Hello Neighbor: Accurate Object Retrieval with k-Reciprocal Nearest Neighbors

Conference Paper *in* Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition · July 2011

DOI: 10.1109/CVPR.2011.5995373 · Source: IEEE Xplore

5 authors, including:

Danfeng Qin
ETH Zurich
5 PUBLICATIONS **99** CITATIONS
SEE PROFILE

Stephan Gammeter
ETH Zurich
8 PUBLICATIONS **226** CITATIONS
SEE PROFILE

Till Quack
ETH Zurich
14 PUBLICATIONS **613** CITATIONS
SEE PROFILE

Luc Van Gool
ETH Zurich
917 PUBLICATIONS **44,015** CITATIONS
SEE PROFILE

Hello neighbor: accurate object retrieval with k-reciprocal nearest neighbors

Danfeng Qin[1] Stephan Gammeter[1] Lukas Bossard[1] Till Quack[1,2] Luc van Gool[1,3]
[1]ETH Zürich [2]Kooaba AG [3]K.U. Leuven
{qind,gammeter,bossard,tquack,vangool}@vision.ee.ethz.ch

Abstract

This paper introduces a simple yet effective method to improve visual word based image retrieval. Our method is based on an analysis of the k-reciprocal nearest neighbor structure in the image space. At query time the information obtained from this process is used to treat different parts of the ranked retrieval list with different distance measures. This leads effectively to a re-ranking of retrieved images. As we will show, this has two benefits: first, using different similarity measures for different parts of the ranked list allows for compensation of the "curse of dimensionality". Second, it allows for dealing with the uneven distribution of images in the data space. Dealing with both challenges has very beneficial effect on retrieval accuracy. Furthermore, a major part of the process happens offline, so it does not affect speed at retrieval time. Finally, the method operates on the bag-of-words level only, thus it could be combined with any additional measures on e.g. *either descriptor level or feature geometry making room for further improvement. We evaluate our approach on common object retrieval benchmarks and demonstrate a significant improvement over standard bag-of-words retrieval.*

1. Introduction

We are interested in retrieving images showing a particular object from a large database of reference images. This is an important problem with applications in Web image retrieval, mobile visual search, or auto-annotation of photos. Typically, object types covered by the images in the database consist of landmark buildings, scenery or other 3D objects.

Most approaches to solve this problem are based on the "visual words" concept, which in essence borrows techniques from text retrieval, after quantizing localized visual features into visual vocabularies [16, 17, 20]. This method has turned out to be very powerful, since it allows for scalable retrieval in databases of millions of images at quite high precision. Even though astonishing progress has been made in terms of scalability and precision, accuracy on common retrieval benchmarks still shows room for significant improvements. And of course, any such accuracy improvement should ideally not affect memory consumption or retrieval time.

Thus, many recent works towards improving accuracy have focused on improving features, visual vocabularies or distance measures on a quite general level. Beyond these rather general measures, a further vantage point for improvement is given by exploiting the specific differences between text and visual retrieval. For instance, in images we can exploit the geometric arrangement of visual features (in 2D or 3D), whereas in text documents we have only access to the sequential arrangements of words in lines of text. Exploiting this geometric structure using RANSAC [9] or similar kinds of estimations is consequently a very common step taken in order to improve retrieval accuracy [17].

In this paper we try to exploit another characteristic specific to visual data in order to improve accuracy of object retrieval results: often, the reference database contains many images showing the same object covering it from varying viewpoints *etc*. We make use of this by constructing a graph on the image database connecting each image with likely related images. At query time this graph is used to construct a set of database images that are closely related to the query image, then based on this *close set* the rest of the database is re-ranked. As we will show this has two benefits: first, treating the two sets with different similarity measures allows for compensation for the "curse of dimensionality", *i.e*. the degradation of distance functions in high dimensional spaces. Second, it allows for dealing with the uneven distribution of images in the data space. Dealing with both challenges has very beneficial effect on retrieval accuracy.

The main contribution of this paper is a method that improves image retrieval purely on the bag-of-words level. It does so without relying on lower-level information like for instance the geometric arrangement of features or the geometry of the descriptor space. As such our method can be used in a wide variety of settings. We also achieve very competitive results at reasonable overhead in memory usage and very little additional computational complexity during

query time.

The remaining part of the paper is structured as follows: We first discuss related work in the immediately following section. Section 3 lies the basis for our method, by discussing some key characteristics of visual words based object retrieval. We introduce our method for more accurate object retrieval in Section 4. Experiments and analysis of the effects of optimization on retrieval tasks follow in Section 5. Section 6 concludes the paper.

2. Related work

Our work relates to recent contributions in the field of object retrieval with visual vocabularies in several aspects. The relevant works build on the common bag-of-features retrieval approach and have proposed improvements, which can be roughly grouped into three categories.

A first group of works deals with improvements on the feature level. In descriptor space the Euclidian distance is often used to assess the similarity of features. However, it has been shown this is not the optimal similarity measure in most situations. In the context of large scale image retrieval this problem has recently been addressed by several works. For instance in [15] a probabilistic relationship between visual words is proposed as an alternative distance measure. It is based on an "oversegmentation" of the descriptor space with an extremely large vocabulary, and probabilistic relations between the visual words. This way, for each feature mapped to a visual word, a statistic of alternative visual words is learned. The relations are learned offline from a large set of feature tracks. Slightly similar is the work [19], where data is used to learn a projection from SIFT feature space to a new Euclidean space, such that clustering is more likely to put matching descriptors into the same visual words.

A second group of works deals with the quantization artifacts introduced while assigning features to visual words. The most common effect of quantization artifacts is, that for two images showing the same object, corresponding features are not assigned to the same visual word. One way of dealing with this problem is by assigning each feature descriptor to multiple visual words as proposed in [18], however the more words are assigned to a feature, the more posting lists in the inverted index have to be traversed, thus increasing the query time. In [10], Jégou *et al*. addressed this problem by first constructing a relatively coarse vocabulary plus a binary signature for each feature. When a feature of the query images is assigned to a visual word of the coarse vocabulary, the binary signature is used to filter out database features by setting a threshold on the Hamming distance.

A third group of works deals with shortcomings on the document retrieval or database level. In [7], Chum *et al*. adopt query expansion (that originated in text retrieval) to the visual domain. Strict geometric verification is applied to the initial top list in order to extract a set of images that are very likely to be relevant to the query. Then a generative model is used to fuse the information provided by the additional images into a new query, which significantly increases recall.

A common cause of problems is due to the independence assumption between visual words, commonly used because of efficiency reasons. In reality this independence assumption is violated and some visual words co-occur more often than others. This can severely degrade retrieval accuracy. If for instance the query contains a set of frequently co-occurring visual words, then it is likely to match to unrelated images that contain the same set of co-occurring visual words. These sets are commonly referred to as *bursts* [11] or *co-ocsets* [6]. In [11], Jégou *et al*. evaluated several voting schemes that account for intra- and inter-image *bursts*. Chum *et al*. [6] addressed this problem by finding and removing sets of frequently co-occurring visual words. In both cases improvement in retrieval accuracy was demonstrated. Also operating on the document vector level, Jégou *et al*. [12, 13] improved the accuracy of visual word retrieval, by accounting for changes in the local distributions of the visual word vectors. To this extent, they introduced an iterative update scheme that modifies the distance function between vectors in a way that nearest neighborhood relationships become more symmetric.

Most similar to this paper are probably [7] and [13]. As we will explain in the following sections in more detail, the key differences to our work are that we do not rely on lower level information like for instance the geometric arrangement of features and we do not symmetrize nearest neighbor relationships (in contrast to [13]).

3. Motivation

In this section we motivate our approach for improving accuracy of object retrieval by two key observations. Before we discuss the observations we give a brief overview of object retrieval with visual words.

Overview of object retrieval with visual words. In visual word based retrieval images are represented as sparse high dimensional visual word vectors. Given a query vector, visual search is formulated as ranking the vectors in the database according to their distance or similarity to the query vector. These vectors are constructed by first extracting a set of local features (usually SIFT [14] or SURF [5] features) for a given image which are then quantized using a visual vocabulary. The visual vocabulary is commonly learned by clustering a random sample of feature descriptors, where the number of cluster centers K is usually somewhere around 10^6. The quantization indices correspond to the non zero elements of the *visual word vectors*.

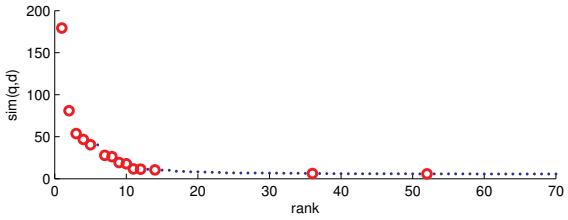


Figure 1: Degradation of similarity in document space for a typical query from the Oxford5k data set. The y-axis shows $\mathrm{sim(q, d)}$, the x-axis the rank of retrieved images. The red circles show true positives at their similarity and rank.

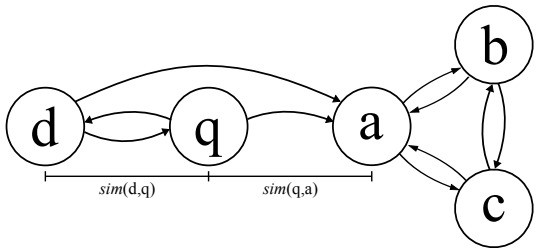


Figure 2: Difference between the unidirectional nearest neighbor set $top(2, q)$ and the 2-reciprocal nearest neighbor set $R(2, q)$.

In the bag-of-words model, each non zero element of the vector counts how many times the visual word appears in the image. Since the number of visual words in an image is usually many orders of magnitude below the size of the visual vocabulary, the visual word vectors are extremely sparse. Using an inverted index this sparsity is exploited to efficiently calculate the similarity of a query vector to all database vectors.

For all experiments in this paper we use the same similarity function as [10], which corresponds to the bag-of-words model with an additional inverse document frequency weighting term:

$$\mathrm{sim}\,(q, d) = \frac{\sum_{i=1}^{K} q_i\, d_i\, idf(i)^2}{\|q\|\,\|d\|} \quad (1)$$

$$\mathrm{idf}\,(i) = \log\left(\frac{\sum_{i=1}^{K}\sum_{d\in D} d_i}{\sum_{d\in D} d_i}\right) \quad (2)$$

where q and d are visual word vectors of length K and $D = \{d^1 \ldots d^N\}$ is the set of database vectors.

Observation 1: Similariy functions degrade quickly in high dimensional spaces. A fundamental issue for visual word based image retrieval is the high dimensionality of the visual word vector space. While this high dimensionality facilitates fast search, it also has the effect, that most distance or similarity measures quickly degenerate at points far away from the query vector.

An illustration of this phenomenon is shown in Figure 1 where we plot the similarity measure $\mathrm{sim(q, d)}$ (*c.f.* Equation 1) for the top 70 ranked images in the Oxford5k data set [2] for a given query. Correctly retrieved matches from the evaluation data set are denoted by a red circle. Most images with high similarity are of course true positives, however the similarity curve quickly flattens out giving relevant and non relevant images almost the same score at lower ranks. So the similarity measure is very useful for images close to the query, but it loses its utility far away from the query. One way of dealing with this problem is by modifying the similarity measure $\mathrm{sim(q, d)}$ in a way that more relevant images are pushed closer to the query and irrelevant images are pushed away from the query. For instance, Hamming embedding [10] or soft visual word assignment [18] do this by reducing quantization artifacts.

Descriptor space learning techniques [19, 15] push relevant images closer to the query by correcting for the fact that the Euclidean norm is not a perfect distance measure in SIFT or SURF descriptor spaces. In addition, filtering irrelevant images from the ranked lists is typically achieved by geometric verification [17] or similar methods (*e.g.* [10]).

In this paper we try to address these effects of the curse of dimensionality in a slightly different way. Accepting that the similarity measure can degrade quickly, we will split the database vectors at query time into two groups. One group which is close to the query, and for which the regular similarity measure $\mathrm{sim(q, d)}$ can still correctly separate relevant from non-relevant vectors, and a second group, for which the similarity measure can not distinguish between relevant from non-relevant vectors anymore. For the second group we use a different similarity measure.

Observation 2: Non-reciprocity of near neighbor relationships. Let us define the k-nearest neighbors (*i.e.* the top-k list) of a vector q as the k most similar vectors in the database D:

$$\mathrm{top}(k, q) \subset D \quad (3)$$

$$|\mathrm{top}(k, q)| = k \quad (4)$$

$$\mathrm{sim(q, a)} > \mathrm{sim(q, b)} \quad \forall \quad a \in \mathrm{top}(k, q),\ b \in D \setminus \mathrm{top}(k, q) \quad (5)$$

While the similarity measure $\mathrm{sim(q, d)} = \mathrm{sim(d, q)}$ itself is symmetric, nearest neighbor relationships are not. This means that $a \in \mathrm{top}(k, b)$ does not imply $b \in \mathrm{top}(k, a)$ in general.

We define the set of k-reciprocal nearest neighbors $\mathrm{R}(k, a)$ of a as

$$\mathrm{R}(k, a) = \{b \in \mathrm{top}(k, a) \wedge a \in \mathrm{top}(k, b)\} \quad (6)$$

which is of course trivially symmetric. The k-reciprocal nearest neighborhood relationship $b \in \mathrm{R}(k,a)$ is also a much stronger indicator of similarity than the unidirectional nearest neighborhood relationship $b \in \mathrm{top}(k,a)$, since it takes into account the local densities of vectors around a and b.

We illustrate in Figure 2 the difference between the unidirectional nearest neighbor set $top(2,q)$ and the 2-reciprocal nearest neighbor set $R(2,q)$. $top(2,q)$ contains the node a and d, $R(2,q)$ only contains node d, even though a and d are at the same distance from the query q. In such a situation it makes sense to assume that d is more relevant to the query q than a, since a has a high similarity to other nodes that share no connection to q.

We are of course not the first to make this observation. Contextual dissimilarity measures [12, 13] for instance are based on exactly this idea. However unlike [12, 13], we do not directly symmetrize nearest neighborhood relationships in this work. Instead we use k-reciprocal nearest neighbors as a tool to find images which are very likely to be related and to disambiguate database vectors that are far away from the query vector.

These two observations are the basis for our object retrieval method, which will be discussed in the following section.

4. Our Approach

At query time we want to separate the database into two disjoint sets, the *close set* which contains images highly relevant to the query and the *far set* which simply refers to the rest of the database. The final ranking list is the concatenation of the *close set* for which parts internally are ranked according to the original similarity measure $sim(q,d)$ (*c.f.* Equation 1) and the *far set* which is ranked according do a different similarity measure. We first discuss how the *close set* is constructed and then describe the similarity measure that is used for the *far set*.

4.1. Close set construction

In order to identify images highly related to the initial query image q, we start by adding the k-reciprocal nearest neighbors $R(k,q)$ of the query to the *close set*.

In Figure 3 we show for a query in the Oxford5k data set how precision and recall of $R(k,q)$ change for various values of k. With higher values of k, recall is increased and saturates while precision rarely decreases. Since in practice some images have very few k-reciprocal nearest neighbors, even for very large k, we grow the initial *close set* $N_{q,t=0}$ by iteratively adding neighboring nodes to increase recall. Nodes are only added if a set of conditions are met which are designed in a way, that only images that are very likely to be related to the query image are added.

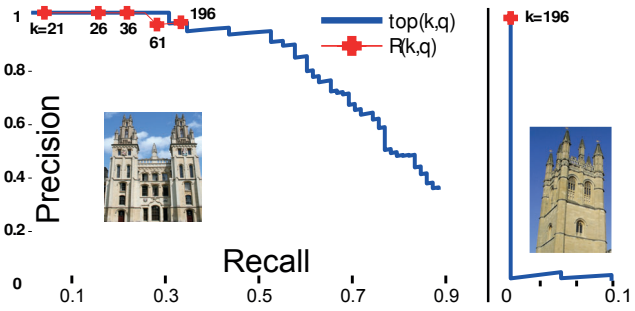


Figure 3: Precision and recall of $R(k,q)$ in comparison to the top(k,q).

We first define the forward rank f-rank(a,q) of a as the position that a has in the top list of q and the backward rank b-rank(a,q) is defined as the position that q occupies in the top-k list of a:

$$\text{f-rank}(a,q) = k \iff a \in \text{top}(k,q) \setminus \text{top}(k\text{-}1,q) \quad (7)$$

$$\text{b-rank}(a,q) = \text{f-rank}(q,a) \quad (8)$$

$$a \in \mathrm{R}(k,q) \iff \text{f-rank}(a,q) < k \wedge \text{b-rank}(a,q) < k \quad (9)$$

Since we are only interested in finding nodes close to the query, we only consider nodes $d \in D$ if their forward and backward rank relative to the query q do not exceed a certain threshold k_{max}:

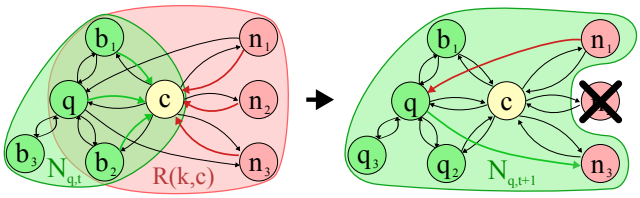
$$\text{f-rank}(q,d) < k_{max} \ \vee \ \text{b-rank}(q,d) < \frac{1}{2}\, k_{max} \quad (10)$$

Ignoring all nodes which do not satisfy these constraints we grow $N_{q,t=0}$ by the following procedure as described in Algorithm 1.

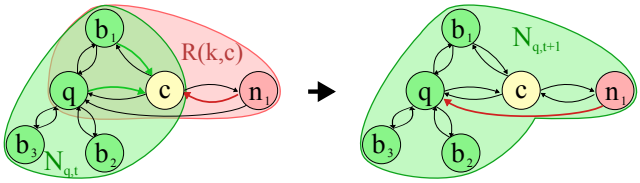
Algorithm 1: Expansion step

```
for t ← 0 to 2 do
    N_{q,t+1} ← N_{q,t};
    foreach n ∈ N_{q,t} do
        if |N_{q,t} ∩ R(k,n)| > ½ |N_{q,t}| then
            N_{q,t+1} ← R(k,n) ∪ N_{q,t+1};
        if |N_{q,t} ∩ R(k,n)| > |R(k,n) \ N_{q,t}| then
            N_{q,t+1} ← R(k,n) ∪ N_{q,t+1};
```

The first condition allows only nodes which are connected to at least half of the *close set* to bring in their neighbors. This high connectivity ensures that added nodes are very likely to be relevant to the query. The second condition relaxes this restriction slightly by allowing weakly connected nodes to bring in their neighbors if the amount of new neighbors is smaller than the amount of connections already made to the *close set*. Nodes added to $N_{q,t+1}$ are sorted according to $sim(q,d)$ and inserted in this order into



(a) Node C's neighborhood is considered, if it contains more than the half of the initial set, *i.e.* $|N_{q,t} \cap \mathrm{R}(k,c)| > \frac{1}{2}|N_{q,t}|$



(b) Node C's neighborhood is considered, if it adds less unknown nodes than known, *i.e.* $|N_{q,t} \cap \mathrm{R}(k,c)| > |\mathrm{R}(k,c) \setminus N_{q,t}|$

Figure 4: Overview over the expansion rules. For the new set $N_{q,t+1}$ only nodes are considered which either occur in the first half of the top-k list of the query image ($q \rightarrow n_3$), or if the query image occurs within the top list of the image ($n_1 \rightarrow q$).

the final *close set*. This procedure can be seen as a form of query expansion, however unlike [7] we do not rely on any geometric or other lower-level information. Figure 4 gives an overview of the two conditions for better visualization and in Figure 5 we give a real world example of the growing procedure.

In order to efficiently construct the *close set* for a new query, we pre-compute a directed graph $(d_1 \dots d_n \in D, (u,v)_i \in V)$ on top of the image database. In this graph every node represents an image and a connection $(u,v) \in V$ from node u to another node v is made if node v appears in the truncated top$(k_{\max}, \cdot)$ list of u:

$$(u,v) \in V \iff v \in \mathrm{top}(k_{\max}, u) \tag{11}$$

where we used $k_{\max} = 1000$ for all experiments in this paper. Using Equations 8 and 9, k-reciprocal neighborhoods $\mathrm{R}(k,q)$ can be efficiently determined.

In order to construct this graph, we query our retrieval system with every image in the database. While this step is quadratic in the number of images, we do not see this yet as a fundamental restriction since computation is trivially distributable. Also the query operation is quite fast, for a set of one million images we could compute the aforementioned graph in less than 5 hours using only 8 machines. It is reasonable to assume, that calculating the graph for 10 million images would still be feasible. For larger data sets, approximations may be used like for instance min hash [8] which has only linear complexity in the number of images. However since we can still deal quite comfortably with up to one million images we have not evaluated this for the purpose of our method. At query time the similarity of the query image to all database images is calculated and the graph is updated to include a node for the query image.

Figure 5: Example for the *close set* in the expansion step.

4.2. Far set re-ranking

Once the *close set* is constructed, it is used to re-rank the rest of the database. Since images outside of the *close set* are likely to have a low similarity to the query, the original similarity measure $sim(q,d)$ is not useful anymore. From the vantage point of the query, images outside of the *close set* all look equally dissimilar. However if we turn the tables, and look from the position of an element in the far set this might not be true.

Images in the *far set* which are closely surrounded by other images in the *far set* will populate their top$(k,\cdot)$ list with their close neighbors but not the ones from the *close set*. However images which are dissimilar to the entire database but still rather close to the initial query can populate their top$(k,\cdot)$ list with images from the *close set*.

Intuitively it also makes sense that images which do not have any close neighbors except for images in the *close set* are more likely to be relevant to the query, than images that have close neighbors which are not related to the *close set*. In order to make use of this contextual similarity we calculate for each document in the *far set* ($f \in D \setminus N_{q,2}$) the average rank that images in the *close set* would have if image f were used as a query:

$$\mathrm{sim}(f,q) = \mathrm{cutoff} - \frac{1}{|N_{q,2}|} \sum_{c \in N_{q,2}} \min\left(\text{b-rank}(f,c), \mathrm{cutoff}\right) \tag{12}$$

where we use a cutoff to account for the fact that only a truncated version of the ranking lists is present at retrieval time. We used cutoff $= 3000$ for all experiments in this paper.

5. Experiments

In this section we evaluate the performance of our method on five different datasets. First we give an overview

of the datasets. Then we asses the performance of the *close set* and the performance of the *far set* re-ranking separately. At the end a comparison of our full method to the baseline approach is given.

5.1. Evaluated datasets

We evaluated our method on the Oxford5k [17, 2], Oxford105k [17], Paris [18, 3], University of Kentucky [16, 4] and the INRIA Holidays [10, 1] dataset. Oxford5k and Paris are relatively small datasets containing $5\,063$ and $6\,412$ database images each. For both datasets 55 queries with ground truth are provided. In the Oxford dataset on average $10\ \%$ of the database is relevant to a query, whereas for the Paris dataset almost $30\ \%$ of images are relevant. Furthermore, there is a high variance in the number of ground truth images for the different queries. The Kentucky dataset consists of $10\,200$ images for which always 4 show the same object. The Holidays dataset consists one million distractor images and $1\,491$ relevant images of which 500 are queries. For a large portion of the queries in this dataset there are only 1 or 2 relevant images.

The Oxford105k dataset consists of Oxford5k and $100\,000$ distractor images. As we did not have access to the original distractor images we downloaded $100\,000$ random geotagged images from Flickr[1] and Panoramio[2] which have been taken at least 500 km away from Oxford in order not to incorporate possibly relevant images that could artificially pollute the results. Furthermore we ensured that all downloaded images have resolutions between 768×1024 and 1024×1024 pixels, as in the original Oxford5k dataset.

We used Hessian Affine SIFT descriptors and approximate k-means [17] to cluster a visual vocabulary with $500\,000$ centroids for Oxford5k, Paris and Kentucky each. For Oxford105k we used the same vocabulary as for Oxford5k. For the Holidays dataset we received the precalculated visual words for a $200k$ visual vocabulary from the authors of [10]. Thus our baseline and the one from [10] are exactly the same.

As performance measure, we used mean average precision (*mAP*) on the Oxford5k, Oxford105k, Paris and Holidays dataset while for the University of Kentucky dataset we use the top-4 score as defined by [16].

5.2. Close set accuracy

In the first part we demonstrate, that the construction of the *close set* which forms the first part of the final ranking list leads to higher accuracy than simply taking the top-k elements of the original ranking list. The size of the *close set* for a given query is dependent on the number of similar images in the database. Queries with many similar images in the database have a larger *close set* than queries with only few similar images in the database. Furthermore, the size of the *close set* depends on the threshold k. By varying k we produce *close sets* of different sizes. The *far set* for which in this experiment regular ranking (*c.f.* Equation 1) is used, is appended to the *close set* to form the final ranking list.

As can be seen by the blue lines in Figures 6,7,8,9,10 this gives a major improvement on all datasets for a wide range of k. The mAP and top-4 score give high importance to the first part of a ranking list, which is exactly where the *close sets* increases accuracy.

5.3. Far set accuracy

We investigate the effect of replacing the *close set* by simply a truncated top-k list for different values of k and re-rank the *far set* using this list according to Equation 12.

As can be seen by the green lines in Figures 6,7,8,9,10 this gives an improvement on all datasets for small k. However as k increases the performance asymptotically degrades back to the base line, since larger and larger portions of the beginning of the final ranking list are ranked using the same similarity measure as the baseline. This is especially visible for the Kentucky dataset. Since the top-4 score only considers the first 4 positions of the ranking list, for $k > 4$ the performance is equal to the baseline.

5.4. Full method

The full method combines the aforementioned rank list construction methods, such that the first entries of the final ranking list consist of the *close set* to which the *far set* is appended.

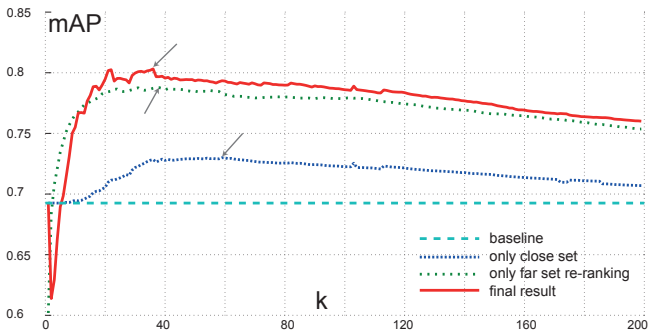


Figure 6: Mean average precision for Paris.

The red line in Figures 6,7,8,9,10 show the final result of the whole method for different thresholds k. The combination of *close set* construction and *far set* re-ranking leads to superior results over the baseline in all cases.

Figure 11 shows the average precision for the baseline versus the average precision of our improved method for individual query images for a fixed $k = \operatorname{argmax}_{\hat{k}} \operatorname{mAP}(\hat{k})$. Off-diagonal markers in the upper left triangle show a performance improvement, markers in the lower right triangle

[1] http://www.flickr.com
[2] http://www.panoramio.com

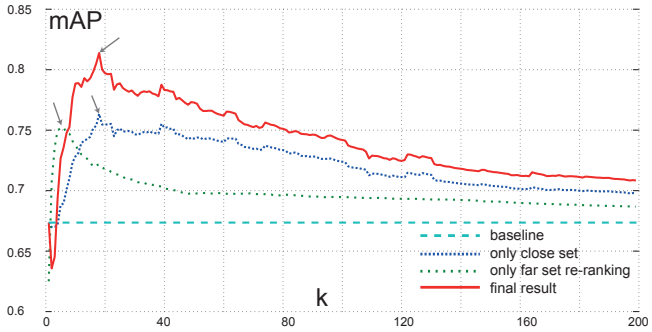


Figure 7: Mean average precision for Oxford5k.

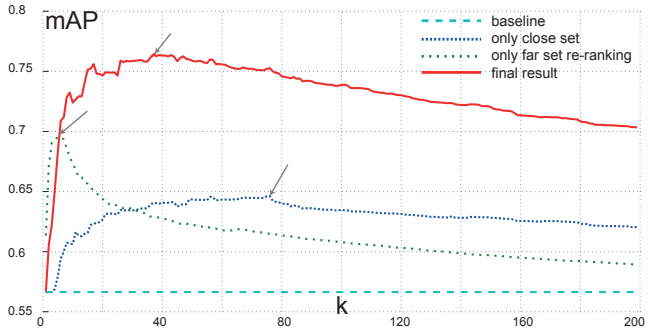


Figure 8: Mean average precision for Oxford105k.

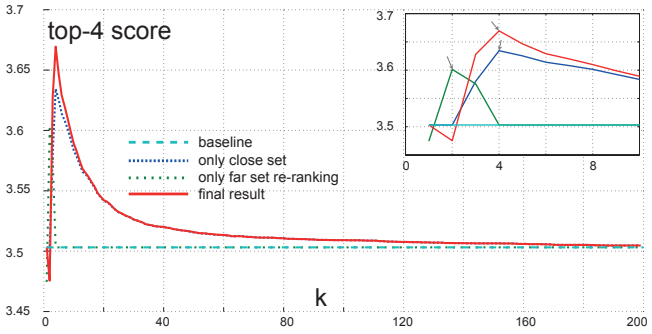


Figure 9: Top-4 score for Kentucky.

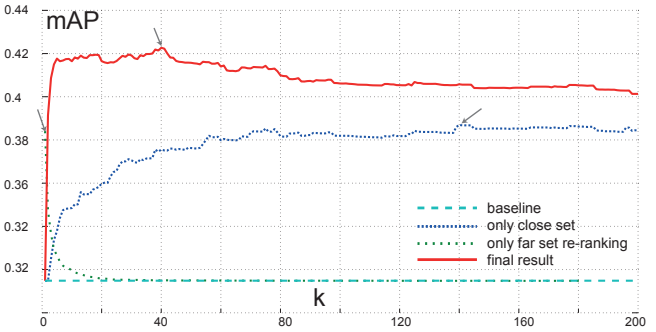
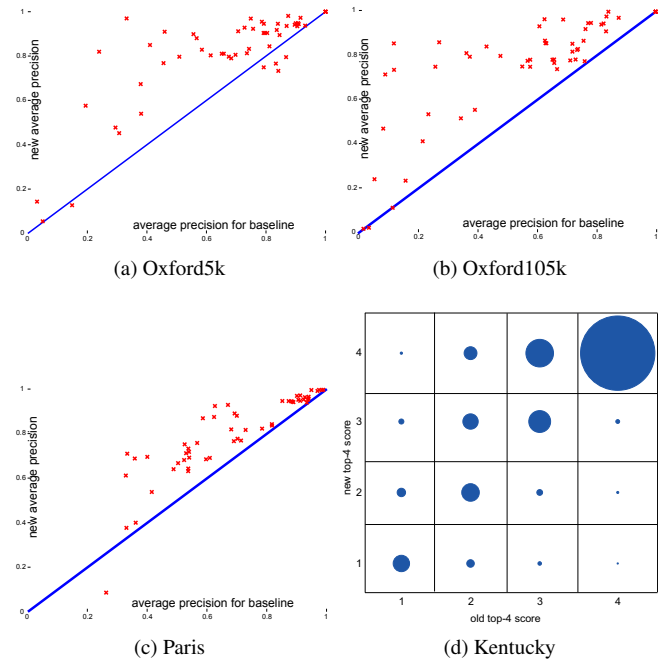


Figure 10: Mean average precision for INRIA Holidays.



(a) Oxford5k (b) Oxford105k

(c) Paris (d) Kentucky

Figure 11: Average precision (*AP*) of the baseline versus AP of our method

a degradation. For the University of Kentucky dataset a slightly different visualisation approach was taken. The top-4 score of the baseline method is plotted against the top-4 score of our new method. Each of the bubble's area corresponds to the number of images at this coordinate.

The combination of both methods yields in all datasets to superior results over the baseline. As the performance decays slowly, k is not as dataset specific as it might seem. Setting it to somewhere between 20 and 40 gives good results for real world datasets used in image retrieval applications. As can be seen in Table 1 for Oxford5k, Oxford105k and Paris we compete with the state of the art, however we do so without exploiting lower level information. For the Kentucky dataset we miss the state of the art only by 0.01 of top-4 precision. For the Holidays dataset it is well know that Hamming Embedding and Weak Geometric Consistency Constraint can greatly improve results, further more the $200k$ visual vocabulary is quite small for such a large dataset. We chose to evaluate our method on this challenging dataset to demonstrate that even under very unfavorable conditions we achieve a significant improvement.

Table 2 shows an overview over the total memory overhead per dataset and the average query time overhead for each query. As for each image in the database the forward- and the backward ranking lists need to be stored, the memory overhead grows linearly with the database size. This overhead is in the same order of magnitude as for instance

Dataset	Baseline	Our method	Jégou *et al.* [13]	Jégou *et al.* [11]	Mikulík *et al.* [15]	Chum *et al.* [7]	Chum *et al.* [6]	Philbin *et al.* [19]
Oxford5k [17, 2]	0.674	0.814		0.685	**0.849**			0.707
Oxford105k [17]	0.567	0.767			0.795	0.782	**0.864**	0.615
Paris [18, 3]	0.693	0.803			**0.824**			0.689
INRIA+1 Mio [10, 1]	0.315	0.423		**0.77**				
Kentucky [16, 4]	3.5	3.67	**3.68**	3.64				

Table 1: mAP for different datasets compared to results of state of the art results.

Dataset	Oxford5k	Oxford105k	Paris	INRIA	Kentucky
Memory [GiB]	0.16	2.35	0.13	22.35	0.23
Avg. time [ms]	5	6	8	30	4

Table 2: Additional memory overhead per dataset and average time overhead per query.

Hamming Embedding [10]. The query time overhead is mainly dependent on the length of the backward ranking list and the chosen threshold k as this restricts the size of the *close* and *far set*.

6. Conclusion

We have demonstrated that a significant improvement in bag-of-words retrieval can be achieved, without considering the geometric arrangement of features in an image nor by modifying the feature quantization step. Our method uses k-reciprocal nearest neighbors to identify an initial set of highly relevant images in the database which are then used to re-rank the remaining part of the database. On many data sets our approach competes with the state of the art. The memory overhead of our method is linear in the number of documents while the average query time overhead is neglectable.

As a secondary contribution, we make a binary executable and a C++ implementation of our our method available at our homepage[3]. Additionally we publish the precalculated visual words for the Oxford5k, Oxford105k, Paris and Kentucky dataset together with an evaluation package to reproduce our results at the same place.

Acknowledgements We are grateful to Matthijs Douze for providing the visual words of the Holidays dataset and for the financial support from the EC STREP project IURO and the SNF project K-Content.

References

[1] http://lear.inrialpes.fr/~jegou/data.php.

[2] http://www.robots.ox.ac.uk/~vgg/data/oxbuildings/index.html.

[3] http://www.robots.ox.ac.uk/~vgg/data/parisbuildings/index.html.

[4] http://www.vis.uky.edu/~stewe/ukbench/.

[5] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *ECCV'06*, 2006.

[6] O. Chum and J. Matas. Unsupervised discovery of co-occurrence in sparse high dimensional data. In *CVPR10*, 2010.

[7] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *CVPR07*, 2007.

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

[9] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In *Comm. of the ACM*, 1981.

[10] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV08*, 2008.

[11] H. Jégou, M. Douze, and C. Schmid. On the burstiness of visual elements. In *CVPR09*, 2009.

[12] H. Jégou, H. Harzallah, and C. Schmid. A contextual dissimilarity measure for accurate and efficient image search. In *CVPR'07*, 2007.

[13] H. Jégou, C. Schmid, H. Harzallah, and J. Verbeek. Accurate image search using the contextual dissimilarity measure. *Trans. PAMI*, 32(1):2–11, january 2010.

[14] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV99*, 1999.

[15] A. Mikulík, M. Perdoch, O. Chum, and J. Matas. Learning a fine vocabulary. In *ECCV'10*, 2010.

[16] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *CVPR'06*, 2006.

[17] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR'07*, 2007.

[18] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR08*, 2008.

[19] J. Philbin, M. Isard, J. Sivic, and A. Zisserman. Descriptor learning for efficient retrieval. In *ECCV10*, 2010.

[20] J. Sivic and A. Zisserman. Video google: a text retrieval approach to object matching in videos. In *ICCV'03*, 2003.

[3] http://www.vision.ee.ethz.ch/datasets/