Bundle min-hashing for logo recognition

Stefan Romberg, Rainer Lienhart

Angaben zur Veröffentlichung / Publication details:

Romberg, Stefan, and Rainer Lienhart. 2013. “Bundle min-hashing for logo recognition.” In *Proceedings of the 3rd ACM conference on International conference on multimedia retrieval - ICMR ’13, April 16 - 20, 2013, Dallas, Texas, USA*, 113–20. New York, USA: ACM Press. https://doi.org/10.1145/2461466.2461486.

Bundle Min-Hashing for Logo Recognition

Stefan Romberg
Multimedia Computing and Computer Vision Lab
Augsburg University
Augsburg, Germany
romberg@informatik.uni-augsburg.de

Rainer Lienhart
Multimedia Computing and Computer Vision Lab
Augsburg University
Augsburg, Germany
lienhart@informatik.uni-augsburg.de

ABSTRACT

We present a scalable logo recognition technique based on feature bundling. Individual local features are aggregated with features from their spatial neighborhood into bundles. These bundles carry more information about the image content than single visual words. The recognition of logos in novel images is then performed by querying a database of reference images. We further propose a novel WGC-constrained RANSAC and a technique that boosts recall for object retrieval by synthesizing images from original query 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%.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; I.5.4 [**Pattern Recognition**]: Computer Vision

General Terms

Algorithms, Experimentation

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; their high-dimensional descriptor vectors are clustered and quantized into individual integer numbers - called visual words. An image is then modeled as an unordered collection of word occurrences, commonly known as bag-of-words. This description provides an enormous data reduction compared to the original descriptors. Its benefits are a fixed-size image description, robustness to occlusion and viewpoint changes, and eventually simplicity, i.e. a small computational complexity.

It has been observed 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. Inspired by this observation we exploit a feature bundling technique that builds on visual words, but aggregates spatial neighboring visual words into feature bundles. An efficient search technique for such bundles based on min-hashing 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 exploits a bundle representation that retrieves approximately 100 times fewer images than bag-of-words while having equal recall performance. The core components of our logo recognition system are:

- We adopt the retrieval technique of [15] based on feature bundles to the problem of logo recognition and show that such a system significantly outperforms the current state-of-the-art.
- A *1*P-WGC-RANSAC variant for fast (real-time) spatial re-ranking is proposed that yields superior results

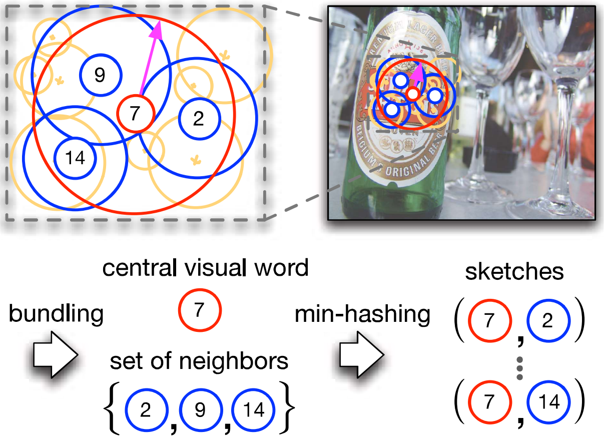

Figure 1: Bundle Min-Hashing: The neighborhood around a local feature, the *central feature* (red), is described by a feature bundle. Features that are too far away or on scales too different from that of the central feature are ignored during the bundling (yellow). The features included in such a bundle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

ICMR'13, April 16–20, 2013, Dallas, Texas, USA.
https://doi.org/10.1145/2461466.2461486

compared to existing approaches by exploiting a weak-geometric constraint to speed up the computation

- We demonstrate that recall of a system targeting high precision for small object retrieval can be increased by exploiting synthetically generated images both for query expansion as well as database augmentation.

2. RELATED WORK

We present related work suited to image and object retrieval. As our approach is based on min-hashing, we also briefly highlight the related work relevant in the context of min-hashing.

Visual Words and Bundling. An early approach to feature bundling was used as a simple post-retrieval verification step where the number of matching neighboring features was exploited to discriminate true feature matches from random matches [17]. Later it was proposed to bundle multiple SIFT features that lie in the same MSER region into a single description [18]. The authors then defined a weak geometric similarity criterion. However, this work used single visual words for lookups in the inverted index, the bundles and the weak geometric similarity are used for post-retrieval verification only. In [3] the most informative projections that map the visual words from the 2-D space into feature histograms (termed "spatial bag-of-words") are learned. An approach, which is similar yet more unbiased to certain image layouts, splits the original feature histograms by random projections into multiple smaller "mini bag-of-features" [8]. Separate lookups and an aggregating scoring are used to find the most similar images in an image database. Another approach bundles triples of visual words including their spatial layout into visual signatures that are then subsequently indexed by a cascaded index making testing of images for the presence of pairs and triples possible and efficient [16].

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 [5]. 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 in Section 3.1.

Geometric min-hashing (GmH). A conceptually similar approach to ours is geometric min-hashing [4]. 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 [4]. These clusters are then 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 [10] 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. However, in [15] we observed that PmH performs worse than mH on the logo dataset.

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 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, i.e. higher precision.

3.1 Min-hashing (mH)

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 a set of l visual words of an image $I = \{v_0, ..., v_{l-1}\}$, the min-hash function is defined as

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

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:

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

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 alone 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. To improve recall, this process is repeated n times and 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

$$P(collision) = 1 - (1 - ovr(I_1, I_2)^k)^n \qquad (3)$$

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 [10]. 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 table offer access 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 k-th sketch is inserted into the k-th 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 simply 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 a region of interest.

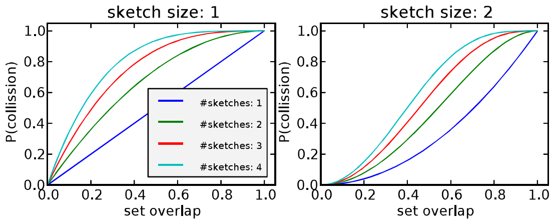

Figure 2: Collision probabilities given the set overlap between bundles. Left: Single min-hash (as used by Bundle Min-Hashing). Right: Sketches of size 2.

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 $\mathbf{x}_i$:

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

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:

$$W_i(b(\mathbf{x}_i)) = \{\ q(\mathbf{x}_j)\ |\ \mathbf{x}_j \in b(\mathbf{x}_i)\} \qquad (5)$$

The resulting set W_i is then subsequently 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 n-th sketch is given as 2-tuple

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

where v_i denotes the visual word label of the central feature and mh_n denotes the min-hash returned by the n-th 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 Figure 1.

The major advantage can be seen when comparing the collision probabilities of a single min-hash and sketches of size 2 (see Figure 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 contain multiple bundles that may overlap. If some bundles do not match (collide) across images, there is the chance that other bundles in the same images collide.

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 [6]. It also rules out features without good neighbors and decreases the number of bundles even below the number of local features in an image. 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).

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 once all coordinates are indexed.

Also, 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.

Adjustable Search. The representation of bundles 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 $1...k$. As the sketch collision probability is proportional to the set overlap, sets (=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 i-th sketch was inserted into the i-th hash table, querying sketches from $1...i$ will yield only bundles were the corresponding sketches and hash functions in tables $1...i$ 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.

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 [17] obtained with a vocabulary of 1M words (see Section 3.4.3). We also address the problem of visual word burstiness by taking the square root of each tf-idf histogram entry as proposed in [7]. 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.

3.4 Experiments

3.4.1 Dataset

The dataset we chose to evaluate our logo retrieval approach is FlickrLogos-32. It consists of 32 classes of brand logos [16]. Compared to other well-known datasets suited for image retrieval, e.g. Oxford buildings, images of a similar class in FlickrLogos-32 share much smaller visually similar regions. For instance, 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%) of the whole image. 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 disjunct subsets. For each logo class, we have 10 train images, 30 validation images, and 30 test images - each containing at least one instance of the respective logo. For both validation and test set the dataset also provides a set of 3000 negative (logo-free) images.

This logo dataset is interesting for the evaluation of small object retrieval and classification since it features logos that can be considered as rigid 2-D 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 tilt and for classification eventually the task of multi-class recognition.

Our evaluation protocol is as follows: All images in the training and validation set, including those that do not contain any logo are indexed by the respective method (In total: 4280 images). The 960 images in the test set which do show a logo (given by the ground truth) are then used as queries to determine the most similar images from the training and validation set. The respective retrieval results are then ranked by the cosine similarity (see Section 3.3).

3.4.2 Visual Features

For all of our experiments we used SIFT descriptors computed from interest points found by the Difference-of-Gaus sian detector. To quantize the descriptor vectors to 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 [12]. This kd-forest then allows to perform approximate nearest neighbor search to find the nearest cluster for a descriptor vector both during clustering as well as when quantizing descriptor vectors to single visual words. The vocabulary and IDF weights have been computed with data from the training and validation set of FlickrLogos-32 only.

3.4.3 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. A system will only gain high mAP

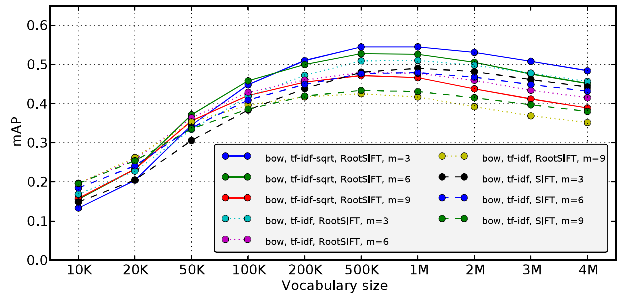

Figure 3: Retrieval score (mAP) for several bag-of-words variants on the FlickrLogos-32 dataset.

scores if both precision and recall are high. Here, the AP is computed as $AP = \sum_{i=1}^{N} \frac{1}{2}(P_i + P_{i-1}) \cdot (R_i - R_{i-1})$ with $R_0 = 0, P_0 = 1$ where P_i, R_i denote precision/recall at the i-th position in the retrieved list.

The response ratio (RR) measures the retrieval efficiency. It describes the number of retrieved images in relation to the database size. The higher the response ratio the more images are 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.

Bag-of-words. As baseline on this particular dataset we show the performance of approaches based purely on the cosine similarity between bag-of-words. 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 Figure 3. In [15] 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 vocabulary show significantly better performance. The peak is consistently at 500K/1M words. The patch size that is described by a SIFT descriptor depends on the scale but is also controlled by a magnification factor m. We further test how this magnifier changes the performance. The best performance is obtained with descriptors computed with a magnifier of $m = 3$ as in Lowe's work. In addition we compare the performance of bag-of-words based on standard 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 [7] 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 Figure 3) as it down-weights repeating and thus less informative visual words ("bursts").

For further experiments 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. For re-ranking the best-performing vocabulary of 1M words is used, disregarding which vocabulary was used when building the feature bundles.

Feature Bundles. 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 each of these 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. More importantly, the bundles are on par with bag-of-words, but have an order of magnitude lower response ratio (RR) as shown in Table 1.

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 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.

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 [13]. 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 hypothesis with the highest score (determined by the symmetric transfer error and truncated quadratic cost function as in [9]) are kept for further refinement. If the top hypothesis have more than 15 inliers these are then refined by a local optimization (LO) step that estimates a fully projective transformation between images 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 the projective re-estimation. Moreover, while this refinement improves the visual quality of the estimated transformation it has little effect on the induced ranking. 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.

For 1P-WGC-RANSAC, a weak geometric consistency constraint (WGC) 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. However, it acts as filtering 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

#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	2468.1 words	1.0
bag-of-words, tf-idf-sqrt weighting				500K	0.545	3.06	0.011	0.932	0.845	2468.1 words	1.0
bag-of-words, tf-idf-sqrt weighting				1M	0.545	**3.16**	0.012	0.911	0.763	2468.1 words	1.0
4	0.5	2.0	1.0	200K	**0.554**	3.14	0.317	0.639	0.025	1640.9	2.66
3	0.5	2.0	1.0	200K	0.545	3.13	0.338	0.623	0.022	1640.9	1.99
2	0.5	2.0	1.0	200K	0.527	3.09	0.367	0.592	0.018	1640.9	1.33
1	0.5	2.0	1.0	200K	0.479	3.04	**0.423**	0.520	**0.012**	1640.9	**0.66**

Table 1: Comparison of bag-of-words retrieval with Bundle Min-Hashing: The upper part shows three different bag-of-words retrieval runs with corresponding scores. The lower part contains 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 Section 3.2) and the vocabulary size. The scores follow in the order of mean Average Precision, average top 4 score, mean precision, mean recall and response ratio. The column *⌀#bundles* denotes the average number of bundles stored in the hash table per image. The last column shows the number of hash table entries (sketches) relative to the number of visual words per image.

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. [13] and Arandjelovic et al. [2] on the Oxford5K dataset [13] following the common test protocol: The top 1000 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) yields slightly higher (1M words) or even significantly higher scores (100K words) than that of Philbin et al. [13] (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.

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: reranked 5813 images ≈ 440 images/s ≈ 2.3 ms per image, single-threaded, including I/O) making it suitable for real-time applications.

Method	Voc	mAP	Time
Philbin et al.[13], bow	100K	0.535	–
Philbin et al.[13], bow+SP	100K	0.597	–
bow, tf-idf, SIFT	100K	0.571	–
1P-RANSAC, incl. LO	100K	0.678	160s
1P-RANSAC, no LO	100K	0.680	72s
1P-WGC-RANSAC, incl. LO	100K	0.693	115s
1P-WGC-RANSAC, no LO	100K	0.692	53s
Philbin et al.[13], bow	1M	0.618	–
Philbin et al.[13], 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	54s
1P-RANSAC, no LO	1M	0.711	15s
1P-WGC-RANSAC, incl. LO	1M	0.704	50s
1P-WGC-RANSAC, no LO	1M	0.703	12s
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	92s
1P-RANSAC, no LO	1M	0.729	17s
1P-WGC-RANSAC, incl. LO	1M	0.723	55s
1P-WGC-RANSAC, no LO ★	1M	0.723	13s

Table 2: Comparison of spatial re-ranking results for the Oxford5K dataset following the protocol in [13].

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

Table 3: FlickrLogos-32: Spatial re-ranking results.

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

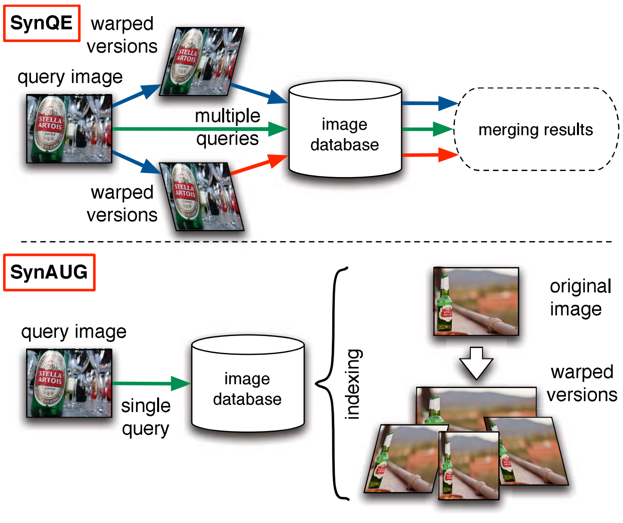

Figure 4: Top: Synthetic query expansion. Bottom: Synthetic database augmentation.

identical but seen 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 Figure 4.

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 method is illustrated in the lower half of Figure 4.

3. *SynQE + SynAUG*: The combination of (1) and (2). This can be seen as counterpart to ASIFT [11] working with discrete visual words and an inverted index or another database instead of comparing raw descriptors between two images.

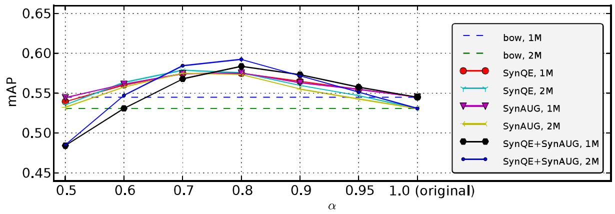

Figure 5: FlickrLogos-32: Impact of synthetic query expansion and database augmentation on bag-of-word retrieval performance.

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.

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 Section 3.4.3 (RootSIFT, tf-idf-sqrt) and a vocabulary of 1M and 2M 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 are shown in Figure 5.

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.

To summarize, the obtained results underline that discrete visual descriptions benefit from synthetic image generation. In the following we refer to the transformation group with $\alpha = 0.7$ when referring to SynQE and SynAUG.

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 set 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 to the reference set.

[1] These are equivalent to the two shearings along y-direction.

Method	Precision	Recall
Romberg et al. [16]	0.98	0.61
Revaud et al. [14]	≥ 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

Table 4: FlickrLogos-32: Logo recognition results.

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 Section 3.3. Optionally, SynQE and SynAUG may be applied: Multiple transformed versions of the original query image are used to query the database multiple times or the database is augmented with synthetic images as described in Section 5. Afterwards the fast spatial re-ranking with 1P-WGC-RANSAC without projective refinement (see Section 4) is applied to the retrieved list. Finally a logo instance 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. The evaluation protocol is identical to that in [16]: The training and validation set including non-logo images are indexed by the respective method. The whole test set including logo and logo-free images (3960 images) is then used to compute the classification scores.

Results. Table 4 shows the obtained results for various approaches. Revaud et al. use a bag-of-words-based approach coupled with learned weights that down-weight visual words that appear across different classes [14]. It can be seen that a bag-of-words-based search as described in Section 3.4.3 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 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 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 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. As there is actually little difference SynAUG is the preferred method as the database augmentation can be performed off-line.

7. CONCLUSION

In this work we introduced a robust logo recognition technique based on finding local feature bundles in a database of reference images. This approach 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.

8. REFERENCES

[1] R. Arandjelovic and A. Zisserman. Multiple queries for large scale specific object retrieval. In *BMVC*, 2012.

[2] R. Arandjelovic and A. Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, 2012.

[3] Y. Cao, C. Wang, Z. Li, and L. Zhang. Spatial-bag-of-features. In *CVPR*, 2010.

[4] O. Chum, M. Perdoch, and J. Matas. Geometric min-Hashing: Finding a (thick) needle in a haystack. In *CVPR*, 2009.

[5] O. Chum, J. Philbin, and A. Zisserman. Near duplicate image detection: min-hash and tf-idf weighting. In *BMVC*, 2008.

[6] H. Jégou, M. Douze, and C. Schmid. Improving Bag -of-Features for Large Scale Image Search. *IJCV*, 2009.

[7] H. Jegou, M. Douze, and C. Schmid. On the burstiness of visual elements. In *CVPR*, 2009.

[8] H. Jegou, M. Douze, and C. Schmid. Packing bag-of-features. In *ICCV*, 2009.

[9] K. Lebeda, J. Matas, and O. Chum. Fixing the Locally Optimized RANSAC. In *BMVC*, 2012.

[10] D. Lee, Q. Ke, and M. Isard. Partition Min-Hash for Partial Duplicate Image Discovery. *ECCV*, 2010.

[11] J. Morel and G. Yu. ASIFT: A New Framework for Fully Affine Invariant Image Comparison. *SIAM*, 2009.

[12] M. Muja and D. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP*, 2009.

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

[14] J. Revaud, C. Schmid, M. Douze, and C. Schmid. Correlation-Based Burstiness for Logo Retrieval. In *ACM MM*, 2012.

[15] S. Romberg, M. August, C. X. Ries, and R. Lienhart. Robust Feature Bundling. In *LNCS*, 2012.

[16] S. Romberg, L. Garcia Pueyo, R. Lienhart, and R. van Zwol. Scalable Logo Recognition in Real-World Images. In *ICMR*, 2011.

[17] J. Sivic and A. Zisserman. Video Google: a text retrieval approach to object matching in videos. *ICCV*, 2003.

[18] Z. Wu, Q. Ke, M. Isard, and J. Sun. Bundling features for large scale partial-duplicate web image search. In *CVPR*, 2009.