Learning a Fine Vocabulary

Andrej Mikulík, Michal Perdoch, Ondřej Chum, and Jiří Matas

CMP, Dept. of Cybernetics, Faculty of EE, Czech Technical University in Prague

Abstract. We present a novel similarity measure for bag-of-words type large scale image retrieval. The similarity function is learned in an unsupervised manner, requires no extra space over the standard bag-of-words method and is more discriminative than both L2-based soft assignment and Hamming embedding.

Experimentally we show that the novel similarity function achieves mean average precision that is superior to any result published in the literature on the standard Oxford 105k dataset/protocol. At the same time, retrieval with the proposed similarity function is faster than the reference method.

1 Introduction

Recently, large collections of images become readily available [1–3] and image-based search in such collections has attracted significant attention of the computer community [4–8]. Most if not all recent state-of-the-art methods build on [4] who represented the image by a histogram of "visual words", *i.e.* discretized SIFT descriptors [9]. The bag-of-words representation possesses many desirable properties required in large scale retrieval. If represented as an inverted file, it is compact and supports fast search. It is

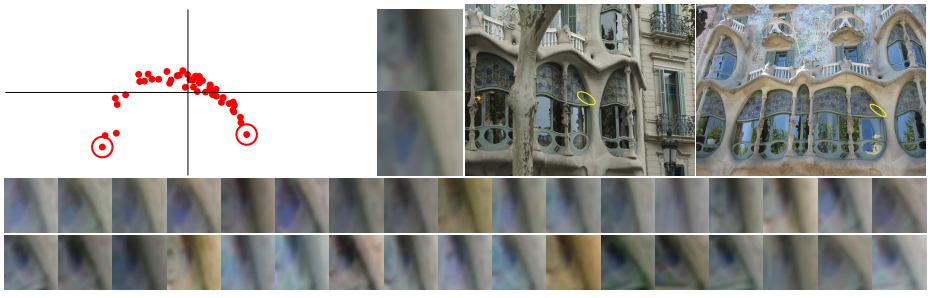


Fig. 1. An example of corresponding patches. A 2D PCA projection of the SIFT descriptors (left); two most distant patches in the SIFT space and the images where they were detected (right); a set of sample patches (bottom). The average SIFT distance within the cluster is 278, the maximal distance is 591.

sufficiently discriminative and yet robust to acquisition "nuisance parameters" like illumination and viewpoint change as well as occlusion[1].

The discretization of the SIFT features is necessary in large scale problems as it is neither possible to compute distances on descriptors efficiently nor feasible to store all the descriptors. Instead, only (the identifier of) the vector quantized prototype for visual word is kept. After quantization, Euclidean distance in a high (128) dimensional space is approximated by a 0–∞ metric - features represented by the same visual word are deemed identical, else they are treated as "totally different". The computational convenience of such a crude approximation of the SIFT distance has a detrimental impact on discriminative power of the representation. Recent methods like soft assignment and in particular the Hamming embedding aim at obtaining a better space-speed-accuracy trade off.

In this paper, unsupervised learning on a large set of images is exploited to improve on the 0–∞ metric. First, an efficient clustering process with spatial verification establishes correspondences within a huge (>5M) image collection. Next, a fine-grained vocabulary is obtained by hierarchical approximate nearest neighbour. The automatically established correspondences are then used to define a similarity measure on the basis of a probabilistic relationships of visual words; we call it the *PR visual word similarity*.

When combined with a 16 million word vocabulary (one or two orders of magnitude larger than commonly used), the PR similarity has the following desirable properties:

(i) it is more accurate, *i.e.* it is more discriminative, than both standard 0–∞ metric and Hamming embedding.
(ii) the memory footprint of the image representation for PR similarity calculation is roughly identical to the standard method and smaller than that of Hamming embedding.
(iii) search with PR similarity is faster than standard bag-of-words.

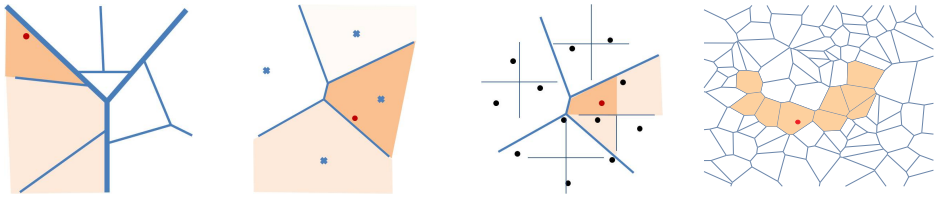
As a main contribution of the paper, we present a novel similarity measure that is learned in an unsupervised manner, requires no extra space (only O(1)) in comparison with the bag-of-words and is more discriminative than both 0–∞ and L2-based soft assignment.

As a secondary contribution, we will make available the database of matching SIFT features, together with the executable of the feature detector (hessian affine) and descriptor used to extract and describe the features.

2 Related Work

In this section, approaches to vocabulary construction and soft assignment suitable for large-scale image search are reviewed and compared.

[1] We only consider and compare with methods that support queries that cover only a (small) part of the test image. Global methods like GIST [10] achieve a much smaller memory footprint at the cost of allowing whole image queries only.



(a) (b) (c) (d)

Fig. 2. Different approaches to the soft assignment (saturation encodes the relevance): (a) hierarchical scoring [5] – the soft assignment is given by the hierarchical structure; (b) soft clustering [11] assigns features to r nearest cluster centers; (c) hamming embedding [12] – each cell is divided into orthants by a number of hyperplanes, the distance of the orthants is measured by the number of separating hyperplanes; (d) the set of alternative words in the proposed PR similarity measure.

In [4], the first 'bag of words' approach to image retrieval was introduced. The vocabulary (the number of visual words $\approx 10^4$) was constructed using a standard k-means algorithm. Adopting methodology from text retrieval applications, the image score is efficiently computed by traversing inverted files related to visual words present in the query. The inverted file related to a visual word W is a list of image ids that contain the visual word W. It follows that the time required for scoring the documents is proportional to the number different visual words in a query and the average length of an inverted file.

Hierarchical clustering. The hierarchical k-means and scoring of Nistér and Stewenius [5] is the first image retrieval approach that scales up. The vocabulary has a hierarchical structure which allows efficient construction of large and discriminative vocabularies. The quantization effect are alleviated by the so called hierarchical scoring. In such a type of scoring, the scoring visual words are not only stored in the leafs of the vocabulary tree. The non-leaf nodes can be thought of as virtual or generic visual words. These virtual words naturally score with lower *idf* weights as more features are assigned to them (all features in their sub-tree).

The advantage of the hierarchical scoring approach is that the soft assignment is given by the structure of the tree and no additional information needs to be stored for each feature. On the downside, experiments in [11] show that the quantization artefacts of the hierarchical k-means are not fully removed by hierarchical scoring, the problems are only shifted up a few levels in the hierarchy. An illustrative example of the soft assignment performed by the hierarchical clustering is shown in Fig. 2(a).

Lost in quantization. In [11], an approximate soft assignment is exploited. Each feature is assigned to $n = 3$ (approximately) nearest visual words. Each assignment is weighted by $e^{-\frac{d^2}{2\sigma^2}}$ where d is the distance of the feature descriptor to the cluster center.

The soft assignment is performed on features in the database as well as the query features. This results in n times higher memory requirements and n^2 times longer running time – the average length of the inverted file is n times longer and there are up to

n times more visual words associated with the query features. For an illustration of the soft assignment, see Fig. 2(b).

Hamming embedding. Jégou et al. [12] have proposed to combine k-means quantization and binary vector signatures. First, the feature space is divided into relatively small number of Voronoi cells (20K) using k-means. Each cell is then divided by n independent hyper-planes into 2^n subcells. Each subcell is described by a binary vector of length n. Results reported in [12] suggest that the hamming embedding provides good quantization. The good results are traded off with higher running time requirements and high memory requirements.

The higher running time requirements are caused by the use of coarse quantization in the first step. The average length of an inverted file for vocabulary of 20K words is approximately 50 times longer than the one of 1M words. Recall that the time required to traverse the inverted files is given by the length of the inverted file. Hence 50 times smaller vocabulary results in 50 times longer scoring time on average. Even if two query features are assigned to the same visual word, the relevant inverted file has to be processed for each of the features separately as they will have different binary signature.

While the reported bits per feature required in the search index ranges from 11 bits [8] to 18 bits [11], hamming embedding adds another 64 bits. The additional information reduces the number of features that can be stored in the memory by a factor of 6.8.

Summary All approaches to soft clustering mentioned above are based on the distance (or its approximation) in the descriptor (SIFT) space. It has been observed that the Euclidian distance is not the best performing measure. Learning a global Mahalanobis distance [13, 14] showed that the matching is improved and / or the dimensionality of the descriptor is reduced. However, even in the original work on SIFT descriptor matching [9] it is shown that the similarity of the descriptors is not only dependent on the distance of the descriptors, but also on the location of the features in the feature space. Therefore, learning a global Mahalanobis metric is suboptimal and a local similarity measure is required. For examples of corresponding pathes where SIFT distance does not predict well the similarity see Figures 1, 3, and 4.

3 The Probabilistic Relation Similarity Measure

Consider a feature in the query image with descriptor $D \in \mathcal{D} \subset R^d$. For most accurate matching, the query feature should be compared to all features in the database. The contribution of the query feature to the matching score should be proportional to the probability of matching the database feature. It is far too slow, *i.e.* practically not feasible, to directly match a query feature to all features in a (large) database. Also, the contribution of features with low probability of matching is negligible.

The success of fast retrieval approaches is based on efficient separation of (potentially) matching features from those that are highly unlikely to match. The elimination is based on a simple idea – the descriptors of matching patches will be close in some appropriate metric (L2 is often used). With appropriate data structure, enumeration of descriptors in proximity is possible in time sub-linear in the size of the database. All bag-of-words based methods use partitioning $\{W_i\}$ of the descriptor space

: $\cup W_i = \mathcal{D}, \quad W_i \cap W_{j \neq i} = \emptyset$. The partitions are then used to separate features that are close (potentially matching) from those that are far (non-matching).

In the case of hard assignment, features are associated with the quantized visual word defined by the closest cluster center. In the scoring that evaluates query and database image match, only features with the same visual word as the query feature are considered.

We argue that the descriptor distance is a good indicator of patch similarity only up to a limited distance, where the variation in the descriptors is caused mostly the imaging noise. In our approach, we abandon the assumption that the descriptor distance provides a good similarity measure of patches observed under different viewing angles or under different illumination conditions. Instead, we propose to exploit the matching probability between a feature observed in the query image and a database feature. Since our aim is to address retrieval in web-scale databases where store requirements are a critical, we constrained our attention to solution that store no extra information per feature, or more exactly, that have a minimum overhead in comparison with the standard inverted file representation.

The proposed approach. We propose to use a fine partitioning of the descriptor space, so that the partitions only compensate for the imagining noise (or even less). Even though the fine partitioning is learned in a data dependent fashion (as in the other approaches), the fine partitioning unavoidable separates matching features into a number of clusters.

For each partition (visual word) we learn which other partitions (called *alternative visual words*) that are likely to contain descriptors of matching features. This step is based on the probability of observing visual word W_j in a matching database image when visual word W_q was observed in the query image

$$P(W_j|W_q). \tag{1}$$

The probability (eqn. 1) is estimated from a large number of matching patches.

A simple generative model, independent for each feature, is adopted. In the model, image features are assumed to be (locally affine) projections of a (locally close to planar) 3D surface patches Z_i. Hence, matching features among different images are those that have the same pre-image Z_i. To estimate the probability $P(W_j|W_q)$ we start with (a large number of) sets of matching features, each set being different projections of a patch Z_i. Using the fine vocabulary (partitioning) the sets of matching features are converted to sets of matching visual words. We estimate the probability $P(w_j|w_q)$ from the feature tracks as

$$P(W_j|W_q) \approx \sum_{Z_i} P(Z_i|W_q)P(W_j|Z_i). \tag{2}$$

For each visual word W_q, a fixed number of alternative visual words that have the highest conditional probability (eqn. 2) is recorded.

3.1 Learning stage

The first step of our approach is to obtain a large number of matching image patches. The links between matching patches are consequently used to infer links between quan-

tized descriptors of those patches, *i.e.* between visual words. As a first step towards unsupervised collection of matching image patches, called (feature) tracks, clusters of matching images are discovered. Within each cluster, feature tracks are found by a wide-baseline matching method. This approach is similar to [15], where the feature tracks are used to produce 3D reconstruction. In our case, it is important to find a larger variety of patch appearances than precise point locations. Therefore, we adopt a slightly different approach to the choice of image pairs investigated.

Image clusters. We start with by analyzing connected components of the image matching graph (graph with images as vertices, edges connect images that can be matched) produced by a large-scale clustering method [16, 17]. Any matching technique is suitable provided it can find clusters of matching images in a very large database. In our case, an image retrieval system was used to produce the clusters of spatially related images. The following structure of image clusters is created. Each cluster of spatially related images is represented as an oriented tree structure (the skeleton of the cluster). The children of each parental node were obtained as results of an image retrieval using the parent image as a query image. Together with the tree structure, an affine transformation (approximately) mapping child image to its parent are recorded. These mappings are later used to guide (speed-up) the matching.

Feature tracks. To avoid any kind of bias (by quantization errors, for example), instead of using vector quantized form of the descriptors, the conventional image matching (based on the full SIFT [9]) has to be used. In principle, one can go back even to the pixel level [18, 19], however such an approach seems to be impractical for large volumes of data.

It is not feasible to match all pairs of images in the image clusters, especially not of clusters with large number of images (say more than 1000). It is also not possible to simply follow the tree structure of image clusters because not all features are detected in all images (in fact, only a relatively small portion of features is actually repeated). The following procedure, that is linear in the number of images in the cluster, is adopted for detection of feature tracks that would exhibit as large variety of patch appearances as possible. For each parental node, a sub-tree of height two is selected. On images in the sub-tree, a $2k$-connected graph called circulant graph [20] is constructed. Algorithm for construction of minimal $2k$-connected graph is summarized in Algorithm 1. Images connected by an edge in such a graph are then matched using standard wide-baseline matching. Since each image in the image cluster participates in at most 3 sub-trees (as father, son and grand-son), the number of edges is limited to $6kN$, where N is the size of the cluster. Instead of using epipolar geometry as a global model, a number of close-to-planar (geometrically consistent) structures is estimated (using affine homography). Unlike the epipolar constraint, such a one-to-one mapping enables to verify the shape of the feature patch. Connected components of matching and geometrically consistent features are called feature tracks.

Tracks that contain two different features from a single image are called inconsistent [15]. These features clearly cannot have a single pre-image under perspective projection and hence cannot be used in the process of 3D reconstruction. Such inconsistent tracks are often caused by repeated patterns. Inconsistent feature tracks are (unlike in [15]) kept as they provide further examples of patch appearance.

Input: K - requested connectivity, N - number of vertices
Output: V a set of vertices, $E \subset V \times V$ a set of edges of $2K$ connected graph (V,E).

1. **if** $2K \geq N - 1$ **then**
   return fully connected graph with N vertices.
   end
2. $S :=$ a random subset of $\{2, \ldots, \lfloor \frac{N-1}{2} \rfloor\}$, $|S| = K - 1$
3. $V := \{v_0, \ldots, v_{N-1}\}$
4. $E := \{(v_i, v_j) \,|\, v_i, v_j \in V, j = (i+1) \bmod N\}$
5. **for** $s \in S$
6. $\quad E := E \cup \{(v_i, v_j) \,|\, v_i, v_j \in V, j = (i+s) \bmod N\}$
7. **end**

Algorithm 1: Construction of the 2K connected graph with a minimal number of edges as a union of circulants.

Large vocabulary generation. To efficiently generate a large visual vocabulary we employ a hybrid approach - approximate hierarchical k-means. A hierarchy tree of two levels is constructed, each level has 4K nodes. In the assignment stage of k-means, approximate nearest neighbour, FLANN [21], is used for efficiency reasons.

First, a level one approximate k-means is applied to a random sub-sample of 5 million SIFT descriptors. Then, a two pass procedure on 10,713 million SIFTs (from almost 6 million images) is performed. In the first pass, each SIFT descriptor is assigned to the level one vocabulary. For each level one visual word a list of descriptors assigned to it is recorded. In the second pass, approximate k-means on each list of the descriptors is applied. The whole procedure takes about one day on a cluster of 20 computers.

Balancing the tree structure. For the average speed of the retrieval, it is important that the vocabulary is balanced, *i.e.* there are approximately the same number of instances of each visual word in the database.

There are two options how to balance the proposed structure. The level one structure can be balanced so that the branches are of approximately equal weight by constraining the length of the mean vectors (this stems from the fact that SIFT features live approximately on a hyper-sphere). Balancing can be also achieved by un-even splitting of the level two – proportional to the weight of the branch. In our implementation we have used the former.

The imbalance measure [12] for our vocabulary is 1.17 for the training image set (¿5M images) and 1.33 for the Oxford 105k (compared to 1.21 in [12]).

Computing the conditional probability. To compute the conditional probability (eqn. 2) from the feature tracks, an inverted file structure is used. The tracks are represented as forward files (named Z_i), *i.e.* lists of matching SIFT descriptors. The descriptors are assigned their visual word from the large vocabulary. Then, for each visual word w_k, a list of patches Z_i so that $P(Z_i|w_k) > 0$ (the inverted file) is constructed. The sum (eqn. 2) is evaluated by traversing the relevant inverted file.

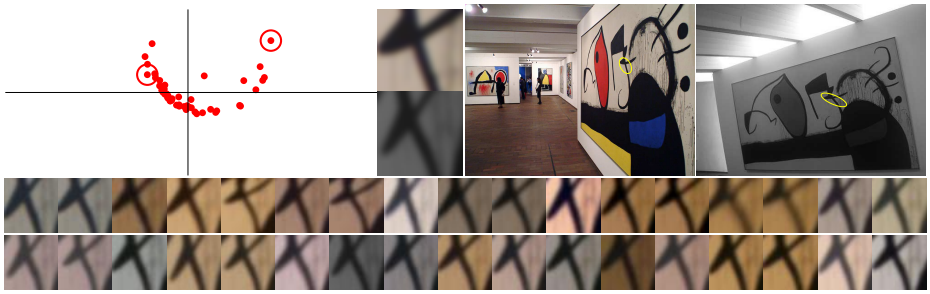


Fig. 3. A 2D PCA projection of a feature track of SIFT descriptors (left); the most distant patches and their images (right); sample of feature patches from the track. The distance of the most distant SIFT descriptors is 542 and is caused by an enormous change in the viewpoint.

Fig. 4. A 2D PCA projection of a feature track of SIFT descriptors (left); the most distant patches and their images (right); sample of feature patches from the track. The distance of the most distant SIFT descriptors is 593 and is caused by the viewpoint and scale change.

Statistics. Over 5 million images were clustered into almost 20 thousand clusters of 750 thousand images. Out of those 733 thousand were successfully matched in the wide-baseline matching stage. Over 111 million of feature tracks were established, out of which 12.3 millions are composed of more than 5 features. In total, 564 million features participated in the tracks, 319.5 million features belong to tracks of more than 5 features. Some examples of feature tracks are shown in Figures 5 and 6.

Memory and time efficiency. For the alternative words storage, only constant space is required, equal to the size of the vocabulary times the number of alternative words. The pre-processing consists of image clustering ([16] reports near linear time in the size of the database), intra-cluster matching (linearity enforced by the $2k$-connected circulant matching graph), and of the evaluation of expression eqn. (2) for all visual words. The worst case complexity of the last step is equal to the number of tracks (correspondences) times size of the vocabulary squared. In practice, due to the sparsity of the representation, the process took less than an hour in our settings for over 5 million images.

3.2 Retrieval stage

The implementation of the retrieval stage is fairly standard, using inverted files [4] for candidate image selection which is followed by fast spatial verification and query expansion [6]. The modifications listed below are the major differences implemented in our retrieval stage.

Unique matching. Despite being assigned to more than one visual word, each query feature is a projection of a single physical patch. Thus it can match only at most one feature in each image in the database. We find that applying this uniqueness constraint adds negligible computational cost and improves the results by approximately 1%.

Weights of alternative words. Contribution of each visual word is weighted by the *idf* weight [22]. A number of re-weighting schemes for alternative words have been tried, none of them affecting significantly the results of the retrieval.

4 Experiments

We have evaluated the performance of the PR similarity on a standard retrieval dataset Oxford 105K[2]. The experiments focus on retrieval accuracy and the retrieval speed. Since both our training set of 6 million images and the Oxford dataset were downloaded from FLICKR, we have explicitly removed all images from the training set that appear (or their scaled duplicate) in the test dataset.

4.1 Retrieval quality

We follow the protocol of 55 queries (11 landmarks, 5 queries each) defined in [23] and use the mean average precision as a measure of retrieval performance. We start by studying the properties of the PR similarity for a visual vocabulary of 16 million words.

In the first experiment, the quality of the retrieval as a function of the number of alternative words was measured, see Figure 7. The plots show that performance improves monotonically for plain retrieval without query expansion and almost monotonically when it is used for post-processing.

The second experiment studies the effects of the vocabulary size, the number of alternative words and compares the PR similarity with soft assignment. The left-hand part of Table 1 shows results obtained with the 16M vocabulary with three different settings ‘std’ – standard tf-idf retrieval with hard assignment of visual words; ‘5L’ and ‘16L’ – retrieval using alternative words (4 and 15 respectively). The righthand part presents results of reference state-of-the-art results [8] obtain with a vocabulary of 1M visual words learned on the PARIS dataset[3]. Two version of the reference algorithm are tested, without (“std”) and with the query soft assignment to 3 nearest neighbours (“SA 3NN”).

The experiments supports the following observations:

(i) For a hard assignment to a single visual word, 1M dictionary outperforms the 16M one. For the 0–∞ metric, the 16M visual word dictionary is too fine.

[2] http://www.robots.ox.ac.uk/∼vgg/data/oxbuildings/

[3] http://www.robots.ox.ac.uk/∼vgg/data/parisbuildings/

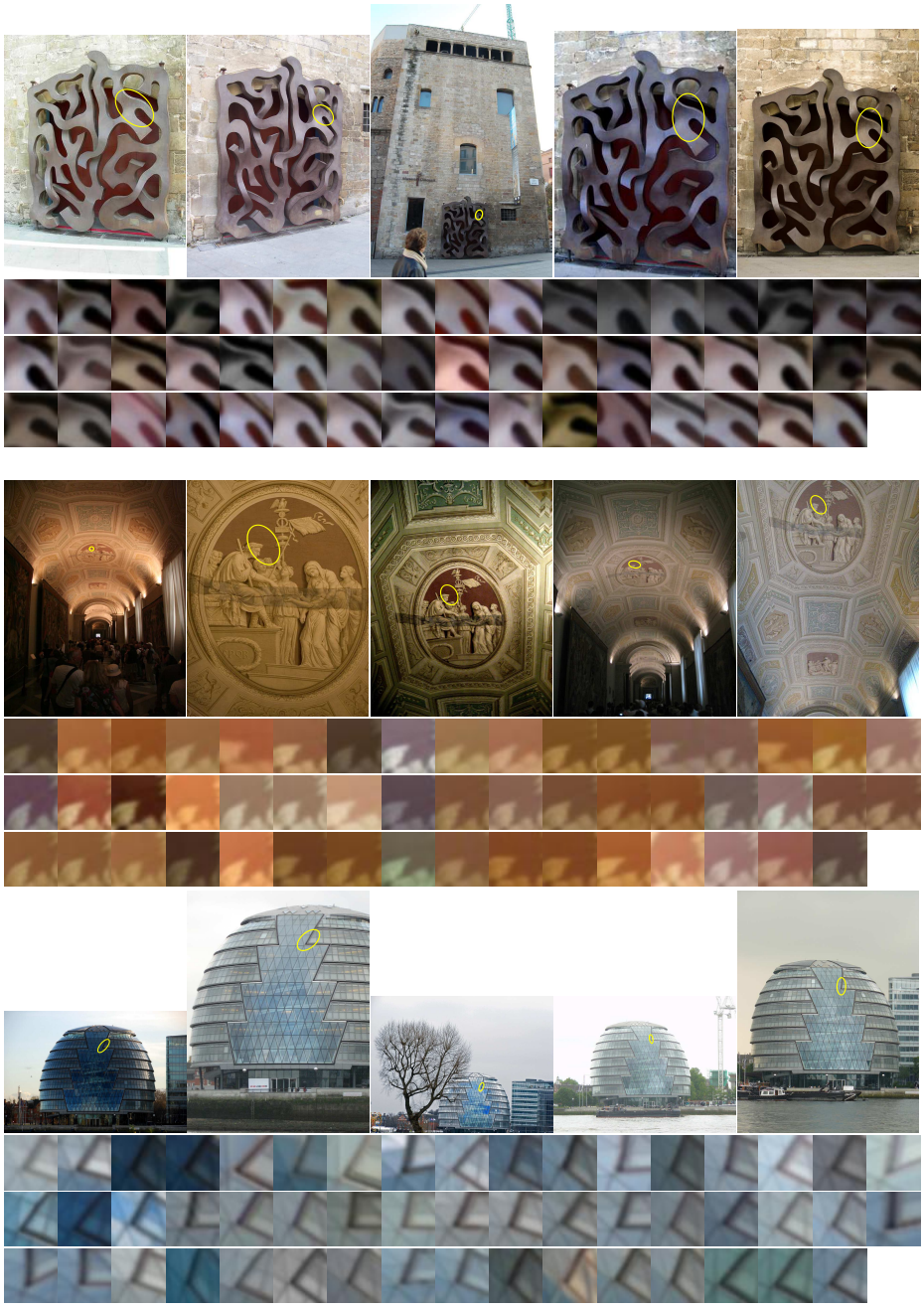


Fig. 5. Three examples of feature tracks of size 50. Five selected images and all 50 patches of the track. Even though the patches are similar, the SIFT distance of some pairs is over 500.

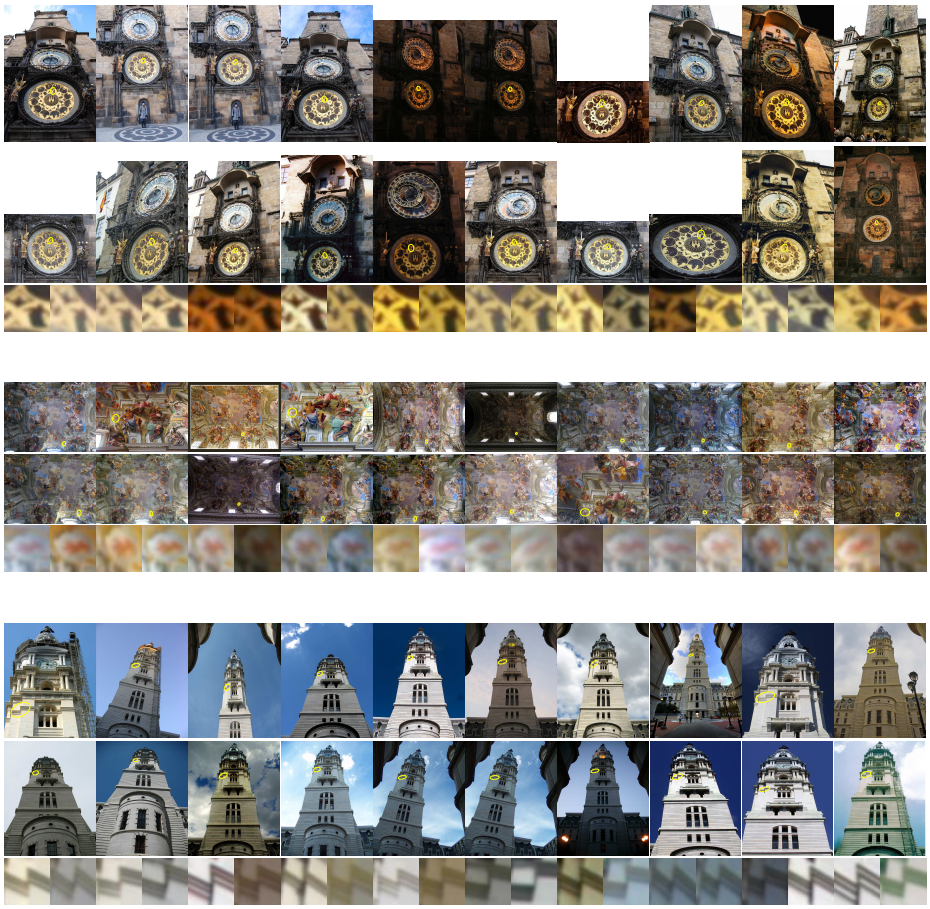


Fig. 6. Three examples of feature tracks of size 20. Images and corresponding patches, note the variation in appearance.

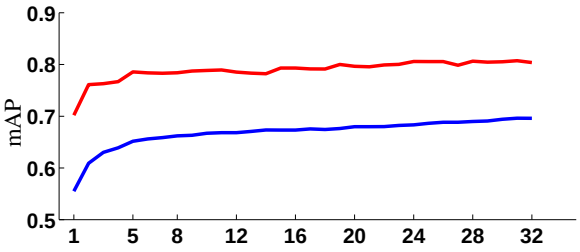


Fig. 7. The quality of the retrieval, expressed as mean average precision (mAP), increases with the number of alternative words. The mAP after (upper curve) and before (lower curve) query expansion is shown.

	16M std	16M L5	16M L16	PARIS 1M std	PARIS 1M SA 3NN
plain	0.554	0.650	**0.674**	0.574	0.652
QE	0.695	0.786	**0.795**	0.728	0.772

Table 1. Mean average precision for selected vocabularies on the Oxford 105k data-set.

	16M std	16M L5	16M L16	PARIS 1M std
Oxford 105K	0.071	0.114	0.195	0.247

Table 2. Average execution time per query in sec.

(ii) Similarity calculation with using the learned alternative words increases significantly the accuracy of the retrieval, both with and without query expansion.
(iii) The PR similarity outperforms soft SA in term of precisions, yet does not share the drawbacks of SA.
(iv) The PR similarity outperforms the Hamming embedding approach combined with query expansion, Jegou et al. [24, 12] report the mAP of 0.692 on this dataset.
(v) The mAP result for 16M L16 is superior to any result published in the literature on the Oxford 105k dataset.

4.2 Query times

To compare the speed of the retrieval, an average query time over the 55 queries defined on the Oxford 105K data set was measured. Running times recorded for the same methods and parameter settings as above are shown in Table 2.

The plot showing dependency of the query time on the number of alternative words is depicted in Figure 8. The times for the references PARIS 1M std method and the 16M L16 are of the same order. This is expected since the average length of inverted files is of the same order for both methods. The proposed method is about 20% faster, but this might be just an implementation artefact.

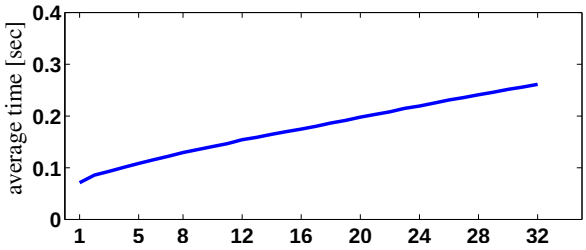


Fig. 8. Dependence of the query time on the number of alternative words.

Finally, we looked at the dependence of the speed of the proposed method as a function of the number alternative words. The relationship shown in Fig. 8 is very close

to linear plus a fixed overhead. The plot demonstrates that speed-accuracy trade-off is controllable via the number of alternative words.

4.3 Results on other datasets

The proposed approach has been tested on a number of standard datasets. These include Oxford, INRIA holidays (with manually corrected orientation of images, where the correct (sky-is-up) orientation is obvious), and Paris datasets. In all cases (Table 3), the use of the alternative visual words improves the results. On all datasets except the INRIA holidays the method achieves the state of the art results.

Dataset	16M std	16M L16	16M QE	16M L16 QE
Oxford 5k	0.618	0.742	0.740	0.849
Paris	0.625	0.749	0.736	0.824
Paris + Oxford 100k	0.533	0.675	0.659	0.773
INRIA holidays rot	0.742	0.749	0.755	0.758

Table 3. Results of the proposed method on a number of publicly available datasets.

5 Conclusions

We presented a novel similarity measure for bag-of-words type large scale image retrieval. The similarity function is learned in an unsupervised manner using geometrically verified correspondences obtained with an efficient clustering method on a large image collection.

The similarity measure requires no extra space in comparison with the standard bag-of-words method. Experimentally we show that the novel similarity function achieves mean average precision that is superior to any result published in the literature on the standard Oxford 105k dataset/protocol. At the same time, retrieval with the proposed similarity function is faster than the reference method.

As a secondary contribution we will make available the database of matching SIFT features, together with the executable of the feature detector (hessian affine) and descriptor used to extract and describe the features.

Acknowledgement. The authors are grateful for the support from EC project ICT-215078 DIPLECS, Czech Government under the research program MSM6840770038, GAČR project 102/09/P423, and Google.

References

1. http://books.google.com/help/maps/streetview/ (www)
2. http://www.panoramio.com/ (www)
3. http://www.flickr.com/ (www)

4. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: Proc. of ICCV. (2003) 1470 – 1477
5. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: Proc. CVPR. (2006)
6. Chum, O., Philbin, J., Sivic, J., Isard, M., Zisserman, A.: Total recall: Automatic query expansion with a generative feature model for object retrieval. In: Proc. ICCV. (2007)
7. Jegou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: Proc. ECCV. (2008)
8. Perdoch, M., Chum, O., Matas, J.: Efficient representation of local geometry for large scale object retrieval. In: CVPR. (2009)
9. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60** (2004) 91–110
10. Oliva, A., Torralba, A.: Building the gist of a scene: The role of global image features in recognition. Visual Perception, Progress in Brain Research **155** (2006)
11. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. In: Proc. CVPR. (2008)
12. Jégou, H., Douze, M., Schmid, C.: Improving bag-of-features for large scale image search. IJCV **87** (2010) 316–336
13. Hua, G., Brown, M., Winder, S.: Discriminant embedding for local image descriptors. In: Proc. ICCV. (2007)
14. Mikolajczyk, K., Matas, J.: Improving sift for fast tree matching by optimal linear projection. In: Proc. ICCV. (2007)
15. Agarwal, S., Snavely, N., Simon, I., Seitz, S., Szeliski, R.: Building Rome in a day. In: Proc. ICCV. (2009)
16. Chum, O., Matas, J.: Large-scale discovery of spatially related images. IEEE PAMI **32** (2010) 371–377
17. Li, X., Wu, C., Zach, C., Lazebnik, S., Frahm, J.M.: Modeling and recognition of landmark image collections using iconic scene graphs. In: Proc. ECCV. (2008)
18. Ferrari, V., Tuytelaars, T., Van Gool, L.: Simultaneous object recognition and segmentation by image exploration. In: Proc. ECCV. (2004)
19. Cech, J., Matas, J., Perdoch, M.: Efficient sequential correspondence selection by cosegmentation. In: Proc. CVPR. (2008)
20. Godsil, C., Royle, G.: Algebraic Graph Theory. Springer (2001)
21. Muja, M., Lowe, D.G.: Fast approximate nearest neighbors with automatic algorithm configuration. In: VISSAPP. (2009)
22. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. ACM Press, ISBN: 020139829 (1999)
23. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: Proc. CVPR. (2007)
24. Jégou, H., Douze, M., Schmid, C.: On the burstiness of visual elements. In: Proc. CVPR. (2009)