described in Sect. 3.5.3 followed by 5-nn majority classification already outperforms this more elaborate approach significantly. In fact, our approach using bag-of-words to retrieve the logos and performing a majority vote among the top 5 retrieved images already outperforms the best results reported in the literature so far.

Bundle min-Hashing also outperforms the former scores *out of the box*. The difference between a ranking based on sketch collision counts ("collision count") and a ranking based on cosine similarity ("CosSim") makes clear that the result lists obtained by BmH must be re-ranked to ensure

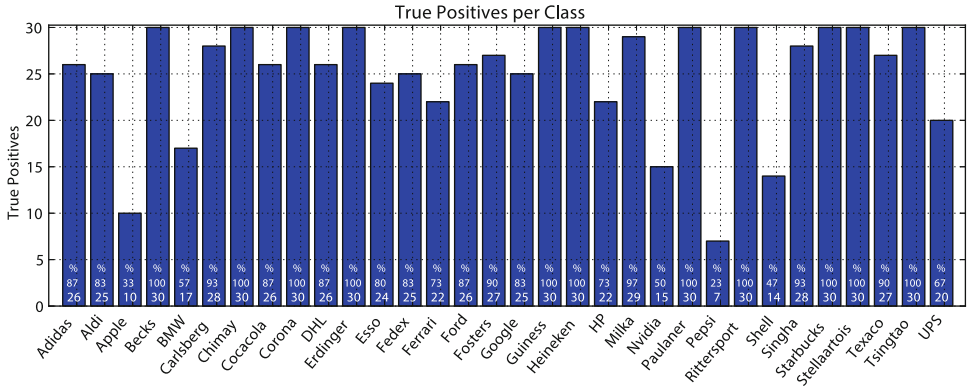

Fig. 11 True positives per class for the best performing system from Table 4 (BmH, 1M, CosSim + SP + SynQE)

Fig. 12 Logo detection by searching for similar bundles via Bundle min-hashing. *Left* local features (*blue circles*) of an image showing the Shell logo. *Middle* the bundles where Bundle min-Hashing (*without* SynQE, SynAUG or spatial re-ranking) found similar bundles associated with a certain class (*color-coded*) in the index. A few false positives are found in the background. *Right* the heat map shows the bundle hits visualized with a multi-scale multi-bandwidth Kernel density estimation incorporating both the scale of the bundles as well as the respective number of collisions. Due to the latter the false positive detections have negligible impact

Fig. 13 Examples of detections by Bundle min-Hashing for logos of the brands "Starbucks", "Coca-Cola" and "HP"

that the top-most images are indeed the most similar ones. We compared BmH with 200K words (highest mAP for BmH only, see Table 1) with a larger vocabulary of 1M words (slightly lower mAP). The preferable vocabulary of 1M words not only slightly improves the results but also reduces the complexity of the system as it eliminates the need for two different vocabularies for bundling and re-ranking. Moreover, the response ratio of this system is 100 times smaller (RR = 0.0096 for BmH with 1M words) than that of bag-of-words. Finally, it can be seen that both SynQE and SynAUG consistently improve the classification performance for both bag-of-words and Bundle min-Hashing. For completeness, the true positives per class for our best system are further shown in Fig. 11. In addition, we demonstrate how Bundle min-Hashing accurately localizes the logos in Figs. 12 and 13.

Failure cases While Bundle min-Hashing works remarkably well for a wide range of object types and object sizes, it is by definition dependent on the performance of local features. Thus, the chance of capturing the visual appearance of an object by aggregating multiple features decreases with decreasing number of detected features on the object. This is especially true for low-contrast or low-structured objects (e.g. the apple logo) where only a few local features are detected at best.

Table 5 Impact of index pruning: mAP on FlickrLogos for BmH configurations as shown in Table 1 when sketches occurring less than m times are removed from hash tables

Sketches	Original		Index pruning			
			$m = 2$		$m = 3$	
	mAP	Keys	mAP	Keys	mAP	Keys
4	0.554	20,311,553	0.558	749,303	0.559	147,382
3	0.545	15,234,817	0.549	561,755	0.549	110,143
2	0.527	10,156,293	0.529	374,437	0.528	73,609
1	0.478	5,078,358	0.475	187,228	0.471	36,577

Keys denotes the total number of sketches stored in hash tables

A further issue in practice is the non-distinctiveness of local features on image content such as text characters. Text usually generates many non-informative visual words yielding spurious matches and random collisions in the hash table. An example for this issue can be seen in the right-most image of Fig. 13. While Bundle min-Hashing is *much* less affected than, e.g., a regular bag-of-words voting scheme it is still not completely unimpaired giving raise to future improvements.

7 Index pruning

When analyzing the key-value distributions of the hash tables in our index we found that there were only $\approx$1.04 values stored per sketch on average. In other words, most of the sketches were generated only once. As the hash table lookup assumes that similar images share similar bundles and therefore sketches, we can further assume that if a sketch was generated only once from all images in the training + validation set, it will likely not have an impact when unknown images are tested for the presence of logos. Consequently, we remove those entries from our index resulting in roughly 30 times less items stored within the index.

The mAP after removing keys from the hash table that have less than m bundles is shown in Table 5. The results are rather surprising: the performance slightly improves even though the hash table then contains approximately 30 ($m = 2$) or 140 times ($m = 3$) fewer entries. Note that the number of remaining keys is even smaller than the visual vocabulary itself. Thus, the Bundle min-Hashing seems to serve as feature (pre-)selection technique; the features are then selected by their occurrence frequency. Effectively this scheme is a lossy yet highly effective index compression.

8 Conclusion

In this work, we described a robust feature bundling technique suitable for object retrieval and evaluated on several datasets. A logo recognition system based on finding local feature bundles in a database of reference images in combination with the new 1P-WGC-RANSAC variant for extremely fast re-ranking as well as synthetic query expansion and synthetic database augmentation significantly outperforms existing approaches. The results from index pruning give raise for future optimizations for scalability and memory consumption.

Acknowledgments This project was funded by Deutsche Forschungsgemeinschaft.

References

1. Arandjelović R, Zisserman A (2012) Multiple queries for large scale specific object retrieval. In: Proceedings of the British machine conference. BMVA Press, Dundee, pp 92.1–92.11
2. Arandjelović R, Zisserman A (2012) Three things everyone should know to improve object retrieval. In: Proceedings of IEEE international conference on computer vision and pattern recognition. pp 2911–2918
3. Bagdanov A, Ballan L, Bertini M, Del Bimbo A (2007) Trademark matching and retrieval in sports video databases. In: International workshop on multimedia information retrieval. pp 79–86
4. Cao Y, Wang C, Li Z, Zhang L (2010) Spatial-bag-of features. In: Proceedings of IEEE international conference on computer vision and pattern recognition. pp 3352–3359
5. Chum O, Perdoch M, Matas J (2009) Geometric min-Hashing: finding a (thick) needle in a haystack. In: Proceedings of IEEE international conference on computer vision and pattern recognition. pp 17–24
6. Chum O, Philbin J, Zisserman A (2008) Near duplicate image detection: min-hash and tf-idf weighting. In: Proceedings of the British machine conference, vol 1. BMVA Press, Dundee, pp 493–502
7. Fu J, Wang J, Lu H (2010) Effective logo retrieval with adaptive local feature selection. In: Proceedings of ACM international conference on multimedia. pp 971–974
8. Jégou H, Douze M, Schmid C (2009) Improving bag-of-features for large scale image search. Int J Comput Vis 87(3):316–336
9. Jegou H, Douze M, Schmid C (2009) On the burstiness of visual elements. In Proceedings of IEEE international conference on computer vision and pattern recognition. pp 1169–1176
10. Jegou H, Douze M, Schmid C (2009) Packing bag-of-features In: Proceedings of IEEE international conference on computer vision. pp 2357–2364
11. Jiang Y, Meng J, Yuan J (2011) Grid-based local feature bundling for efficient object search and localization. In: Proceedings of international conference on image processing. pp 113–116

12. Joly A, Buisson O (2009) Logo retrieval with a contrario visual query expansion. In: Proceedings of ACM international conference on multimedia. pp 581–584
13. Kalantidis Y, Pueyo L G, Trevisiol M, Van Zwol R, Avrithis Y (2011) Scalable Triangulation-based Logo Recognition. In: Proceedings of ACM international conference on multimedia retrieval (article 20)
14. Lebeda K, Matas J, Chum O (2012) Fixing the Locally Optimized RANSAC. In: Proceedings of the British machine conference. BMVA Press, Dundee, pp 95.1–95.11
15. Lee D, Ke Q, Isard M (2010) Partition min-Hash for partial duplicate image discovery. In: Proceedings of ACM European conference on computer vision. pp 648–662
16. Letessier P, Buisson O, Joly A (2011) Consistent visual words mining with adaptive sampling. In: Proceedings of ACM international conference on multimedia retrieval (art. 49)
17. Mikolajczyk K, Schmid C (2004) Scale & affine invariant interest point detectors. Int J Comput Vis 60(1):63–86
18. Muja M, Lowe D (2009) Fast approximate nearest neighbors with automatic algorithm configuration. In: International conference on computer vision theory and application. pp 331–340
19. Nistér D, Stewénius H (2006) Scalable recognition with a vocabulary tree. In: Proceedings of IEEE international conference on computer vision and pattern recognition. pp 2161–2168
20. Philbin J, Chum O, Isard M, Sivic J, Zisserman A (2007) Object retrieval with large vocabularies and fast spatial matching. In: Proceedings of IEEE international conference on computer vision and pattern recognition. pp 1–8
21. Revaud J, Douze M, Schmid C (2012) Correlation-based burstiness for logo retrieval. In: Proceedings of ACM international conference on multimedia. pp 965–968
22. Romberg S, August M, Ries CX, Lienhart R (2012) Robust Feature Bundling. In: Advances in multimedia information processing—PCM 2012. Lecture notes in computer science, vol 7674. pp 45–56
23. Romberg S, Lienhart R (2013) Bundle min-Hashing for logo recognition. In: Proceedings of ACM international conference on multimedia retrieval
24. Romberg S, Pueyo LG, Lienhart R, van Zwol R (2011) Scalable logo recognition in real-world images. In: Proceedings of ACM international conference on multimedia retrieval (article 25)
25. Sivic J, Zisserman A (2003) Video Google: a text retrieval approach to object matching in videos. In: Proc. of IEEE international conference on computer vision, vol 2, pp 1470–1477.
26. Wu Z, Ke Q, Isard M, Sun J (2009) Bundling features for large scale partial-duplicate web image search. In: Proceedings of IEEE international conference on pattern recognition. pp 25–32
27. Yu G, Morel J (2009) A fully affine invariant image comparison method. In: Proceedings of IEEE conference on acoustics, speech and signal processing. pp 1597–1600
28. Zhang S, Tian Q, Hua G, Huang Q, Li S (2009) Descriptive visual words and visual phrases for image applications. In: Proceedings of ACM international conference on multimedia. pp 75–84