Robust Feature Bundling

Stefan Romberg[⋆], Moritz August, Christian X. Ries, and Rainer Lienhart

Multimedia Computing and Computer Vision Lab, Augsburg University
{romberg,ries,lienhart}@informatik.uni-augsburg.de
http://www.multimedia-computing.de

Abstract. In this work we present a feature bundling technique that aggregates individual local features with features from their spatial neighborhood into bundles. The resulting bundles carry more information of the underlying image content than single visual words. As in practice an exact search for such bundles is infeasible, we employ a robust approximate similarity search with min-hashing in order to retrieve images containing similar bundles.

We demonstrate the benefits of these bundles for small object retrieval, i.e. logo recognition, and generic image retrieval. Multiple bundling strategies are explored and thoroughly evaluated on three different datasets.

1 Introduction

In computer vision, the bag-of-visual words approach has been very popular in recent years. Hereby, an image is described by multiple local features; their high-dimensional descriptor vectors are clustered and quantized to a single integer number - called visual word - that represents the cluster center. An image is then usually modeled as an unordered collection of word occurrences, the so-called bag-of-words. This description provides an enormous data reduction compared to the original descriptor vectors. Its benefits are a fixed-size image description, robustness to occlusion and viewpoint changes and eventually simplicity, i.e. small computational complexity.

In this work we describe a novel approach that builds on visual words and aggregates spatially close visual words into bundles. Such bundles are more distinctive than individual visual words alone, i.e. objects and image regions are described with more expressiveness. We propose a robust method for approximate similarity search for such bundles, with performance close to the standard bag-of-words method, but with higher precision and much lower response ratio, i.e. less false positives.

Two different bundling strategies are evaluated thoroughly on three different dataset and compared to bag-of-words retrieval and two recent min-hashing approaches. We show that the retrieval using feature bundles yields similar performance as standard bag-of-words retrieval and outperforms two other min-hash-based approaches while providing a response ratio as low as the latter.

[⋆] This project is funded by Deutsche Forschungsgesellschaft (DFG).

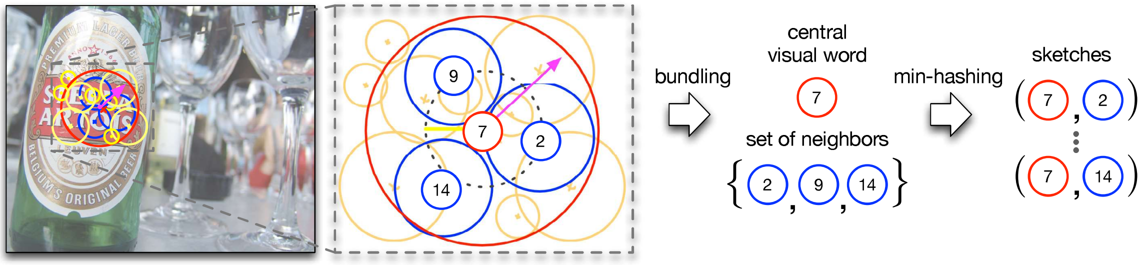

Fig. 1. Feature bundles: The neighborhood around a local feature, the *central feature* (red), is described by a feature bundle. Features that are too far away or on scales too different from that of the central feature are ignored during the bundling (yellow). The features included in such a bundle (blue) are represented as set of visual word occurrences and indexed by min-hashing (see Section 4).

2 Motivation

It has been observed several times that the retrieval performance of bag-of-word-based methods improves much more by reducing the number of mismatching visual words, e.g. by large vocabularies or Hamming Embedding [5], than by reducing quantization artifacts. 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 our contribution is a feature bundling technique that builds on bag-of-words but does not describe each visual word individually but rather aggregates the spatial neighboring visual words into feature bundles. We propose an efficient indexing and search technique for such bundles based on min-hashing, that allows for similarity search without requiring exact matching.

As illustrated in Figure 1 on the left, we combine a visual word with its spatially neighboring visual words into a bundle in order to obtain a more expressive description of the respective image region.

Such bundles carry more information than individual visual words. Thus, we expect that more false positives are suppressed during the retrieval and the returned result set is much smaller and cleaner, compared to traditional bag-of-words. Small result sets are beneficial because expensive post-retrieval steps only need to be applied to a small number of images.

3 Related Work

As the core of our approach is based on min-hashing, we briefly highlight the related work on min-hashing relevant in the context of our approach.

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 [3] and extended to the approximation of weighted set overlap as well as histogram intersection [4]. In each of these settings an image is modeled 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 or simply sets representing whether a visual word is present or not.

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

Given the set of l visual words $I = \{v_0, ..., v_{l-1}\}$ of an image, 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 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 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

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

This probability function depends on the set overlap and in practice the overlap between non-near-duplicate images that still show the same object is very close to 0. In fact, the average overlap for a large number of partial near-duplicate images was reported to be 0.019 in [6]. This makes clear 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 very close to 0.

The indexing of sets and the approximate search are then 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 similar sets for a query set, one simply computes the corresponding sketches and searches for these sets in the database, that have one or more sketches in common with the query. This is performed by doing lookups of each query sketch and determining 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 > 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, i.e. the k-th sketch is inserted into the k-th hash table.

Geometric min-hashing (GmH). A conceptually similar approach to ours is geometric min-hashing [2]. However, its statistical pre-conditions 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. In summary, this makes GmH very memory efficient, but not suitable for generic image retrieval because of bad recall. Consequently, the authors use it to quickly retrieve images from a large database in order to build initial clusters of highly similar images [2][1]. 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 [6] a scheme is introduced that divides the image into several partitions. Unlike the global min-hashing (mH), min-hashes and sketches are computed for all partitions independently. The search then proceeds by determining the sketch collisions for each of the partitions. As the partitions may overlap and are processed step by step this scheme is conceptually similar to a sliding window search. The authors show that this scheme has identical collision probabilities for sketches as mH in the worst case, but better recall and precision if the duplicate image region only covers a small area. Furthermore PmH is significantly faster than mH. We include PmH in our evaluation and find that it performs not significantly better than mH on our dataset.

4 Feature Bundles

We build our bundling technique on min-hash mainly for two reasons: (1) Feature bundles can be naturally represented as sparse sets and (2) min-hash 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 detections, and their quantizations 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.

We consider the proposed bundling technique 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 with much lower response ratio and therefore higher precision.

4.1 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.

More specifically, given a feature $\mathbf{x}_i$ its corresponding feature bundle $b(\mathbf{x}_i)$ is then 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 $\mathbf{x}_i$ for which we propose two different definitions in the following 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:

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

The resulting set W_i is then subsequently indexed by min-hashing which samples min-hashes based on the corresponding hash functions from this set and indexes them as sketches.

In extensive experiments we observed the following: First, sketches of size ≥ 3 do not work very well, therefore we perform all our experiments with sketches of size 2. Second, we found that the overall performance increases drastically if the first sketch element is not determined by min-hash 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

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

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 also illustrated in Figure 1.

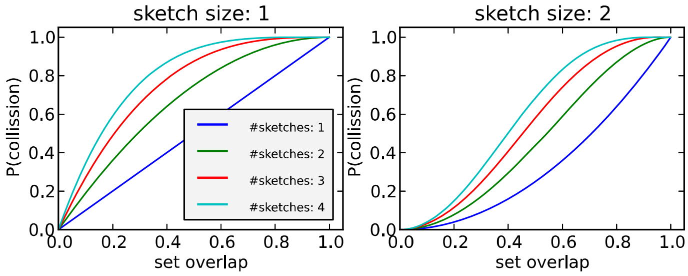

Fig. 2. Collision probabilities given the set overlap between bundles. Left: Collision probability for a single min-hash. Right: Sketches of size 2.

In Figure 2 the collision probabilities of sketches of size 1 (a single min-hash) and size 2 given 1 to 4 sketches are shown. One can see that even two bundles with an overlap of only 0.2, have a 0.5 chance to have one of 4 sketches colliding. 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. Throughout our experiments we therefore describe each feature bundle by 4 sketches, limiting the overall memory requirement to at most 4 times the storage of bag-of-words.

4.2 Bundling Strategies

In this section we introduce two strategies to select the features around a central feature which are then combined into a feature bundle. Each feature $\mathbf{x}_i$ in an image that has at least 2 features in its neighborhood $N(\mathbf{x}_i)$ is used to compute a feature bundle. Features with less or no neighbors are ignored.

Strategy 1: Bundles of Equal Area. The first bundling strategy is based on the intuition that features which are spatially close to each other might describe the same object. That is, given a central feature we bundle it with its direct spatial neighbors. We do not induce any further constraints except requiring that all features of a bundle must be on a similar scale. This is in line with the observation that true feature correspondences are often the same scale [5]. Thus, each feature that is closer to a given central feature $\mathbf{x}_i$ than a given cut-off radius is included in the respective bundle $b(\mathbf{x}_i)$:

$$N_{md}(\mathbf{x}_i) = \{\mathbf{x}_j \mid \|\mathbf{p}_i - \mathbf{p}_j\| \leq s_i \cdot r_{max},\ s_{min} \cdot s_i \leq s_j \leq s_{max} \cdot s_i\} \tag{7}$$

Here, $\mathbf{p}_i$ denotes the location of the feature in the 2-D image plane and s_i denotes the corresponding scale determined by the interest point detector. The scale is linked with the patch size that is described by the descriptor. For simplicity we assume that s_i denotes the radius of the patch in pixels. Only those neighboring features are included in the bundle which are closer to the central feature than the maximum distance r_{max} relative to the scale i.e. patch size of the central feature s_i. The minimum and maximum scales s_{min} and s_{max} control the scales considered for determining the neighborhood relative to the scale of the central feature. Figure 1 shows the bundling criterion for $r_{max} = 0.5$ (dashed gray circle), $s_{min} = 0.25$ and $s_{max} = 1.0$.

Strategy 2: Bundles of Equal Size. In this strategy, the neighborhood of a bundle is not determined by the size or scale of the central feature. Instead, the neighborhood is chosen such that it includes exactly the m visual words which are closest to the respective central feature and on a scale in between s_{min} and s_{max} relative to the scale of the central feature.

This definition is based on the assumption that image regions showing the same content in different images will yield roughly the same number of feature detections. Most importantly, this neighborhood definition has the major advantage that all bundles are of equal size, i.e. the overlap between these bundles will be easily comparable. The redundancy and the robustness of the min-hash-based search for bundles deals with missing or additionally included outlier features and still retrieves similar bundles.

4.3 Ranking and Filtering

As mentioned above, we use min-hashing in order to find images which share similar bundles with the query image. Once these images are determined, they may be ranked by their similarity to the query image. In preliminary experiments we evaluated several ways to compute a similarity score between query and retrieved images, based on the number of sketch collisions or number of matching bundles, either as absolute value or normalized in various ways. It turns out that the simple absolute count of sketch collisions was always on par with more complex similarity measures.

However, a ranking based on the cosine similarity between the full bag-of-words histogram of the query image and retrieved images still performs significantly better than a ranking based on the sketch collision counts only. Thus, in our experiments we rank all retrieval results by the cosine similarity between the bag-of-words histograms describing the full image that have been obtained with the same vocabulary size as used for bundling.

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. Subsequently, these images are then ranked by the cosine similarity. The small response ratio of the retrieval with bundles is a major benefit: Small result sets may be processed quickly even with more elaborate re-ranking methods.

5 Experiments

For all of our experiments we used SIFT descriptors as visual features computed from interest points found by the Difference-of-Gaussian (DoG) 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. Here, we use a forest of 8 randomized kd-trees to index the visual word centers [7]. 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. To avoid overfitting, the clustering was performed with data from the training and validation set of FlickrLogos-32 only.

5.1 Dataset and Evaluation Method

We evaluate our approach on three different datasets: FlickrLogos-32 [10], UkBench [8] and Oxford [9]. We use the FlickrLogos-32 dataset to perform parameter sweeps and optimization of our approach and compare the performance of some selected well-performing configurations to several baselines. Then the bundling is evaluated with unchanged configurations - without further tuning - on both the UkBench and the Oxford dataset to demonstrate how this technique generalizes.

As a retrieval system should have both good precision and good recall, we measure the retrieval performance as the mean average precision (mAP) which describes the area under the precision recall curve. It characterizes both aspects; a system will only gain high mAP scores if both precision and recall are high.

The response ratio (RR) is then 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 are in the result set, which is usually post-processed or verified by computationally expensive methods. A low response ratio will therefore increase the overall efficiency of the search.

The retrieval on the UkBench dataset is measured by the average top 4 score (Top4), defined as the average number of correctly retrieved images among the top 4 results. A perfect retrieval would retrieve 4 correct top-ranked images and therefore yield a score of 4.0. We also report this score where appropriate.

5.2 FlickrLogos-32

The first dataset we use is FlickrLogos-32 (FlickrLogos) which is a recently published dataset consisting of 32 classes of brand logos [10]. Compared to other well-known datasets suited for image retrieval, e.g. Oxford, 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 (derived from groundtruth annotation) 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

Table 1. Retrieval results on the FlickrLogos dataset obtained with min-hash (left) and Partition min-hash (right). $100K/1M$: visual vocabulary size, k: sketch size, n: number of sketches, p number of partitions, np: number of sketches per partition. The overlap of the p partitions was 0.5 in all Partition min-hash runs.

Min-hash	mAP	Top4	RR
100K k: 2 n: 128	0.072	1.56	0.0155
100K k: 2 n: 256	0.113	1.97	0.0303
100K k: 2 n: 512	0.178	1.32	0.0553
100K k: 2 n: 1024	0.256	2.49	0.1011
1M k: 2 n: 128	0.036	0.96	0.0007
1M k: 2 n: 256	0.059	1.37	0.0012
1M k: 2 n: 512	0.098	1.78	0.0020
1M k: 2 n: 1024	0.142	2.17	0.0036

Partition min-hash	mAP	Top4	RR
100K k: 2, p: 4, np: 256	0.243	2.47	0.0675
100K k: 2, p: 16, np: 64	0.235	2.44	0.0457
100K k: 2, p: 64, np: 8	0.150	2.23	0.0327
100K k: 2, p: 64, np: 16	0.221	2.44	0.0623
1M k: 2, p: 4, np: 256	0.150	2.30	0.0037
1M k: 2, p: 16, np: 64	0.152	2.41	0.0037
1M k: 2, p: 64, np: 8	0.108	2.09	0.004
1M k: 2, p: 64, np: 16	0.167	2.54	0.0077

is 9% (median: 5%) of the whole image. As the retrieval of the Oxford building is sometimes coined “object retrieval”, the retrieval task on the FlickrLogos dataset can be truly considered “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 (non-logo) images downloaded from Flickr by the query terms “building, ”“friends”, “nature” and “people”. This dataset of logos is interesting for both retrieval and classification since it features logos which can be considered as rigid objects with approximate 2-D planar surface visible from a single viewpoint only. The difficulty arises from the great variance of object sizes, from tiny logos in the background to image-filling views.

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). These 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 4.3).

We evaluate the retrieval performance of all approaches for varying vocabulary sizes. As we are especially interested in the impact of extremely large visual vocabularies on the overall performance, we vary the vocabulary sizes from 10,000 (10K) to 4,000,000 (4M) words for all of our experiments.

Min-Hash and Partition Min-Hash. We compare the performance of our approach to the performance of the standard min-hashing approach (mH) as well as our Partition min-hash (PmH) implementation. These approaches are specifically meant for near-duplicate and partial-near-duplicate image search. This comparison shows how these methods perform for small object search on the FlickrLogos dataset when used with typical parameters.

Table 1 lists the obtained results for typical parameter constellations. From the results it can be seen that both min-hash and Partition min-hash show reasonable

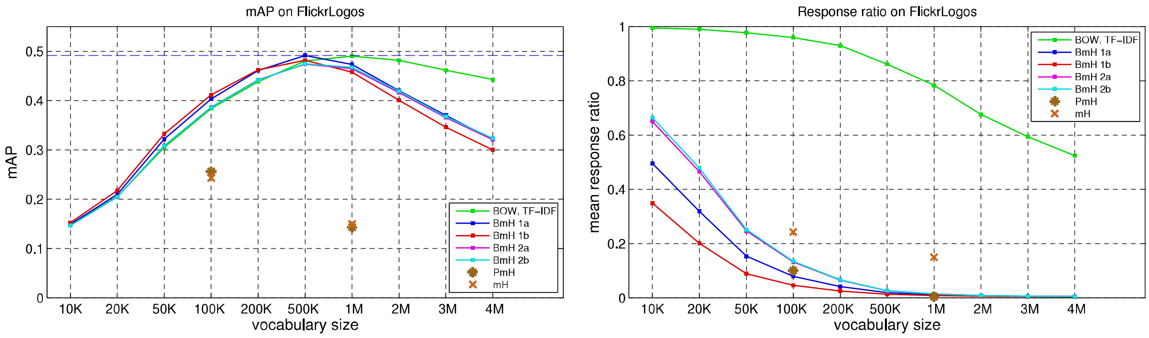

Fig. 3. Retrieval results on the FlickrLogos-32 dataset: The performance of the bundles is on par with the bag-of-words model (left) but the response ratio is an order of magnitude lower (right)

Table 2. Selected bundle configurations

Name	Bundling	r	s_{min}	s_{max}
BmH 1a	Strategy 1	1.0	0.5	1.0
BmH 1b	Strategy 1	1.5	0.7	1.42

Name	Bundling	m	s_{min}	s_{max}
BmH 2a	Strategy 2	4	0.7	1.42
BmH 2b	Strategy 2	6	0.7	1.42

performance at retrieving the top-most similar images but vary greatly in their mAP. In the following experiments, we compare the results for the arguably best parameter settings, i.e. 1024 sketches for min-hash and 256 sketches with 4 partitions for PmH, to our approach and the bag-of-words baseline.

Bag-of-Words and Feature Bundles. We evaluate the performance of both of our bundling strategies 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 [9]. Figure 3 shows the obtained results on the FlickrLogos dataset for 10 different vocabularies. One can clearly see that the bag-of-words with tf-idf weighting has its peak at a vocabulary of 1 million words, which confirms that large vocabularies are beneficial for image search [9].

In order to find the best bundle configurations we have done an extensive parameter sweep on the parameters of the bundle configuration. Due to limited space, we cannot show a detailed evaluation for each of these parameters. Therefore we report the performance of 4 selected well-performing bundle configurations (two for each bundling strategy) shown in Table 2.

As can be seen clearly in our figures, the two different bundling strategies (denoted as BmH1 and BmH2) perform equally well. Similar to bag-of-words they profit from large vocabularies, but the peak is at $500K$ words. More importantly, the bundles are on par with bag-of-words, but have an order of magnitude lower response ratio as can be seen in Figure 3 on the right.

Note that we rank the result set with the same metric for all approaches, i.e. by the cosine similarity as determined 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

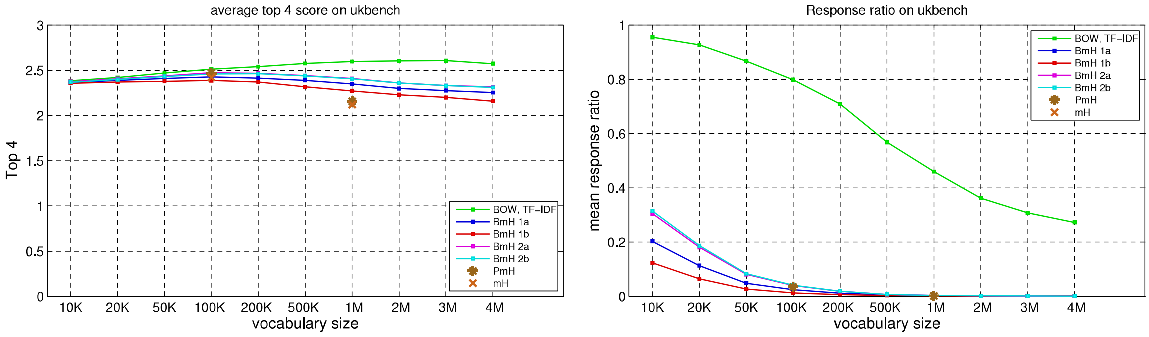

Fig. 4. Results on the UkBench dataset: The bundles yield an average top 4 score similar to bag-of-words as well as min-hashing and partition min-hash

the discriminative power of feature bundles for efficient filtering before more expensive post-retrieval steps are applied to the result set.

5.3 UkBench

We report the average top 4 score obtained on the UkBench dataset [8] (see Figure 4) to show the performance of feature bundles on a pure near-duplicate retrieval task. We do not optimize the bundle configurations specifically for this dataset. Instead, we show the performances for the bundle configurations as in Table 2, since we want to demonstrate how the bundle configurations obtained on the FlickrLogos dataset generalize on another dataset. From the results it can be seen that the retrieval precision of the bundling is similar or better than that of min-hashing and partition min-hashing and slightly lower than that of bag-of-words. Again, the response ratio is much lower and expresses the efficiency with which near-duplicates are retrieved.

5.4 Oxford Buildings

Finally, we also compare the performance of the feature bundles with bag-of-words retrieval, min-hash and partition min-hash on the Oxford buildings dataset [9]. This dataset contains 5063 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 and is one of the most well-known datasets for image retrieval.

Again, we use the previously obtained bundle configurations and just report the retrieval performance as obtained with the evaluation protocol of the Oxford dataset. Figure 5 shows the results. One can observe that bag-of-words performs best, while the bundles are worse yet outperform mH and PmH. Interestingly, the bundles outperform the bag-of-words retrieval if the database is increased by adding 100, 000 distractor images downloaded from Flickr. In that case one can observe a performance drop of both bag-of-words and feature bundles (see curve BOF, TF-IDF, 100k in Figure 5), but the bundles retain their extremely low response ratio. This demonstrates that bundling spatially related features suppresses false positives.

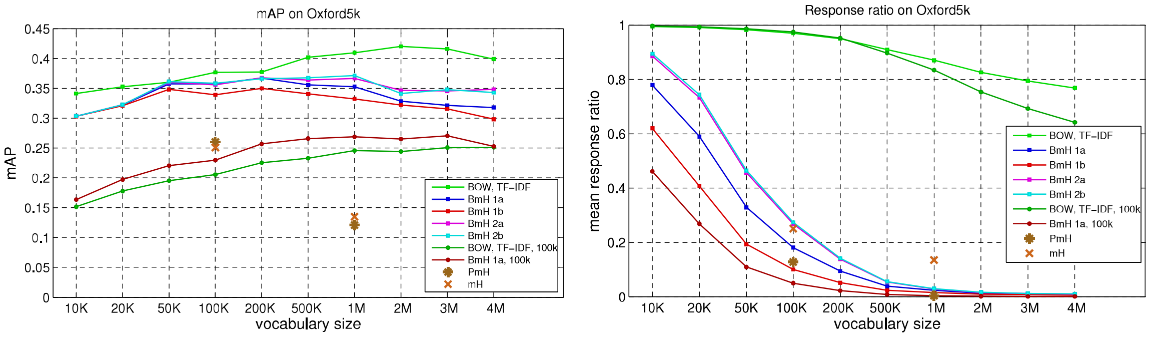

Fig. 5. Retrieval results on the Oxford dataset

6 Conclusion

In this work we introduced a robust technique for efficient indexing and search of feature bundles. Each bundle carries the information of individual visual words and their surrounding neighborhood. We showed that the bundles have a performance on par with bag-of-words models but with significant lower false positives, i.e. the result set is reduced by an order of magnitude. This makes much more complex and expensive post-retrieval operations on the small result set feasible.

References

1. Chum, O., Matas, J.: Large-scale discovery of spatially related images. PAMI, 371–377 (2010)
2. Chum, O., Perdoch, M., Matas, J.: Geometric min-Hashing: Finding a (thick) needle in a haystack. In: Proc. CVPR (2009)
3. Chum, O., Philbin, J., Isard, M.: Scalable near identical image and shot detection. In: Proc. CIVR (2007)
4. Chum, O., Philbin, J., Zisserman, A.: Near duplicate image detection: min-hash and tf-idf weighting. In: Proc. BMVC (2008)
5. Jégou, H., Douze, M., Schmid, C.: Improving Bag-of-Features for Large Scale Image Search. IJCV 87(3), 316–336 (2009)
6. Lee, D.C., Ke, Q., Isard, M.: Partition Min-Hash for Partial Duplicate Image Discovery. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 648–662. Springer, Heidelberg (2010)
7. Muja, M., Lowe, D.: Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. In: Proc. VISAPP (2009)
8. Nistér, D., Stewénius, H.: Scalable Recognition with a Vocabulary Tree. In: Proc. CVPR (2006)
9. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: Proc. CVPR (2007)
10. Romberg, S., Garcia Pueyo, L., Lienhart, R., van Zwol, R.: Scalable Logo Recognition in Real-World Images. In: Proc. ICMR (2011)