A contextual dissimilarity measure for accurate and efficient image search

Hervé Jégou, Harzallah Hedi, Cordelia Schmid

A contextual dissimilarity measure for accurate and efficient image search

Herve Jegou
INRIA-LEAR
herve.jegou@inria.fr

Hedi Harzallah
INRIA-LEAR
hedi.harzallah@inria.fr

Cordelia Schmid
INRIA-LEAR
cordelia.schmid@inria.fr

Abstract

In this paper we present two contributions to improve accuracy and speed of an image search system based on bag-of-features: a contextual dissimilarity measure (CDM) and an efficient search structure for visual word vectors.

Our measure (CDM) takes into account the local distribution of the vectors and iteratively estimates distance correcting terms. These terms are subsequently used to update an existing distance, thereby modifying the neighborhood structure. Experimental results on the Nistér-Stewénius dataset show that our approach significantly outperforms the state-of-the-art in terms of accuracy.

Our efficient search structure for visual word vectors is a two-level scheme using inverted files. The first level partitions the image set into clusters of images. At query time, only a subset of clusters of the second level has to be searched. This method allows fast querying in large sets of images. We evaluate the gain in speed and the loss in accuracy on large datasets (up to 1 million images).

1. Introduction

In this paper we address the problem of finding images containing the same object or scene seen from different viewpoints, with different background and occlusion. Initial approaches used simple voting based techniques [9, 10]. More recently they were extended based on the bag-of-features image representation [12, 14]. Our paper builds upon these approaches and presents two contributions.

First, we introduce a contextual dissimilarity measure (CDM) which takes into account the neighborhood of a point. This measure is iteratively obtained by regularizing the average distance of each point to its neighborhood. Our CDM is learned in a unsupervised manner, in contrast with a large number of works which learn the distance measure from a set of training images [2, 4, 6, 16]. However, in the context of a large database, supervised learning is simply too time consuming. Furthermore, in contrast to category classification where class members are clearly defined and represented by a sufficiently large set, this does not necessarily hold in our case. Note that the different weighting schemes from text retrieval, namely the term frequency/inverse document frequency weighting [13], can be seen as a simple way to improve the distance [12, 14] in an unsupervised manner. Experimental results show that the gain due to our distance is significantly higher that the one due to a weighting scheme. Note that the two approaches can be combined.

Second, we introduce an efficient search technique for visual word vectors. There is a large body of approximate search techniques [1, 3, 5]. A few works have addressed this problem in computer vision. For example, Nistér et Stewénius [12] propose an efficient method to assign descriptors to visual words based on a hierarchical k-means approach. Similarly, Moosmann et al. [11] use a forest of random trees to rapidly and precisely assign descriptors to clusters. However, to our knowledge nobody has addressed the problem of rapidly accessing visual word frequency vectors. The inverted file system [12, 14, 17] avoids comparing the feature vectors individually by storing for each visual word the set of image references in which this visual word appears. However, this approach is still linear in the number of images in the dataset. In contrast our approach subdivides the dataset into a number of subsets and uses a two-level inverted file system. The subdivision is based on a k-medoids algorithm which preserves the sparsity of the visual word vectors and is compatible with the CDM. We first search for the most similar cluster(s) represented by their centers and then search through the set of images belonging to the cluster. In our experiments we measure the trade-off between the number of clusters considered and the accuracy. A good accuracy is observed when searching through a small number of closest clusters.

This paper is organized as follows. Section 2 reviews the bag-of-words image retrieval approach of [14] and describes some variants. The CDM design is described in Section 3 and the clustering-based strategy that improves the efficiency in Section 4. The relevance of the approach and the parameter analysis is shown in Section 5.

2. Overview of the image search scheme

In the following, we present the different steps of our image search framework, similar to [14].

Descriptors: The n database images are described with local descriptors. We combine the SIFT descriptor [9] with the affine Hessian region extractor [10]. As a variant, the 128-dimensional SIFT descriptors are reduced to 36-dimensional vectors with principal component analysis (PCA), similar to [8].

Visual words: The visual words quantize the space of descriptors. Here, we use the k-means algorithm to obtain the visual vocabulary. Note that, although the generation of the visual vocabulary is performed off-line, it is time consuming and becomes intractable as the number of visual word increases (> 100000). The fast hierarchical clustering described in [12] allows the generation of such huge vocabularies in a reasonable time.

Assigning the descriptors to visual words: Each SIFT descriptor of a given image i is assigned to the closest visual word. The histogram of visual word occurrences is subsequently normalized with the L1 norm, generating a frequency vector $f_i = (f_{i,1}, \ldots, f_{i,k})$. As a variant, instead of choosing the nearest neighbor, a given SIFT descriptor is assigned to the k-nearest visual words. This variant will be referred to as multiple assignment (MA) in the experiments.

Weighting frequency vectors: The frequency vector's components are then weighted using a strategy similar to the one in [12]. Denoting by n the number of images in the database and by n_j the number of images containing the j^{th} visual word, the j^{th} component $w_{i,j}$ associated with image i is given by

$$w_{i,j} = f_{i,j} \log \frac{n}{n_j}. \quad (1)$$

The resulting *visual word frequency vector* $w_i = (w_{i,1}, \ldots, w_{i,j}, \ldots, w_{i,k})$, or simply visual word vector, is a compact representation of the image.

Distance: Searching similar images in the database amounts to computing the visual word vector w_q of the query and to finding the description vector(s) w_i minimizing $d(w_q, w_i)$, where the relation $d(\cdot, \cdot)$ is a distance on the visual word vector space. Note that the weighting scheme previously described can be seen as part of the distance definition.

Our contextual dissimilarity measure described in Section. 3, operates at this stage. It updates a given distance $d(\cdot, \cdot)$, e.g., the Manhattan distance, by applying a weighting factor δ_i that depends on the vector w_i to which the distance is computed:

$$\text{CDM}(w_q, w_i) = d(w_q, w_i)\, \delta_i. \quad (2)$$

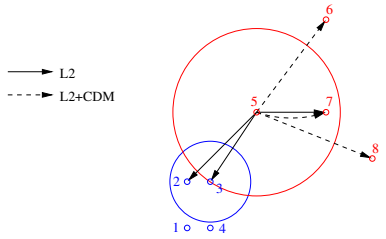

Figure 1. Toy example: the 3-nearest neighbors of vector 5 without and with CDM. The circles depict the average distances of the vectors 3 and 5 to their neighborhood.

The distance update term δ_i is computed off-line for each visual word vector of the database. The extra-storage required to store this scalar is negligible.

Efficient search: The distance computation is optimized with an inverted file system exploiting the sparsity of the visual word vectors [17]. Such an inverted file can be used for any Minkowski norm [12] when the vectors are of unit norm. For huge vocabulary sizes, the hierarchical clustering proposed in [12] greatly reduces the cost of assigning the descriptors to visual words.

Note, however, that the visual word vector search complexity remains linear with the database size. This is the critical step for huge databases, as the steps extraction and assignment of the SIFT descriptors do not depend on the database size. The clustering approach proposed in Section 4 reduces this complexity, hence decreasing the search time by an order of magnitude.

3. Contextual dissimilarity measure

Let us consider Fig. 1. On this toy example vector 3 is a 3-nearest neighbor of vector 5, but the converse is not true. This observation underlines the fact that the neighborhood relationship is not symmetric in a k-nearest neighbor framework. By contrast, it is the case in an ε-search framework.

The dissimilarity measure described in this section improves the symmetry of the k-neighborhood relationship by updating the distance, such that the average distance of a vector to its neighborhood is almost constant. This regularization is performed in the spirit of a local Mahalanobis distance for each vector. Indeed, assuming all the directions to be equivalent, the average distance computed on the neighborhood can be thought of as a local variance computed for each vector. Furthermore, assuming a Bayesian framework, the distance to a vector can be thought of as a likelihood. In order to push further the similarity, one would have to give an interpretation to the iterative CDM construction proposed in 3.2.

Let us consider the neighborhood $\mathcal{N}(i)$ of a given visual word vector w_i and $\#\mathcal{N}(i)$ the cardinal of this set (which is

a constant within the k-nearest neighbors framework). The quantity defined hereafter, and referred to as the *neighborhood symmetry rate*, is an objective measure of the notion of neighborhood symmetry:

$$s = \frac{1}{n} \sum_{w_i} \frac{1}{\#\mathcal{N}(i)} \sum_{w_j \in \mathcal{N}(i)} \text{sym}(w_i, w_j), \qquad (3)$$

where the $sym(w_i, w_j) = 1$ if w_i is a neighbor of w_j and w_j is a neighbor of w_i, 0 otherwise. By definition, the symmetry rate is maximized in the ε-search framework, due to the distance symmetry property. Although we believe that such a perfect neighborhood symmetry is not likely to be properly enforced in the framework of k-nearest neighbors search, it can improve.

In the rest of this section, we first introduce the update procedure of the dissimilarity measure. This first step of the procedure, by itself, produces a new dissimilarity measure (non-iterative approach). The proposed CDM is then obtained by iterating this update step until a stopping criterion is satisfied.

3.1. Non-iterative approach

Let us consider the neighborhood $\mathcal{N}(i)$ of a given visual word vector w_i defined by its $\#\mathcal{N}(i) = n_{\mathcal{N}}$ nearest neighbors. We define the neighborhood distance $r(i)$ as the mean distance of a given visual word vector w_i to the vectors of its neighborhood:

$$r(i) = \frac{1}{n_{\mathcal{N}}} \sum_{x \in \mathcal{N}(i)} d(w_i, x), \qquad (4)$$

where $d(\cdot, \cdot)$ is a distance or dissimilarity measure, e.g. the distance derived from the L1-norm. The quantity $r(i)$ is shown in Fig. 1 by the circle radii. It is computed for each visual word vector and subsequently used to define a first dissimilarity measure $d^*(.,.)$ between two visual word vectors:

$$d^*(i,j) = d(i,j) \left(\frac{\bar{r}^2}{r(i)r(j)} \right)^{\alpha}, \qquad (5)$$

where $0 < \alpha < 1$ is a smoothing factor and $\bar{r}$ is the geometric mean neighborhood distance obtained by

$$\bar{r} = \prod_i r(i)^{\frac{1}{n}}. \qquad (6)$$

This quantity is computed in the log domain. Note that the arithmetic mean can be used as well and leads to similar results. The relation $d^*(\cdot, \cdot)$, referred to as *non-iterative contextual dissimilarity measure* (NICDM), is not a distance: although the symmetry and the separation axioms are satisfied, the triangular inequality does not hold. The nearest neighbors of a given vector w_i can nevertheless be obtained for this relation.

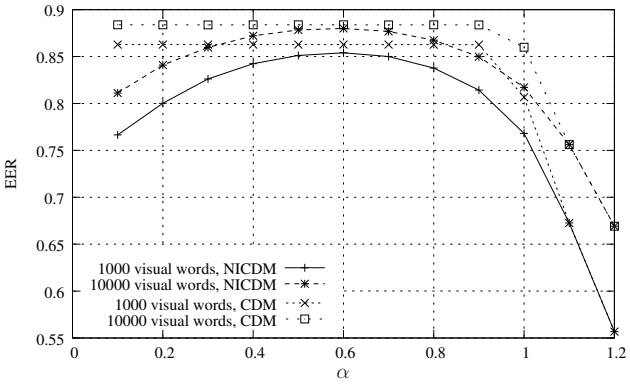

Figure 2. Impact of smoothing factor α on the relevance for the non-iterative (NICDM) and the iterative (CDM) approaches in the case of 1000 and 10000 visual words. N-S dataset [15].

Note that in (5), the terms $r(i)$ and $\bar{r}$ do not impact the nearest neighbors of a given vector. They are used to ensure that the relation is symmetric. The best values for the factor α have been experimentally observed to lie between 0.4 and 0.8, as shown in Fig. 2. The relevance is measure by the equal error rate (EER). Its definition as well as the description of the Nistér-Stewénius dataset [12] (N-S dataset) used here is detailed in the experimental section 5. Note that $\alpha = 0$ amounts to using the original distance $d(\cdot, \cdot)$.

Let us now consider the impact of the approach on the average distance of a given vector w_i to the others. This impact is formalized by the following ratio:

$$\frac{\prod_j d^*(i,j)}{\prod_j d(i,j)} = \left(\prod_j \frac{\bar{r}^2}{r(i)r(j)} \right)^{\alpha}. \qquad (7)$$

Together with the observation that $\prod_j r(j) = \bar{r}^n$, we have

$$\frac{\prod_j d^*(i,j)}{\prod_j d(i,j)} = \left(\frac{\bar{r}}{r(i)} \right)^{\alpha n}, \qquad (8)$$

which in essence means that the NICDM $d^*(\cdot, \cdot)$ favors isolated vectors (with $r(i) > \bar{r}$) and, conversely, penalizes vectors lying in dense areas.

3.2. Iterative approach

The update of Eq. (5) is iterated on the new matrix of dissimilarities. The rationale of this iterative approach is to integrate the neighborhood modification from previous distance updates. Denoting with a superscript $^{(k)}$ the quantities obtained at iteration k, we have

$$d^{(k+1)}(i,j) = d^{(k)}(i,j) \left(\frac{\bar{r}^{(k)}}{r^{(k)}(i)} \frac{\bar{r}^{(k)}}{r^{(k)}(j)} \right)^{\alpha}. \qquad (9)$$

Note that, at each iteration, the new neighborhood distances $r^{(k)}(i)$ are computed for each visual word vector w_i.

The objective of this iterative approach is to minimize a function representing the disparity of the neighborhood distances, in other terms to optimize the homogeneousness of the dissimilarity measures in the neighborhood of a vector. This function, here defined as

$$S^{(k)} = \sum_i |r^{(k)}(i) - \bar{r}^{(k)}|, \tag{10}$$

is clearly positive. Its minimum is zero and satisfied by the trivial fixed-point of Eq. 9 such that

$$\forall i,\ r(i) = \bar{r}. \tag{11}$$

Let us define a small quantity $\varepsilon > 0$. As a stopping criterion, the algorithm terminates when the inequality $S^{(k)} - S^{(k+1)} > \varepsilon$ is not satisfied anymore. This ensures that the algorithm stops within a finite number of steps. In practice, for ε small enough, we observed that this criterion led $r^{(k)}(i)$ to converge towards the fixed-point of Eq. (11).

At this point, we can only compute the CDM between visual word vectors of the database, due to the iterative design of this distance. In order to compute directly the CDM from the original distance, one has to maintain a cumulative distance correcting term $\delta_i^{(k)}$ during iterations, as

$$\delta_i^{(k+1)} = \delta_i^{(k)} \left(\frac{\bar{r}^{(k)}}{r^{(k)}(i)} \right)^{\alpha}. \tag{12}$$

Denoting by δ_i the quantity $\delta_i^{(k-1)}$ when the algorithm terminates, it is easy to show that

$$d^k(i,j) = d(i,j)\ \delta_i\ \delta_j. \tag{13}$$

The k-nearest neighbors of a given query q are then the minima given by

$$NN(q) = \text{k-argmin}_j\ d(q,j)\ \delta_j. \tag{14}$$

Note that finding the nearest neighbors of a query vector w_q does not require the knowledge of the update term associated with w_q, as shown in Eq. 14. That's why we will prefer the asymmetric version of the CDM to the one given in Eq. 13, as

$$\text{CDM}(i,j) = d(i,j)\ \delta_j. \tag{15}$$

By default, the term CDM will be dedicated to the definition of the asymmetric measure $\text{CDM}(\cdot,\cdot)$ of Eq. 15. The advantage of this CDM is that it can be computed for a query vector which is not in the database. One has just to store together with a given database visual word vector w_i the corresponding distance update terms $(\delta_i)_{1 \le i \le n}$, which in terms of storage overhead is clearly negligible.

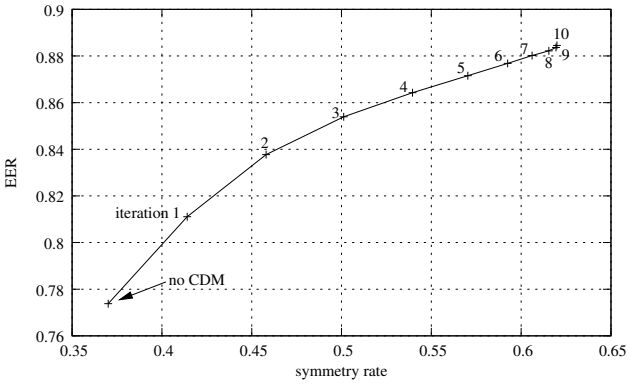

Figure 3. Evolution during iterations of the EER and the symmetry rate (N-S dataset, neighborhood size=10, $\alpha = 0.1$).

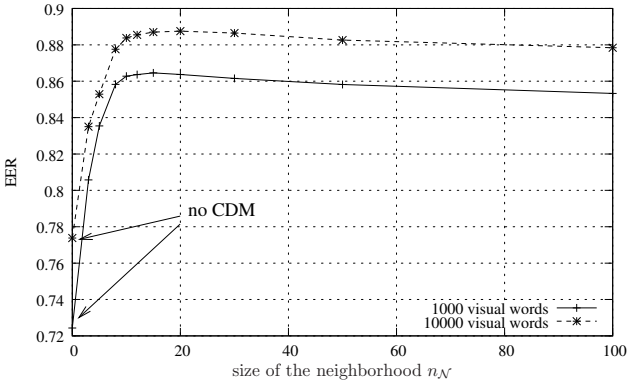

Figure 4. CDM: impact of the neighborhood size. N-S dataset.

3.3. Discussion

Symmetry rate: Fig. 3 shows the impact during iterations of the distance update on the symmetry rate and the relevance measured by the EER. As expected, the symmetry rate increases jointly with the relevance. The factor α has been set to 0.1 for illustration purpose, but note that by setting α to 0.5 the convergence is faster for identical results.

Impact of the parameters: Two parameters have to be set: the neighborhood size $n_{\mathcal{N}}$ and the smoothing factor α.

Fig. 2 shows the impact of the smoothing factor α on the performance for both the direct and the iterative approaches. For the former, the best results are obtained for a value of α lying between 0.4 and 0.8, with a maximum for $\alpha = 0.6$ approaching the performance of CDM. It also appears that the EER behavior of the CDM is remarkably stable when $\alpha < 0.8$. This is the main advantage of the iterative approach over the direct approach, as in practice the algorithm converges towards a set of distance correcting terms that do not depend on α. We have fixed $\alpha = 0.5$ for all the experiments in the following.

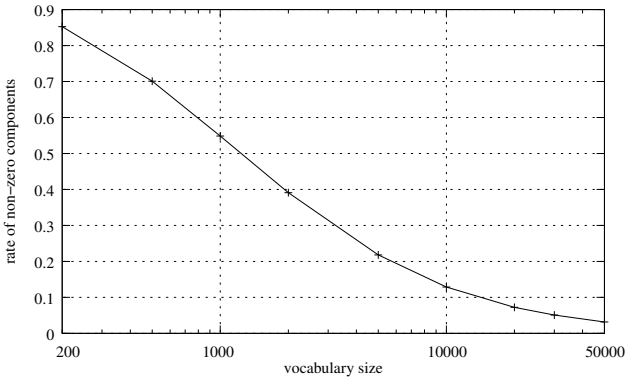

Figure 5. Vector sparsity *vs* vocabulary size. N-S dataset.

Fig. 4 depicts the impact of the neighborhood size on the performance of the iterative approach. Once again, the sensitivity to this parameter is moderate. In the rest of this paper, the size $n_{\mathcal{N}}$ has been fixed to 10, although better results may be obtained by optimizing this quantity.

4. Efficient search for visual word vectors

Although for a database of several thousands of images, finding the closest visual word vectors is quite fast (about 0.15s for a database of 10200 images and 10000 visual words), its search complexity is linear in the database size and is the critical stage for huge databases.

In order to reduce the complexity, we propose a two-level structure for efficient visual word search. The first level consists in an inverted file of medoids. The second level is composed of the set of clusters, each of which being searched by an inverted file, as illustrated by Fig. 6.

4.1. Choice of the medoids

Two strategies are proposed to choose the medoids used in the first level of the inverted file. The first is based on a k-medoids algorithm [7] clustering. The second simply amounts to randomly extracting a subset of visual words.

In the context of visual word vectors clustering, the k-medoids has several advantages over a simple k-means. Firstly, by choosing representative visual word vectors as centroids, the algorithm allows the exploitation of the intrinsic sparsity of visual word vectors. Indeed, the computation of the distance between two sparse vectors (or only one sparse vector) is proportional to the number of non-zeros components. This is especially useful for big visual vocabularies for which the frequency vector sparsity is high, as depicted in Fig. 5. Indeed, searching the nearest neighbors of a given visual word vector query is at least 10 ten times faster for 30000 visual words with a basic implementation and even more if an efficient inverted file structure is used.

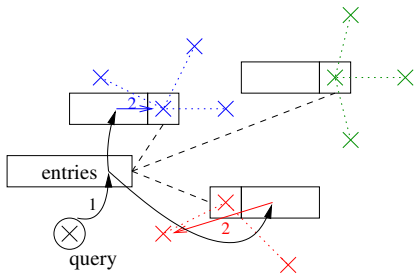

Figure 6. Illustration of our efficient search structure: 1) the inverted file associated with entries is searched to find the best k' (here 2) clusters, 2) the k'' nearest neighbors (here 2) are retrieved using the inverted files associated with these k' clusters.

By contrast, the k-means centroids, for which the centroids are the means of a great number of vectors, have a rate of non-zero components greater than 0.5. Therefore this algorithm can not exploit the sparsity of the visual word vectors.

Secondly and still unlike the k-means algorithm, the k-medoids clustering does not implicitly assume that the distance is Euclidean. It only needs the matrix of distances between visual word vectors, hence allowing the use of the Manhattan distance.

The bottleneck of the k-medoids algorithm is the preliminary computation of the matrix of distances between image frequency vectors. For very large datasets, we extract random visual frequency vectors which serve as medoids. This results in a moderate loss of accuracy.

4.2. Structure construction and querying

Fig. 6 illustrates our clustering-based efficient search structure, which is constructed as follows:

- the medoids are extracted from the set of visual word vectors, producing a set of k medoids called *entries*;
- an inverted file is created for the entries and for each of the k clusters.

For a given query w_q, the search procedure illustrated in Fig. 6 then amounts to performing the following steps:

- use the inverted file of entries to find the k' nearest medoids of the query and the corresponding clusters ;
- for each of the k' corresponding clusters, compute the distances using the inverted files, then return the list of nearest neighbors.

The CDM is exploited at both stages of the search procedure by simply applying the distance update factors δ_i to the distances computed by the inverted files.

Assuming that the clusters are balanced and that the computational cost of searching within an inverted file is roughly linear in the number of elements stored, the complexity of the search is in $\mathcal{O}\left(k + n\frac{k'}{k}\right)$, where the integers

k, k' and n respectively denote the number of medoids, the number of clusters parsed and the total number of vectors. Assuming in addition that the number of clusters k is small in comparison with the size n and that the clusters contain the same number of vectors, then the search cost is approximately divided by k/k'.

4.3. Approximate NICDM/CDM

For very large datasets, the bottleneck of the CDM is the computation of the distances between all the frequency vectors, which is of quadratic complexity with the number of vectors. In what follows we propose to use the efficient search structure to compute the update terms.

For this purpose, we first construct the efficient search structure by choosing as medoids random visual frequency vectors extracted from the dataset. Then, the neighborhood distance of Eq. 4 is computed using the visual word vectors associated with a limited number of clusters (e.g., 5%), which subsequently allows the computation of approximated update terms.

5. Experiments

5.1. Datasets and evaluation criteria

The evaluation is performed on two datasets, namely the N-S dataset [15] and a set of frames extracted from the Lola movie [14]. The first one is composed of 2550 objects or scenes, each of which being taken from 4 different viewpoints. Hence the dataset contains 10200 images. The Lola dataset is composed of 164 video frames taken at 19 different locations in the movie. A different dataset has been used to perform the clustering on uncorrelated data. For this purpose we have taken a subsample of SIFT descriptors extracted from the Corel image database.

Three different measures have been used to evaluate the impact of the various parameters and variants: the EER, the average normalized rank (ANR) and the measure used by Stewénius and Nistér [15]. The EER is the point on the precision/recall curve such that $precision = recall$. It is obtained when the number of images retrieved is equal to the number of relevant images. The ANR is given by

$$\text{ANR} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{n\,n_{\text{rel}(i)}}\sum_{j=1}^{n_{\text{rel}(i)}}\text{rank}(j) - \frac{n_{\text{rel}(i)}(n_{\text{rel}(i)}+1)}{2}, \tag{16}$$

where n is the number of dataset images and $n_{\text{rel}}(i)$ is the number of images which should be retrieved for the image i.

For the sake of comparison, we will also use the Nistér score. This simple measure [15] counts the average number of correct images among the four first images returned for a given query. This measure is meaningful because there are 4 relevant images per object in the N-S dataset. Note that, for this dataset, this measure is equal to four times the EER.

5.2. Evaluation of the parameters

Table 1 and Table 2 summarize the impact of the parameters on the retrieval accuracy for the two datasets respectively. The analysis focuses on the following parameters: CDM, the SIFT clustering algorithm, the visual vocabulary size, the norm and the use of the PCA and of MA.

CDM: All the experiments in Table 1 and Table 2 show a significant improvement when using CDM. Fig. 7 illustrates some typical queries for which the CDM significantly improves the results. For the N-S dataset (first two lines), the query with no CDM returns flowers, which are often irrelevantly returned. The capability of the CDM to reduce the impact of the too-often-selected images is clear in this context. The query on the Lola database (two last lines) is even more impressive. The first three images are correct with and without CDM. Although the four next images seem wrong for both queries, they are in fact correct for the CDM, as the images correspond to the same location (the Deutsche Transfer Bank) observed from significantly different viewpoints.

Clustering: We have implemented our own version of the hierarchical clustering according to [12]. Although Table 1 shows (Exp. #1 and #3) that the accuracy is somewhat reduced, this approach is very efficient and greatly reduces the computing cost associated with the assignment of SIFT descriptors to visual words when the vocabulary size is big. Note however that SIFT assignment is not the most critical stage for efficiency.

The choice of the learning set is also shown to have a strong impact on the accuracy. By default we have used the uncorrelated Corel dataset. However, the results are significantly improved by using instead a subsample of the dataset on which the experiments are performed, as shown in Table 2 by Exp. #4 and #9.

Vocabulary size: On the experiments #2 #3 #4 and #5 of Table. 1 and the experiments #1 #3 #6 and #7 of Table. 2, one can see that bigger vocabularies provide better retrieval accuracy. However, for the N-S dataset (See Table 1), the gain is rather low when using vocabulary sizes greater than 10000.

Norm: It was observed in [12] that the Manhattan distance provides better results than the Euclidean one. This observation is confirmed in our experiments for the two databases and is also true when the CDM is used.

Relevance of the variants: In Table 1, Exp. #5 and Exp. #7 show that the PCA marginally reduces the accuracy of the scheme, while decreasing the computational cost associated with the visual word assignment. However, the impact on the efficiency of this dimensionality reduction is confined to the word frequency processing stage. Since the hierarchi-

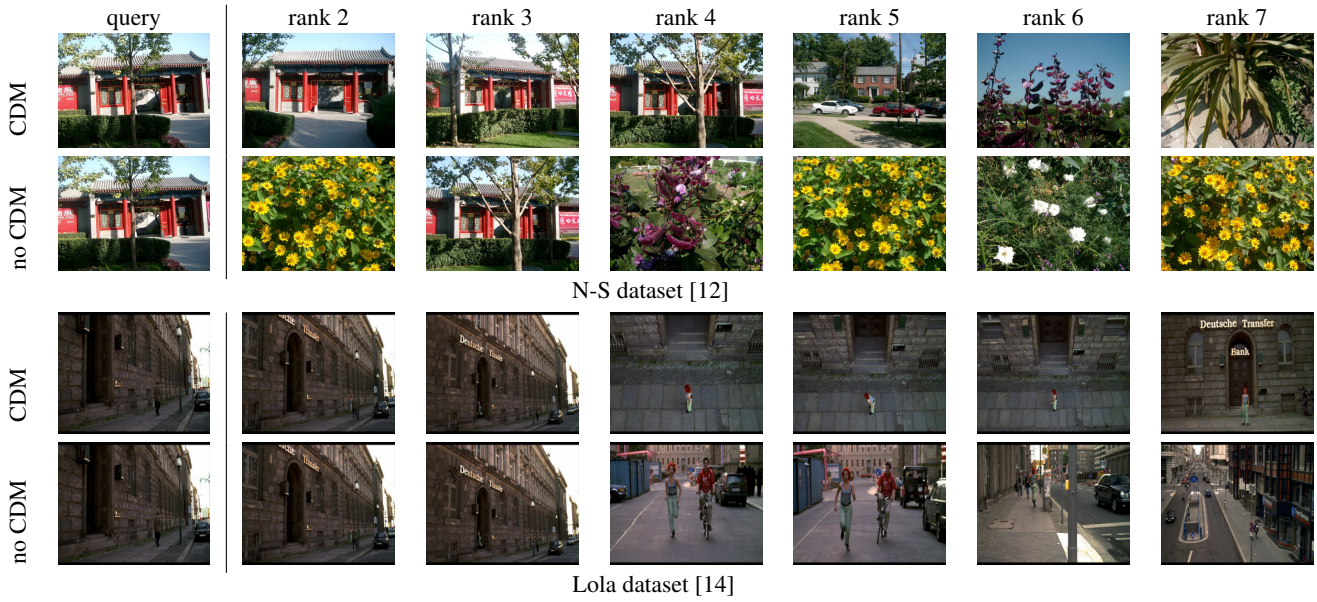

Figure 7. Query examples: short lists returned for a given query with and without the CDM.

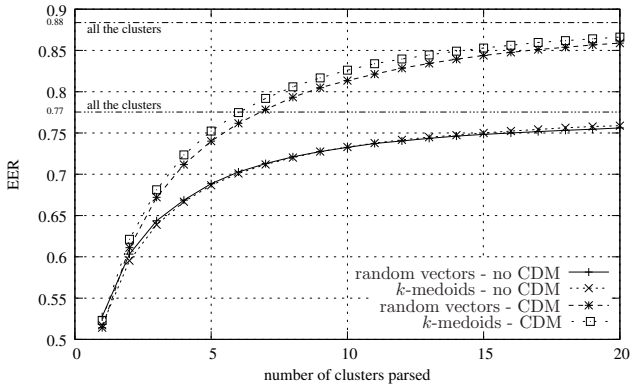

Figure 8. Efficient search structure: trade-off between the number of image clusters visited and the EER. The total number of clusters is 200.

	SIFT clustering	vocab. size	EER no CDM	EER CDM	
#1	hierarchical	10000	0.724	0.853	
#2	k-means	1000	0.724	0.862	
#3	k-means	10000	0.774	0.884	
#4	k-means	20000	0.781	0.885	
#5	k-means	30000	**0.784**	0.887	
#6	k-means	30000	0.669	0.842	L2
#7	k-means	30000	0.753	0.873	PCA
#8	k-means	30000	0.780	0.898	MA×2
#9	k-means	30000	0.767	**0.899**	MA×3

Table 1. Nistér and Stewénius dataset. Impact of the CDM ($n_{\mathcal{N}} = 10$) and of the following parameters: clustering algorithm (k-means or hierarchical [12]), vocabulary size, norm (L1 if not specified or L2), use of the PCA (36 dimensions), multiple assignment (MA) of descriptors to visual words.

cal SIFT assignment of [12] improves the efficiency much more, the interest of the PCA in this context is limited.

The MA of SIFT descriptors to visual words somewhat improves the accuracy of the search (compare Exp. #8 and Exp. #9 to Exp. #5 in Table 1) at the cost of an increased search time, due to the impact of the method on the visual word vector sparsity. It should be used for applications requiring high accuracy. Note that the number of assignments must be small, e.g. 2 or 3, as we have observed that the accuracy decreases for larger values.

Efficient medoids-based search structure: Fig. 8 shows how the method introduced in Section 4 trades accuracy against efficiency. The maximum score is almost attained for 20 clusters, which corresponds to 10% of the 200 clusters. In this case, the search time is approximately reduced by a factor 7. These results have been obtained using either the k-medoids algorithm or a random subset for the first level of the search structure. Using random visual word vectors instead of medoids, on can observe a moderate loss of accuracy and efficiency (the clusters are less balanced).

5.3. Comparison with the state-of-the-art

For the N-S dataset, our approach obtains a Nistér score of 3.60 (maximum 4) for a CDM computed with $n_{\mathcal{N}} = 10$ neighbors and 30000 visual words. The best score presented [15] is 3.19 for the most time consuming approach.

	training set	vocab. size	norm	$n_{\mathcal{N}}$	ANR no CDM	ANR CDM
#1	corel	10000	L1	30	0.0522	0.0148
#2	corel	20000	L1	10	0.0476	0.0238
#3	corel	20000	L1	20	0.0476	0.0156
#4	corel	20000	L1	30	0.0476	0.0145
#5	corel	20000	L2	30	0.0528	0.0224
#6	corel	30000	L1	30	0.0468	0.0133
#7	corel	50000	L1	30	0.0416	**0.0118**
#8	lola	10000	L1	30	0.0321	0.0063
#9	lola	20000	L1	30	0.0240	**0.0046**

Table 2. Lola dataset. Impact of the vocabulary size, the norm (Manhattan L1 or Euclidean L2) and the number of neighbors $n_{\mathcal{N}}$ used in the CDM calculation.

The visual vocabulary have respectively been learned on the Corel dataset in our case, and on the Flip dataset in [15].

The best ANR obtained for the Lola movie is 0.0046, significantly outperforming the best score 0.0132 of [14]. Note that, by contrast to this work, we only use one kind of descriptor (in that case the best score of [14] is 0.0196) and no temporal filtering. Our approach is still better (0.0118) if the visual words are learned on uncorrelated data.

5.4. Large-scale evaluation

To assess the scalability of our approaches, the CDM and the efficient search structure have been used jointly for large scale image search. For this purpose, we have merged the N-S dataset with a set of images downloaded from the web. The images producing less that 10 interest points have been removed. To reduce the overall computing cost, we have used a maximum of 1000 descriptors per image, chosen according to their cornerness. We have not used the *tf-idf* scheme and the number of visual words (learned on a different dataset) has been set to 10000. For the search structure, we have randomly extracted 400 medoids from the dataset. We have then opted for the NICDM with $\alpha = 0.6$. The approximate NICDM update factors have been obtained according to the guidelines of subsection 4.3.

Table 3 shows that the NICDM improves the results for any dataset size. Querying with 40 image clusters out of 400 does not significantly alter the search accuracy. Interestingly, using a subset of clusters to compute the CDM does not significantly impact the accuracy of it. Hence, for the largest dataset, as the clusters contain many images, we only used 3 of them.

6. Acknowledgements

We would like to acknowledge J. Sivic, A. Zisserman, D. Nistér and H. Stewénius for kindly providing their datasets. Hedi Harzallah was funded by the INRIA student exchange program.

dataset size	number of clusters δ_i	number of clusters query	EER no NICDM	EER NICDM
10200	20	20	0.715	0.771
10200	20	40	0.747	0.815
10200	20	400 (all)	0.772	0.849
10200	400 (all)	400 (all)	0.772	0.851
100000	20	20	0.678	0.736
100000	20	40	0.707	0.772
200000	10	20	0.700	0.727
200000	10	40	0.697	0.762
500000	3	20	0.656	0.712
500000	3	40	0.682	0.745
1000000	3	20	0.644	0.701
1000000	3	40	0.669	0.732

Table 3. Large-scale evaluation with and without NICDM. Impact of the dataset size and the number of image clusters used (a) to compute the distance update terms δ_i and (b) for querying. Parameters: 10000 distinct visual words, $\alpha = 0.6$, 400 medoids.

References

[1] A. Andoni and P. Indyk. Near-optimal hashing algorithms for near neighbor problem in high dimensions. In *Proc. Symp. Foundations Computer Science*, pages 459–468, 2006.

[2] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, pages 539–546, 2005.

[3] R. Fagin, R. Kumar, and D. Sivakumar. Efficient similarity search and classification via rank aggregation. In *ACM SIGMOD Conf.*, pages 301–312, 2003.

[4] A. Frome, Y. Singer, and J. Malik. Image retrieval and classification using local distance functions. In *NIPS*, 2006.

[5] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimension via hashing. In *Proc. Intl. Conf. Very Large DataBases*, pages 518–529, 1999.

[6] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. In *NIPS*, 2004.

[7] L. Kaufman and P. Rousseeuw. *Finding groups in data: an introduction to cluster analysis*. J. Wiley & Sons, 1990.

[8] Y. Ke and R. Sukthankar. PCA-SIFT: a more distinctive representation for local image descriptors. In *CVPR*, pages 506–513, 2004.

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

[10] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *IJCV*, 60(1):63–86, 2004.

[11] F. Moosmann, B. Triggs, and F. Jurie. Randomized clustering forests for building fast and discriminative visual vocabularies. In *NIPS*, 2006.

[12] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *CVPR*, pages 2161–2168, Jun 2006.

[13] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.

[14] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, pages 1470–1477, Oct. 2003.

[15] H. Stewénius and D. Nistér. Object recognition benchmark. http://vis.uky.edu/%7Estewe/ukbench/.

[16] K. Weinberger, J. Blitzer, and L. Saul. Distance metric learning for large margin nearest neighbor classification. In *NIPS*, 2006.

[17] J. Zobel, A. Moffat, and K. Ramamohanarao. Inverted files versus signature files for text indexing. *ACM Transactions on Database Systems*, 23(4):453–490, 1998.