Query Adaptive Similarity for Large Scale Object Retrieval

Danfeng Qin Christian Wengert Luc van Gool
ETH Zürich, Switzerland
{qind,wengert,vangool}@vision.ee.ethz.ch

Abstract

Many recent object retrieval systems rely on local features for describing an image. The similarity between a pair of images is measured by aggregating the similarity between their corresponding local features. In this paper we present a probabilistic framework for modeling the feature to feature similarity measure. We then derive a query adaptive distance which is appropriate for global similarity evaluation. Furthermore, we propose a function to score the individual contributions into an image to image similarity within the probabilistic framework. Experimental results show that our method improves the retrieval accuracy significantly and consistently. Moreover, our result compares favorably to the state-of-the-art.

1. Introduction

We consider the problem of content-based image retrieval for applications such as object recognition or similar image retrieval. This problem has applications in web image retrieval, location recognition, mobile visual search, and tagging of photos.

Most of the recent state-of-the-art large scale image retrieval systems rely on local features, in particular the SIFT descriptor [14] and its variants. Moreover, these descriptors are typically used jointly with a bag-of-words (BOW) approach, reducing considerably the computational burden and memory requirements in large scale scenarios.

The similarity between two images is usually expressed by aggregating the similarities between corresponding local features. However, to the best of our knowledge, few attempts have been made to systematically analyze how to model the employed similarity measures.

In this paper we present a probabilistic view of the feature to feature similarity. We then derive a measure that is adaptive to the query feature. We show - both on simulated and real data - that the Euclidean distance density distribution is highly query dependent and that our model adapts the original distance accordingly. While it is difficult to know the distribution of true correspondences, it is actually quite easy to estimate the distribution of the distance of non-corresponding features. The expected distance to the non-corresponding features can be used to adapt the original distance and can be efficiently estimated by introducing a small set of random features as negative examples. Furthermore, we derive a global similarity function that scores the feature to feature similarities. Based on simulated data, this function approximates the analytical result.

Moreover, in contrast to some existing methods, our method does not require any parameter tuning to achieve its best performance on different datasets. Despite its simplicity, experimental results on standard benchmarks show that our method improves the retrieval accuracy consistently and significantly and compares favorably to the state-of-the-art.

Furthermore, all recently presented post-processing steps can still be applied on top of our method and yield an additional performance gain.

The rest of this paper is organized as follows. Section 2 gives an overview of related research. Section 3 describes our method in more detail. The experiments for evaluating our approach are described in Section 4. Results in a large scale image retrieval system are presented in Section 5 and compared with the state-of-the-art.

2. Related Work

Most of the recent works addressing the image similarity problem in image retrieval can be roughly grouped into three categories.

Feature-feature similarity The first group mainly works on establishing local feature correspondence. The most famous work in this group is the bag-of-words (BOW) approach [24]. Two features are considered to be similar if they are assigned to the same visual word. Despite the efficiency of the BOW model, the hard visual word assignment significantly reduces the discriminative power of the local features. In order to reduce quantization artifacts, [20] proposed to assign each feature to multiple visual words. In contrast, [8] rely on using smaller codebooks but in conjunction with short binary codes for each local feature, refining the feature matching within the same Voronoi cell. Additionally, product quantization [12] was used to estimate the pairwise Euclidean distance between features, and the top k nearest neighbors of a query feature is considered as matches. Recently, several researchers have addressed the problem of the Euclidean distance not being the optimal similarity measure in most situations. For instance in [16], a probabilistic relationship between visual words is learned from a large collection of corresponding feature tracks. Alternatively, in [21], they learn a projection from the original feature space to a new space, such that Euclidean metric in this new space can appropriately model feature similarity.

Intra-image similarity The second group focuses on effectively weighting the similarity of a feature pair considering its relationship to other matched pairs.

Several authors exploit the property that the local features inside the same image are not independent. As a consequence, a direct accumulation of local feature similarities can lead to inferior performance. This problem was addressed in [4] by down-weighting the contribution of non-incidentally co-occurring features. In [9] this problem was approached by re-weighting features according to their burstiness measurement.

As the BOW approach discards spatial information, a scoring step can be introduced which exploits the property that the true matched feature pairs should follow a consistent spatial transformation. The authors of [19] proposed to use RANSAC to estimate the homography between images, and only count the contribution of feature pairs consistent with this model. [26] and [23] propose to quantize the image transformation parameter space in a Hough voting manner, and let each matching feature pair vote for its correspondent parameter cells. A feature pair is considered valid if it supports the cell of maximum votes.

Inter-image similarity Finally, the third group addresses the problem of how to improve the retrieval performance by exploiting additional information contained in other images in the database, that depict the same object as the query image. [5] relies on query expansion. That is, after retrieving a set of spatially verified database images, this new set is used to query the system again to increase recall. In [22], a set of relevant images is constructed using k-reciprocal nearest neighbors, and the similarity score is evaluated on how similar a database image is to this set.

Our work belongs to the first group. By formulating the feature-feature matching problem in a probabilistic framework, we propose an adaptive similarity to each query feature, and a similarity function to approximate the quantitative result. Although the idea of adapting similarity by dissimilarity has already been exploited in [11][17], we propose to measure dissimilarity by mean distance of the query to a set of random features, while theirs use k nearest neighbors (kNN). According to the fact that, in a realistic dataset, different objects may have different numbers of relevant images, it is actually quite hard for the kNN based method to find an generalized k for all queries. Moreover, as kNN is an order statistic, it could be sensitive to outliers and can't be used reliably as an estimator in realistic scenarios. In contrast, in our work, the set of random features could be considered as a clean set of negative examples, and the mean operator is actually quite robust as shown later.

Considering the large amount of data in a typical large scale image retrieval system, it is impractical to compute the pairwise distances between high-dimensional original feature vectors. However, several approaches exist to relieve that burden using efficient approximations such as [12, 13, 3, 6]. For simplicity, we adopt the method proposed in [12] to estimate the distance between features.

3. Our Approach

In this section, we present a theoretical framework for modeling the visual similarity between a pair of features, given a pairwise measurement. We then derive an analytical model for computing the accuracy of the similarity estimation in order to compare different similarity measures. Following the theoretical analysis, we continue the discussion on simulated data. Since the distribution of the Euclidean distance varies enormously from one query feature to another, we propose to normalize the distance locally to obtain similar degree of measurement across queries. Furthermore, using the adaptive measure, we quantitatively analyze the similarity function on the simulated data and propose a function to approximate the quantitative result. Finally, we discuss how to integrate our findings into a retrieval system.

3.1. A probabilistic view of similarity estimation

We are interested in modeling the visual similarity between features based on a pairwise measurement.

Let us denote as x_i the local feature vectors from a query image and as $\mathcal{Y} = \{y_1, ..., y_j, ..., y_n\}$ a set of local features from a collection of database images. Furthermore, let $m(x_i, y_j)$ denote a pairwise measurement between x_i and y_j. Finally $T(x_i)$ represents the set of features which are visually similar to x_i, and $F(x_i)$ as the set of features which are dissimilar to x_i. Instead of considering whether y_j is similar to x_i and how similar they look, we want to evaluate how likely y_j belongs to $T(x_i)$ given a measure m. This can be modeled as follows

$$f(x_i, y_j) = p(y_j \in T(x_i) \mid m(x_i, y_j)) \tag{1}$$

For simplicity, we denote $m_j = m(x_i, y_j)$, $T_i = T(x_i)$, and $F_i = F(x_i)$. As y_j either belongs to T_i or F_i, we have

$$p(y_j \in T_i \mid m_j) + p(y_j \in F_i \mid m_j) = 1 \tag{2}$$

Furthermore, according to the Bayes Theorem

$$p(y_j \in T_i \mid m_j) = \frac{p(m_j \mid y_j \in T_i) \times p(y_j \in T_i)}{p(m_j)} \tag{3}$$

and

$$p(y_j \in F_i \mid m_j) = \frac{p(m_j \mid y_j \in F_i) \times p(y_j \in F_i)}{p(m_j)} \quad (4)$$

Finally, by combining Equations 2, 3 and 4 we get

$$p(y_j \in T_i \mid m_j) = \left(1 + \frac{p(m_j \mid y_j \in F_i)}{p(m_j \mid y_j \in T_i)} \times \frac{p(y_j \in F_i)}{p(y_j \in T_i)}\right)^{-1} \quad (5)$$

For large datasets the quantity $p(y_j \in T_i)$ can be modeled by the occurrence frequency of x_i. Therefore, $p(y_j \in T_i)$ and $p(y_j \in F_i)$ only depend on the query feature x_i.

In contrast, $p(m_j \mid y_j \in T_i)$ and $p(m_j \mid y_j \in F_i)$ are the probability density functions of the distribution of m_j, for $\{y_j \mid y \in T_i\}$ and $\{y_j \mid y \in F_i\}$. We will show in Section 3.3, how to generate simulated data for estimating these distributions. In Section 3.5 we will further exploit these distributions in our framework.

3.2. Estimation accuracy

Since the pairwise measurement between features is the only observation for our model, it is essential to estimate its reliability. Intuitively, an optimal measurement should be able to perfectly separate the true correspondences from the false ones. In other words, the better the measurement distinguishes the true correspondences from the false ones, the more accurately the feature similarity based on it can be estimated. Therefore, the measurement accuracy can be modeled as the expected pureness. Let $\mathcal{T}$ be a collection of all matched pairs of features, i.e,

$$\mathcal{T} = \{(x, y) \mid y \in T(x))\} \quad (6)$$

The probability that a pair of features is a true match given the measurement value z can be expressed as

$$p(\mathcal{T} \mid z) = p((x, y) \in \mathcal{T} \mid m(x, y) = z) \quad (7)$$

Furthermore, the probability of observing a measurement value z given a corresponding feature pair is

$$p(z \mid \mathcal{T}) = p(m(x, y) = z \mid (x, y) \in \mathcal{T}) \quad (8)$$

Then, the accuracy for the similarity estimation is

$$Acc(m) = \int_{-\infty}^{\infty} p(\mathcal{T} \mid z) \times p(z \mid \mathcal{T}) \mathrm{d}z \quad (9)$$

with m some pairwise measurement and $Acc(m)$ the accuracy of the model based on m. Since

$$p(\mathcal{T} \mid z) \leq 1 \text{ and } \int_{-\infty}^{\infty} p(z \mid \mathcal{T}) \mathrm{d}z = 1 \quad (10)$$

the accuracy of a measure m is

$$Acc(m) \leq 1 \quad (11)$$

and

$$Acc(m) = 1 \Leftrightarrow p(\mathcal{T} \mid z) = 1, \forall p(z \mid \mathcal{T}) > 0 \quad (12)$$

This measure allows to compare the accuracy of different distance measurements as will be shown in the next section.

3.3. Ground truth data generation

In order to model the property of $T(x_i)$, we simulate corresponding features using the following method: First, regions $r_{i,0}$ are detected on a random set of images by the Hessian Affine detector[15]. Then, we apply numerous random affine warpings (using the affine model proposed by ASIFT [25]) to $r_{i,0}$, and generate a set of related regions. Finally, SIFT features are computed on all regions resulting in $\{x_{i,1}, x_{i,2}, ..., x_{i,n}\}$ as a subset of $T(x_{i,0})$.

The parameters for the simulated affine transformation are selected randomly and some random jitter is added to model the detection errors occurring in a practical setting. The non-corresponding features $F(x_i)$ are simply generated by selecting $500K$ random patches extracted from a different and unrelated dataset. In this way, we also generate a dataset $\mathcal{D}$ containing $100K$ matched pairs of features from different images, and $1M$ non-matched paris. Figure 1 depicts two corresponding image patches randomly selected from the simulated data.

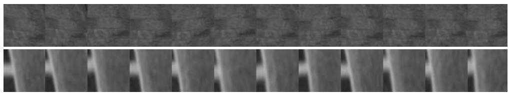


Figure 1. Corresponding image patches for two randomly selected points of the simulated data

3.4. Query adaptive distance

It has been observed that the Euclidean distance is not an appropriate measurement for similarity [21, 16, 11]. We argue that the Euclidean distance is a robust estimator when normalized locally.

As an example, Figure 2 depicts the distributions of the Euclidean distance of the corresponding and non corresponding features for the two different interest points shown in Figure 1. For each sample point x_i, we collected a set of 500 corresponding features $T(x_i)$ using the procedure from Section 3.3 and a set of $500K$ random non-corresponding features $F(x_i)$. It can be seen, that the Euclidean distance separates the matching from the non-matching features quite well in the local neighborhood of a given query feature x_i.

However, by averaging the distributions of $T(x_i)$ and $F(x_i)$ respectively for all queries x_i, the Euclidean distance loses its discriminative power. This explains, why the Euclidean distance has inferior performance in estimating vi-

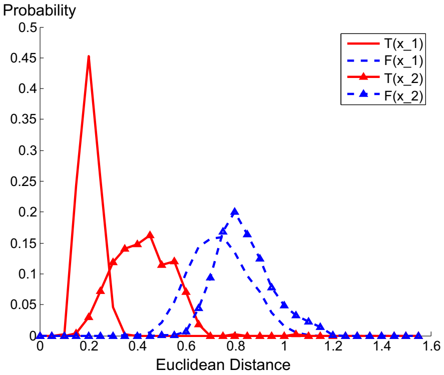


Figure 2. Distribution of the Euclidean distance for two points from the simulated data. The solid lines show the distribution for corresponding features $T(x_i)$, whereas the dotted line depict non-corresponding ones $F(x_i)$.

sual similarity from a global point of view. A local adaptation is therefore necessary to recover the discriminability of the Euclidean Distance.

Another property can also be observed in Figure 2: if a feature has a large distance to its correspondences, it also has a large distance to the non-matching features. By exploiting this property, a normalization of the distance can be derived for each query feature

$$d_n(x_i, y_j) = d(x_i, y_j)/N_{d(x_i)} \tag{13}$$

where $d_n(\cdot,\cdot)$ represents the normalized distance, $d(\cdot,\cdot)$ represents the original Euclidean distance and $N_{d(x_i)}$ represents the expected distance of x_i to its non-matching features. It is intractable to estimate the distance distribution between all feature and their correspondences, but it is simple to estimate the expected distance to non-corresponding features. Since the non-corresponding features are independent from the query, a set of randomly sampled, thus unrelated features can be used to represent the set of non-correspondent features to each query. Moreover, if we assume the distance distribution of the non-corresponding set to follow a normal distribution $\mathcal{N}(\mu, \sigma)$, then the estimation error of its mean based on a subset follows another normal distribution $\mathcal{N}(0, \sigma/N)$, with N the size of the subset. Therefore, $N_{d(x_i)}$ can be estimated sufficiently well and very efficiently from even a small set of random, i.e. non-corresponding features.

The probability that an unknown feature matches to the query one when observing their distance z can be modeled as,

$$\begin{aligned} p(\mathcal{T} \mid z) &= \frac{N_T \times p(z \mid \mathcal{T})}{N_T \times p(z \mid \mathcal{T}) + N_F \times p(z \mid \mathcal{F})} \\ &= \{1 + \frac{N_F}{N_T} \times \frac{p(z \mid \mathcal{F})}{p(z \mid \mathcal{T})}\}^{-1} \end{aligned} \tag{14}$$

with N_T and N_F the number of corresponding and non-corresponding pairs respectively. In practical settings, N_F is usually many orders of magnitude larger than N_T. Therefore, once $p(z \mid \mathcal{F})$ starts getting bigger than 0, $p(\mathcal{T} \mid z)$ rapidly decreases, and the corresponding features would be quickly get confused with the non-corresponding ones.

Figure 3 illustrates how the adaptive distance recovers more correct matches compared to the Euclidean distance.

Moreover, by assuming that $N_F/N_T \approx 1000$ the measurement accuracy following Equation 9 can be computed. For the Euclidean distance, the estimation accuracy is 0.7291, and for the adaptive distance, the accuracy is 0.7748. Our proposed distance thus significantly outperforms the Euclidean distance.

3.5. Similarity function

In this section, we show how to derive a globally appropriate feature similarity in a quantitative manner. After having established the distance distribution of the query adaptive distance in the previous section, the only unknown in Equation 5 remains $\frac{p(y_j \in F_i)}{p(y_j \in T_i)}$.

As discussed in Section 3.1, this quantity is inversely proportional to the occurrence frequency of x_i, and it is generally a very large term. Assuming $c = \frac{p(y_j \in F_i)}{p(y_j \in T_i)}$ being between 10 and 100000, the full similarity function can be estimated and is depicted in Figure 4.

The resulting curves follow an inverse sigmoid form such that the similarity is 1 for $d_n \to 0$ and 0 if $d_n \to 1$. They all have roughly the same shape and differ approximately only by an offset. It is to be noted, that they show a very sharp transition making it very difficult to correctly estimate the transition point and thus to achieve a good separation between true and false matches.

In order to reduce the estimation error due to such sharp transitions, a smoother curve would be desirable. Since the distance distributions are all long-tailed, we have fitted different kinds of exponential functions to those curves. However, we observe similar results. For the reason of simplicity, we choose to approximate the similarity function as

$$f(x_i, y_j) = \exp(-\alpha \times d_n(x_i, y_j)^4) \tag{15}$$

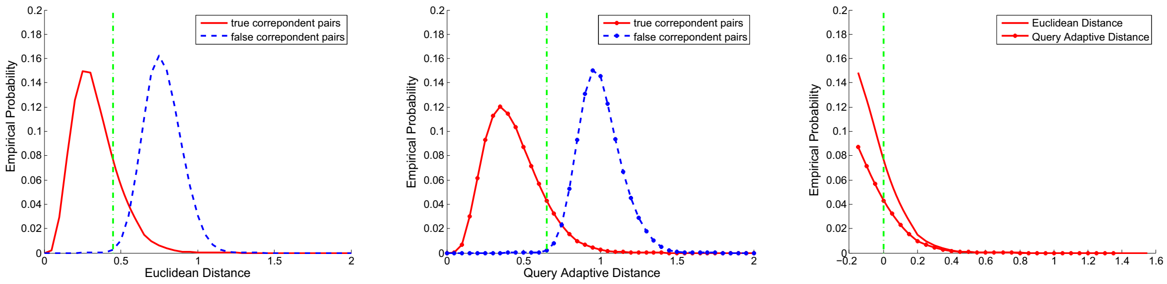
As can be seen in Figure 4, this curve is flatter and covers approximately the full range of possible values for c.

In Equation 15, α can be used to tune the shape of the final function and roughly steers the slope of our function, we achieved best results with $\alpha = 9$ and keep this value throughout all experiments.

In the next section, the robustness of this function in real image retrieval system will be evaluated.

3.6. Overall method

In this section we will integrate the query adaptive distance measurement and the similarity function presented



(a) Euclidean Distance (b) Query Adaptive Distance (c) Comparison

Figure 3. The comparison of our adaptive distance to the Euclidean distance on dataset $\mathcal{D}$. The solid lines are the distance distribution of the matched pairs, and the dotted lines are the distance distribution of non-matched pairs. The green dashed lines denotes where the probability of the non-matching distance exceed 0.1%, i.e, the non-matching feature is very likely to dominate our observation. A comparison of the right tails of both distributions is shown in (c).

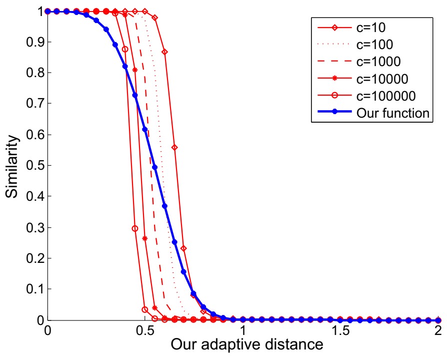


Figure 4. Feature similarity evaluated on dataset $\mathcal{D}$. Red lines are the visual similarity for different c evaluated on the simulated data. The blue line is our final similarity function with $\alpha = 9$.

before into an image retrieval system.

Let the visual similarity between the query image $q = \{x_1, ..., x_m\}$ and a database image $d = \{y_1, ..., y_n\}$ be

$$sim(q,d) = \sum_{i=1}^{m} \sum_{j=1}^{n} f(x_i, y_j) \tag{16}$$

with $f(x_i, y_j)$ the pairwise feature similarity as in Equation 15. As mentioned before, $d_n(x_i, y_j)$ and $N_{d(x_i)}$ are estimated using the random set of features.

For retrieval, we use a standard bag-of-words inverted file. However, in order to have an estimation of the pairwise distance $d(x_i, y_j)$ between query and database features, we add a product quantization scheme as in [12] and select the same parameters as the original author. The feature space is firstly partitioned into $N_c = 20'000$ Voronoi cells according to a coarse quantization codebook $\mathcal{K}_c$. All features located in the same Voronoi cell are grouped into the same inverted list. Each feature is further quantized with respect to its coarse quantization centroid. That is, the residual between the feature and its closest centroid is equally split into $m = 8$ parts and each part is separately quantized according to a product quantization codebook $\mathcal{K}_p$ with $N_p = 256$ centroids. Then, each feature is encoded using its related image identifier and a set of quantization codes, and is stored in its corresponding inverted list.

We select random features from Flickr and add 100 of them to each inverted list. For performance reasons, we make sure that the random features are added to the inverted list before adding the database vectors.

At query time, all inverted lists whose related coarse quantization centers are in the k nearest neighborhood of the query vector are scanned.

With our indexing scheme, the distances to non-matching features are always computed first, with their mean value being directly $N_{d(x_i)}$. Then, the query adaptive distance $d_n(x_i, y_j)$ to each database vector can directly be computed as in Equation 13. In order to reduce unnecessary computation even more, a threshold β is used to quickly drop features whose Euclidean distance is larger than $\beta \times N_{d(x_i)}$. This parameter has little influence on the retrieval performance, but reduces the computational load significantly. Its influence is evaluated in Section 4.

As pointed out by [9], local features of an image tend to occur in bursts. In order to avoid multiple counting of statistically correlated features, we incorporate both "intra burstiness" and "inter burstiness" normalization [9] to re-weight the contributions of every pair of features. The similarity function thus changes to

$$sim(q,d) = \sum_{i=1}^{m} \sum_{j=1}^{n} w(x_i, y_j) f(x_i, y_j) \tag{17}$$

with $w(x_i, y_j)$ the burstiness weighting.

4. Experiments

In this part, we first introduce the evaluation protocol. Then we give some implementation details of our algorithm. Furthermore, we discuss the influence of each parameter and experimentally select the best ones. Finally, we evaluate each part of our method separately.

4.1. Datasets and performance evaluation protocol

We evaluated our method on the Oxford5k[19], Paris[20], Holidays[8] and Oxford105k dataset. Oxford105k consists of Oxford5k and 100285 distractor images. The 100285 distractor images are a set of random images that we downloaded from Flickr having the same resolution of 1024×768 as the original Oxford5k dataset.

We follow the same evaluation measurement method as proposed in the original publications, that is, the mean average precision (mAP) is calculated as the overall performance of the retrieval system.

4.2. Implementation details

Preprocessing For all experiments, all images are resized such that their maximum resolution is 1024×768. In each image, interest points are detected using the Hessian Affine detector and a SIFT descriptor is computed around each point. As in [2] a square root scaling is applied to each SIFT vector, yielding a significantly better retrieval performance when using the Euclidean metric.

Codebook training The vocabularies were trained on an independent dataset of images randomly downloaded from Flickr in order to prevent overfitting to the datasets.

Random feature dataset preparation Random images from Flickr (however different from the codebook training dataset) are used to generate the random feature dataset.

4.3. Parameter selection

In this section, we evaluate the retrieval performance of our approach on the Oxford5K dataset for different settings of parameters. There are two parameters in our method: the number of random features in each inverted list, and the cut-off threshold β for filtering out features whose contribution is negligible.

The influence of the number of the random features Table 1 shows the retrieval performance by varying the number of random features for each inverted list. The performance remains almost constant for a very large range of number of random features. This supports the assumption, that the mean distance of a query feature to the dissimilar features can be robustly estimated even with a small number of random features. We select 100 random features per inverted list throughout the rest of this paper.

Length	50	100	500	1000	10000
mAP	0.739	0.739	0.739	0.739	0.738

Table 1. Influence of the size of the random feature set for each inverted list on Oxford5k

β	0.80	0.85	0.9	0.95
similarity score	0.025	0.009	0.003	0.001
#selected features	13	43	124	292
mAP	0.733	0.739	0.740	0.739

Table 2. Influence of the cut-off value β on Oxford5k

The influence of the cut-off threshold β Table 2 shows that features with a distance larger than $\beta \times N_{d(x_i)}$ with $\beta \in [0.8, 0.95]$ have almost no contribution to the retrieval performance. In order to reduce the number of updates of the scoring table, we select $\beta = 0.85$ for all experiments.

4.4. Effectiveness of our method

Local adaptive distance In order to compare the adaptive distance function to the Euclidean distance, we use a threshold for separating matching and non-matching features. Figure 4.4 shows the retrieval performance for a varying threshold both for the Euclidean distance as well as for the adaptive distance. Overall, the best mAP using the adaptive distance is 3% better than the Euclidean distance. Furthermore, the adaptive distance is less sensitive when selecting a non-optimal threshold. It is to be noted that in the final setup, our method does not require any thresholding.

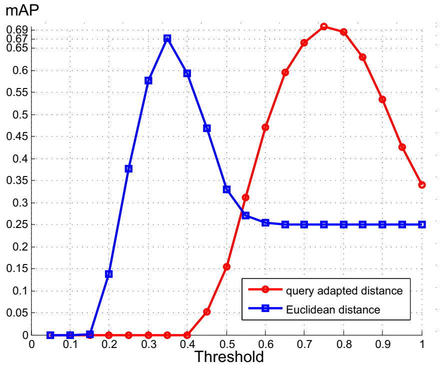


Figure 5. Comparison of our adaptive distance with Euclidean distance on Oxford5k dataset

Contributions of other steps In order to justify the contribution of other steps that are contained in our method, we evaluate the performance of our method by taking them out of the pipeline. For the experiment on Oxford5k, we find out that without the feature scaling, mAP will drop from 0.739 to 0.707, while without burstiness weighting, mAP will drop to 0.692. With multi-assignment only on the query side, mAP can increase from 0.739 to 0.773 for $MA = 5$, and 0.780 for $MA = 10$. MA denotes the number of inverted lists that are traversed per query feature.

5. Results

Throughout all experiments, the set of parameters was fixed to the values obtained in the previous section and vocabularies were trained always on independent datasets. Table 3 shows the retrieval performance on all typical benchmarks both with single assignment (SA) and multi-assignment ($MA = 10$). As expected, multi-assignment (scanning of several inverted lists) reduces the quantization artifacts and improves the performance consistently, however, in exchange for more computational load.

Furthermore, we applied an image level post-processing step on top of our method. We choose to use reciprocal nearest neighbors (RNN) [22], for the reason that it can be easily integrated on top of a retrieval system independently from the image similarity function. We adopt the publicly available code [1] provided by the original authors and the default settings. RNN significantly improves the results on Oxford5K and Paris datasets, but slightly lowers the result on Holidays. Considering that RNN tries to exploit additional information contained in other relevant database images, which are scarce in Holidays (in average only 2 to 3 relevant database images per query), it is difficult for query expansion methods to perform much better.

Dataset	SA	MA	MA + RNN
Oxford5k	0.739	0.780	0.850
Oxford105k	0.678	0.728	0.816
Paris	0.703	0.736	0.855
Holidays	0.814	0.821	0.801

Table 3. Performance of our method on public datasets.

5.1. Comparison with state-of-the-art

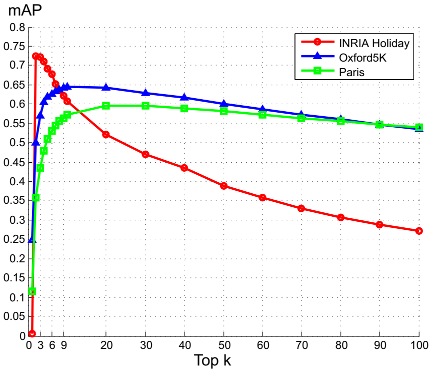


Figure 6. Retrieval Performance by using top k nearest neighbor as similar features [12]

We first compare the performance of our method to [12] which relies on using the top k nearest neighbors of the Euclidean distance for selecting the similar features of a query. This work is closest to ours, both in memory overhead and computational complexity. It can be seen in Figure 6, that no single k maximizes the performance for all datasets, showing that this parameter is very sensitive to the data. Moreover, our method outperforms the peak results from [12] consistently by roughly 10 points of mAP.

Table 4 shows the comparison to several other methods without applying any image-level post-processing step. As pointed out by [10], training a vocabulary on independent data rather than the evaluated dataset itself can better represent the search performance in a very large dataset. We only compare to state-of-the-art methods using codebooks trained on independent datasets. We achieve the best performance for Oxford5k, Oxford105k, and Holidays and fall only slightly behind [16] on Paris.

Dataset	Ours	[16]	[7]	[18]
Oxford5k	**0.780**	0.742	0.704	0.725
Oxford105k	**0.728**	0.675*	-	0.652
Paris	0.736	**0.749**	-	-
Holidays	**0.821**	0.749**	0.817	0.769/0.818**

Table 4. Comparisons with state-of-the-art methods without applying image level post-processing. * indicates the score of merging Oxford5k and Paris and 100K distractor images. ** denotes the result obtained by manually rotating all images in the Holidays dataset to be upright.

Furthermore, Table 5 gives a comparison for the results when additional image-level post-processing steps are applied. We argue, that any post-processing step can directly benefit from our method and illustrate with RNN as example that the best performance can be achieved.

Dataset	Ours+RNN	[16]	[18]	[2]
Oxford5k	**0.850**	0.849	0.822	0.809
Oxford105k	**0.816**	0.795	0.772	0.722
Paris	**0.855**	0.824	-	0.765
Holidays	**0.801**	0.758**	0.78	-

Table 5. Comparisons with the state of art methods with post-processing in image level. ** denotes the result obtained by manually rotating all images in the Holidays dataset to be upright.

In all of the previous experiments, each feature costs 12 bytes of memory. Specifically, 4 bytes is used for the image identifier and 8 bytes for the quantization codes. As [11] mainly show results using more bytes for feature encoding, we also compare our method to theirs with more bytes per feature. As shown in Table 6, using more bytes further improves the retrieval results. Even with less bytes than [11], better performance is achieved on all datasets.

In all experiments, we compare favorably to the state-of-the-art by exploiting a simple similarity function without any parameter tuning for each dataset. The good results

Dataset	Ours	Ours	[11]
Bytes	12	36	44
Oxford5k	0.780	**0.831**	0.764
Paris	0.736	**0.756**	0.728
Holidays	0.821	**0.844**	**0.844**

Table 6. Comparison to [11] using more bytes per feature.

justify our previous analysis and the effectiveness of our method.

5.2. Computational Complexity

In a small scale experiment, e.g for Oxford5k, we observe that our method is 30% faster than the original product quantization algorithm[12] while traversing the inverted lists, for the reason that our method requires no heap structure. However, for a large scale experiment, we observe similar timing of our method to theirs as each inverted list contains a very long list of database features, and thus the computation of the Euclidean distance will dominate the computational time.

6. Conclusion

In this paper, we present a probabilistic framework for the feature to feature similarity for high-dimensional local features such as SIFT. We then propose a query adaptive feature to feature distance measurement and derive a global image to image similarity function. Despite the simplicity of this approach, it achieves consistently good results on all evaluated datasets, supporting the validity of our model. Furthermore, it does not require parameter tuning to achieve optimal performance.

Acknowledgements The authors gratefully acknowledge support from KTI project and the ERC project VarCity.

References

[1] http://www.vision.ee.ethz.ch/~qind/HelloNeighbor.html. 7

[2] R. Arandjelović and A. Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, 2012. 6, 7

[3] A. Babenko and V. S. Lempitsky. The inverted multi-index. In *CVPR*, pages 3069–3076, 2012. 2

[4] O. Chum and J. Matas. Unsupervised discovery of co-occurrence in sparse high dimensional data. In *CVPR*, 2010. 2

[5] 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 *ICCV*, 2007. 2

[6] Y. Hwang, B. Han, and H.-K. Ahn. A fast nearest neighbor search algorithm by nonlinear embedding. In *CVPR*, pages 3053–3060, 2012. 2

[7] M. Jain, H. Jégou, and P. Gros. Asymmetric Hamming Embedding. In *ACM Multimedia*, Scottsdale, United States, October 2011. QUAERO. 7

[8] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, October 2008. 1, 6

[9] H. Jégou, M. Douze, and C. Schmid. On the burstiness of visual elements. In *CVPR*, June 2009. 2, 5

[10] H. Jégou, M. Douze, and C. Schmid. Improving bag-of-features for large scale image search. *IJCV*, 87(3):316–336, February 2010. 7

[11] H. Jégou, M. Douze, and C. Schmid. Exploiting descriptor distances for precise image search. Research report, INRIA Rennes, June 2011. 2, 3, 7, 8

[12] H. Jégou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 33(1):117–128, January 2011. 1, 2, 5, 7, 8

[13] H. Jégou, R. Tavenard, M. Douze, and L. Amsaleg. Searching in one billion vectors: re-rank with source coding. In *ICASSP*, Prague Czech Republic, 2011. 2

[14] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 1

[15] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *IJCV*, 60(1), 2004. 3

[16] A. Mikulik, M. Perdoch, O. Chum, and J. Matas. Learning a fine vocabulary. In *ECCV*, 2010. 2, 3, 7

[17] D. Omercevic, O. Drbohlav, and A. Leonardis. High-dimensional feature matching: employing the concept of meaningful nearest neighbors. In *ICCV*, 2007. 2

[18] M. Perdoch, O. Chum, and J. Matas. Efficient representation of local geometry for large scale object retrieval. In *CVPR*, June 2009. 7

[19] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007. 2, 6

[20] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*, 2008. 1, 6

[21] J. Philbin, M. Isard, J. Sivic, and A. Zisserman. Descriptor learning for efficient retrieval. In *ECCV*, 2010. 2, 3

[22] D. Qin, S. Gammeter, L. Bossard, T. Quack, and L. van Gool. Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors. In *CVPR*. IEEE, 2011. 2, 7

[23] X. Shen, Z. Lin, J. Brandt, S. Avidan, and Y. Wu. Object retrieval and localization with spatially-constrained similarity measure and k-nn re-ranking. In *CVPR*, 2012. 2

[24] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, 2003. 1

[25] G. Yu and J.-M. Morel. ASIFT: An Algorithm for Fully Affine Invariant Comparison. *Image Processing On Line*, 2011, 2011. 3

[26] Y. Zhang, Z. Jia, and T. Chen. Image retrieval with geometry-preserving visual phrases. In *CVPR*, 2011. 2