

Accurate image search using the contextual dissimilarity measure

Hervé Jégou, Cordelia Schmid, Hedi Harzallah, Jakob Verbeek

▶ **To cite this version:**

Hervé Jégou, Cordelia Schmid, Hedi Harzallah, Jakob Verbeek. Accurate image search using the contextual dissimilarity measure. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010, 32 (1), pp.2-11. 10.1109/TPAMI.2008.285 . inria-00439311v3

HAL Id: inria-00439311

https://inria.hal.science/inria-00439311v3

Submitted on 15 Mar 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Accurate image search using the contextual dissimilarity measure

Hervé Jégou, Cordelia Schmid, Hedi Harzallah and Jakob Verbeek

Pre-print version of November 10th 2008, updated March 15th 2011

March 15, 2011

Abstract

This paper introduces the *contextual dissimilarity measure* which significantly improves the accuracy of bag-of-features based image search. Our measure takes into account the local distribution of the vectors and iteratively estimates distance update terms in the spirit of Sinkhorn's scaling algorithm, thereby modifying the neighborhood structure. Experimental results show that our approach gives significantly better results than a standard distance and outperforms the state-of-the-art in terms of accuracy on the Nistér-Stewénius and Lola datasets.

This paper also evaluates the impact of a large number of parameters, including the number of descriptors, the clustering method, the visual vocabulary size and the distance measure. The optimal parameter choice is shown to be quite context-dependent. In particular using a large number of descriptors is interesting only when using our dissimilarity measure. We have also evaluated two novel variants, multiple assignment and rank aggregation. They are shown to further improve accuracy, at the cost of higher memory usage and lower efficiency.

1 Introduction

In this paper we address the problem of finding images of the same object or scene viewed under different imaging conditions. Initial approaches used simple voting based techniques [1, 2]. More recently they were extended based on a bag-of-features image representation [3, 4]. Our paper builds upon these approaches and presents methods to improve the accuracy.

The main contribution of this paper is the contextual dissimilarity measure (CDM) which takes into account the neighborhood of a vector. This measure is obtained by iteratively regularizing the average distance of each vector to its neighborhood. This regularization is motivated by the observation that a

"good ranking" is usually not symmetrical in an image search system. To be more precise, if an image i is well-ranked for a query j, then j is not necessarily well-ranked for query i. Intuitively, this phenomenon yields suboptimal accuracy, as will be confirmed in this paper.

The dissimilarity measure described in this paper 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 the Sinkhorn's scaling algorithm [5]. It is also somewhat similar to a local Mahalanobis distance. Indeed, assuming all directions to be equivalent, the average distance computed over the neighborhood can be seen as a local variance.

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 [6, 7, 8, 9]. In contrast to category classification where class members are clearly defined and represented by a sufficiently large set, this does in general not hold for an image search system. Our approach is somewhat similar to the weighting schemes from text retrieval, e.g. the term frequency/inverse document frequency weighting [10], which can be seen as a simple way to improve the distance [3, 4]. Experimental results show that the gain due to our CDM is significantly higher than the one obtained by a weighting scheme. Furthermore, these approaches can be combined.

This paper also analyzes the impact of different parameters. We show that using a large number of descriptors is not always desirable, except when using the CDM. As previously shown in [11], the dataset used to generate the visual vocabulary strongly impacts the accuracy of the search. Accuracy is much higher when the vocabulary is learnt on the dataset to search, especially when using large visual vocabularies.

The first proposed variant consists in assigning each local descriptor to several visual words instead of only one. It provides a moderate but consistent improvement, especially for large visual vocabularies. As it significantly impacts the efficiency of the query, it should only be considered when very high accuracy is required. The second proposed variant is rank aggregation [12], which has not been used in the context of bag-of-features based image search before. The idea is to combine the results of several concurrent image search systems which differ in the visual vocabularies learnt on distinct subsamples of descriptors. The performance improves significantly, and increases with the number of image search systems used in parallel. However, using several image search systems has a high cost in terms of memory and computation time. We have found experimentally that the choice of three systems is a good compromise, as it provides most of the accuracy improvement.

This paper is organized as follows. Section 2 reviews the bag-of-words image retrieval approach of [4] and describes some variants. The contextual dissimilarity measure and its relationship with Sinkhorn's algorithm are de-

scribed in Section 3. Section 4 presents the approach for rank aggregation. The relevance of our approach and its parameters are then analyzed 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 [4], and the tested variations.

Descriptors: The n database images are described with local descriptors. We combine the SIFT descriptor [1] with the affine Hessian region extractor [2]. As a variant, the 128-dimensional SIFT descriptors are reduced to 36-dimensional vectors using principal component analysis.

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). As a variant, we use the fast hierarchical clustering approach described in [3].

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 L_1 norm, generating a frequency vector $f_i = (f_{i,1}, \ldots, f_{i,V})$. 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 components of the frequency vector are weighted with a strategy similar to the one in [3]. 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 weighted component $w_{i,j}$ associated with image i is given by

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

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

Distance: Given the visual word vector w_q of a query, similar images in the database are represented by 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 two weighting factors δ_i and δ_j that depends on the vectors w_i and w_j

between which the distance is computed:

$$\text{CDM}(w_i, w_j) = d(w_i, w_j)\ \delta_i\ \delta_j. \tag{2}$$

The distance update term is computed off-line for each visual word vector of the database. The extra-storage required to store this scalar is negligible. We will show in Section 3 that computing this term is not required for the query vector.

Efficient search: The distance computation is optimized with an inverted file system exploiting the sparsity of the visual word vectors [13]. Such an inverted file can be used for any Minkowski norm [3] when the vectors are of unit norm. For huge vocabulary sizes, the strategies proposed in [3] and [14] greatly reduce the cost of assigning the descriptors to visual words.

Rank aggregation: The idea is to use multiple visual vocabularies, i.e., to combine the results of several image search systems where each one uses a different vocabulary. Each vocabulary is learnt on a different subset of the descriptors. The approach is described in Section 4.

3 Contextual dissimilarity measure

In this section, we first motivate our new measure. We then introduce the update procedure of the dissimilarity measure. This first step of this 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. Finally, we underline the relationship between our approach and the projection to doubly-stochastic matrices. We also show how to efficiently compute the CDM for large datasets.

3.1 Neighborhood non-reversibility and its impact

The toy example of Fig. 1 illustrates the non-reversibility of the neighborhood for a k-nearest neighbor search. Vector 3 is a 3-nearest neighbor of vector 5, but the converse is not true. In contrast, it is (trivially) the case for an ε-search, where the neighborhood of a vector q is defined as the set of vectors x such that $d(q, x) < \varepsilon$.

To measure the reversibility of a neighborhood we introduce the *neighborhood reversibility rate*. Let us consider the neighborhood $\mathcal{N}(i)$ of a given visual word vector w_i and $\#\mathcal{N}(i)$ the cardinality of this set. Obviously, $\#\mathcal{N}(i) = k$ is constant within the k-nearest neighbors framework. The *neighborhood reversibility rate* is then defined as follows:

$$\frac{1}{n} \sum_{w_i} \frac{1}{\#\mathcal{N}(i)} \sum_{w_j \in \mathcal{N}(i)} \text{revers}(w_i, w_j), \tag{3}$$

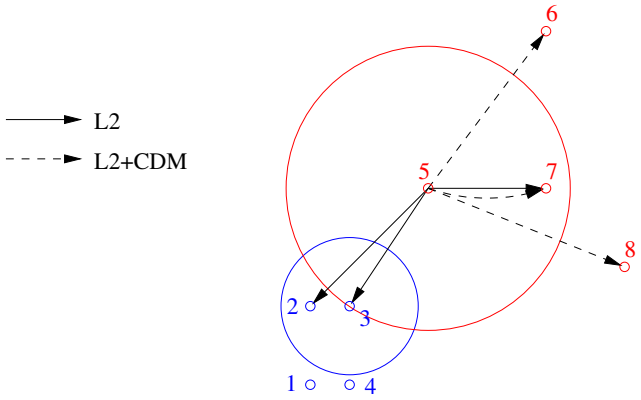


Figure 1: Toy example: the 3-nearest neighbors of vector 5 without (solid) and with CDM (dashed). The circles display the average distances of vector 3 (blue) and 5 (red) to their neighborhood.

where $revers(w_i, w_j) = 1$ if w_i is a neighbor of w_j and w_j is a neighbor of w_i, 0 otherwise.

The neighborhood properties of a bag-of-feature image search system are illustrated in Fig. 2. The first line shows the returned images for the query image on the left. We can observe that the three relevant images are not ranked in the first three positions. However, if we submit each of the 10 highest ranked images to the system, we can observe that the initial query is returned in the 10-neighborhood of the relevant images only, see[1] columns of Fig. 2. In other words: the neighborhood reversibility is satisfied for the relevant images only. This suggests that accuracy may be improved by

- verifying for each returned image of the short-list, if the reversibility property is satisfied. However, note that this would require to perform many additional queries, i.e. one per image of the short-list[2].
- by improving the reversibility of the neighborhood.

Due to the non-reversibility of the neighborhood, we can observe that some images are returned relatively often, while others are returned only when submitting the image itself. These images are referred to as *too-often-selected images* and *never seen images* and are defined for a given neighborhood size k. Section 5.2.3 shows experimentally that the CDM significantly reduces the number of too-often-selected and never seen images. Note that the never seen images

[1]Note that the reversibility rate for this query and a neighborhood of size 10 is equal to 0.3.

[2]We have tested this variant and observed that its performance is inferior to the distance regularization proposed in this paper.

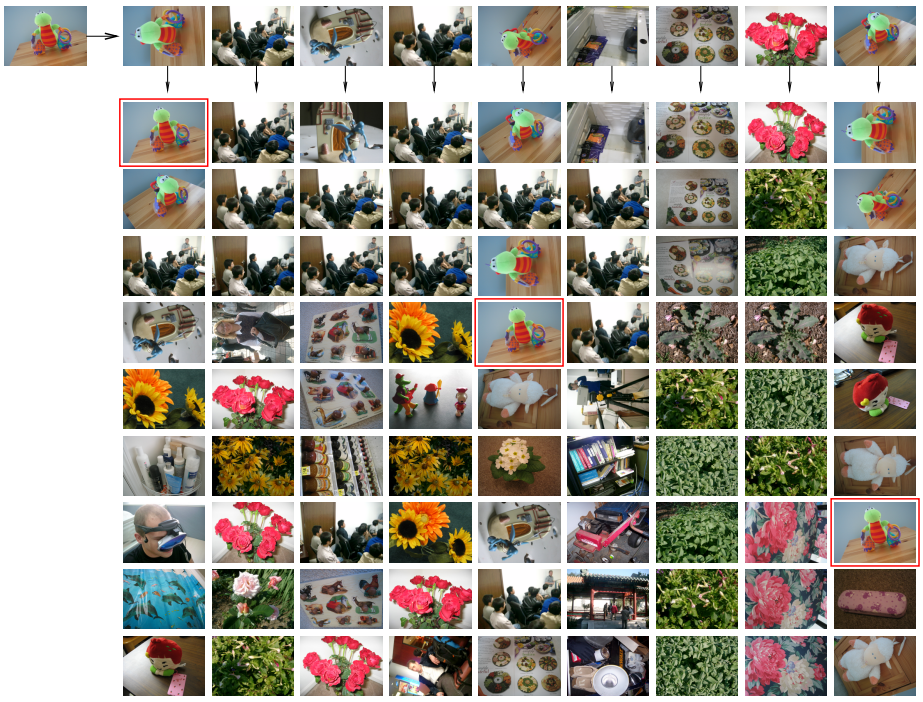


Figure 2: Illustration of the neighborhood non-reversibility of a bag-of-features image search system (L_1 distance). The top-left image is the query image. The first line displays the ordered set of images returned for this query. The three relevant images are ranked 1^{st}, 5^{th} and 9^{th}. Each column represents a new image search performed by querying with the top image of this column. The boxes indicates where the initial query (top-left image) is returned for the new queries.

can not be retrieved even when iteratively browsing the dataset, i.e., when choosing any image in the short-list of size k as new query.

3.2 Non-iterative approach

The above mentioned problems of neighborhood non-reversibility suggest a solution which regularizes the visual word vector space. Intuitively, we would like the k-neighborhoods to have similar diameters in order to approach the reversible ε-neighborhood.

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 vec-

tors of its neighborhood:

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

where $d(\cdot, \cdot)$ is a distance or dissimilarity measure, e.g. the distance derived from the L_1-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:

$$d^*(w_i, w_j) = d(w_i, w_j) \frac{\bar{r}}{\sqrt{r(i)r(j)}}, \tag{5}$$

where $\bar{r}$ is the geometric mean neighborhood distance obtained by

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

This quantity is computed in the log domain. Note that the arithmetic mean can be used as well and leads to similar results. In contrast to [15], we do not use any smoothing factor denoted α in [15]. Indeed, the new update term $\bar{r}/\sqrt{r(i)r(j)}$ used here (instead of its square) amounts to choosing $\alpha = 0.5$. For this non-iterative approach, it provides close to optimum results in terms of accuracy, see [15].

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. This is not a problem in the context of finding the nearest neighbors of a given vector w_i. Comparison measures that do no satisfy the properties of a distance have been used in information retrieval. For instance, the image search system of [16] explores the use of the Shannon-Jenson divergence and a metric derived from a LDA model.

Note that in Eq. 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.

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:

$$\frac{\prod_j d^*(w_i, w_j)}{\prod_j d(w_i, w_j)} = \prod_j \frac{\bar{r}}{\sqrt{r(i)r(j)}}. \tag{7}$$

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

$$\frac{\prod_j d^*(w_i, w_j)}{\prod_j d(w_i, w_j)} = \left(\sqrt{\frac{\bar{r}}{r(i)}} \right)^n, \tag{8}$$

which in essence means that, on average, the NICDM $d^*(\cdot, \cdot)$ reduces distances associated with isolated vectors (with $r(i) > \bar{r}$) and, conversely, increases the ones of vectors lying in dense areas.

3.3 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

$$d^{(k+1)}(w_i, w_j) = d^{(k)}(w_i, w_j) \frac{\bar{r}^{(k)}}{\sqrt{r^{(k)}(i) r^{(k)}(j)}}. \tag{9}$$

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.

Let us recall that, by contrast to [15] where a smoothing factor had to be set, here Eq. 9 amounts to choosing $\alpha = 0.5$. This choice does not impact the accuracy, as it is observed that for $\alpha < 0.9$, the algorithm converges towards the same set of values.

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

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

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

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

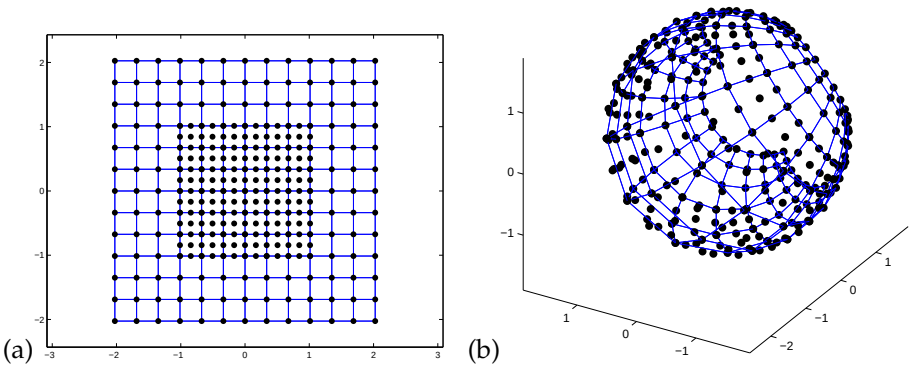


Figure 3: (a) A set of points in $\mathbb{R}^2$ with zones of different density. (b) Projection onto the sphere in $\mathbb{R}^3$ using Sinkhorn's algorithm to yield doubly stochastic distance matrix. Note the much more uniform density of the points on the sphere.

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

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

Note that finding the nearest neighbors of a query vector q does not require the knowledge of the update term associated with q, as shown in Eq. 14. That is why we only need to compute the partial term

$$d(q, w_j) \; \delta_j. \tag{15}$$

Hence, it is possible to find the nearest neighbors with the CDM for a 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 term δ_i, which in terms of storage overhead is clearly negligible. Given the original distance matrix and the parameters k (neighborhood size) and ε (convergence threshold), the pseudo-code for computing the update terms of the CDM is given by Algorithm 25. This algorithm may be advantageously implemented in the log-domain.

3.4 Relationship between CDM and projection to doubly-stochastic matrices

Here we briefly describe the projection of distance matrices to doubly-stochastic matrices, which is closely related to the CDM introduced above. As discussed

Algorithm 1 – `Compute_CDM_update_terms`($\mathbf{D}$,k,ε)

Input $\mathbf{D}$: $N \times N$ matrix of pairwise vector distances
Input k: neighborhood size
Input ε: convergence threshold

$\boldsymbol{\delta} = (\delta_1,\ \delta_2, \ldots, \delta_N) := (1,\ 1, \ldots 1)$
$\mathbf{D_0} = \mathbf{D}$
repeat
  % compute neighborhood average distances
  for $i = 1$ to N **do**
    $\mathcal{N}(i) :=$ { k nearest neighbors of the i^{th} vector }
    $r(i) := \frac{1}{k} \sum_{j \in \mathcal{N}(i)} \mathbf{D}(i,j)$
  end for

  % compute their geometric mean
  $\bar{r} := (\prod_i r(i))^{\frac{1}{n}}$

  % compute update terms
  for $i = 1$ to N **do**
    $\delta_i := \delta_i \sqrt{\frac{\bar{r}}{r(i)}}$
  end for

  % update the pairwise vector dissimilarity measures
  % `diag`($\boldsymbol{\delta}$) is the diagonal matrix with diagonal $\boldsymbol{\delta}$
  $\mathbf{D} :=$ `diag`($\boldsymbol{\delta}$) $\times \mathbf{D_0} \times$ `diag`($\boldsymbol{\delta}$)

  % test convergence
  $S_{\text{old}} := S_{\text{new}}$
  $S_{\text{new}} := \sum_i |r_i - \bar{r}|$
until $S_{\text{old}} - S_{\text{new}} < \varepsilon$
return $(\delta_1, \ldots, \delta_N)$

in the previous section the CDM rescales distances $d(w_q, w_i)$ by a scalar factor δ_i. The correction factors δ_i are set in such as manner that for all points w_i the average distance from w_i to its nearest neighbors, $r(i)$, become similar: $\forall_i :\ r(i) \approx \bar{r}$.

The CDM is a modification of Sinkhorn's scaling algorithm [5]. Sinkhorn's algorithm takes a positive matrix $\mathbf{A}$ and iteratively normalizes the rows and columns to have unit L_1 norm. The algorithm is guaranteed to converge to a unique fixed point, which is a doubly stochastic matrix: all rows and columns sum to unity. Interestingly, when applied to a squared distance matrix $\mathbf{A}$ there is a geometric interpretation of Sinkhorn's algorithm [17]. In this case Sinkhorn's algorithm yields a matrix $\mathbf{B} = \boldsymbol{\Delta}\mathbf{A}\boldsymbol{\Delta}$ which is doubly stochastic, and $\boldsymbol{\Delta}$ is a diagonal matrix with diagonal elements δ_i. If P is the set of points that generated the square distance matrix $\mathbf{A}$, then elements of $\mathbf{B}$ correspond to

the square distances between the corresponding points in a set Q, where the points in Q are obtained by a stereographic projection of the points in P. The points in Q are confined to a hypersphere of dimension d embedded in $\mathbb{R}^{d+1}$, where d is the dimensionality of the subspace spanned by the points in P (or, if the points in P are confined to a hypersphere, d is the dimension of that hypersphere). In Fig. 3 we illustrate the projection of a set of points $P \in \mathbb{R}^2$ to a sphere in $\mathbb{R}^3$.

Note that the points in Q all have the same average squared distance to other points, since all rows (and columns) of $\mathbf{B}$ sum to unity. Thus, if we consider the "new" distances between the i-th point and other points j, given by $[\mathbf{B}]_{ij} = [\mathbf{A}]_{ij}\delta_i\delta_j$, we see that they are given by a scalar correction δ_j of the original distances $[\mathbf{A}]_{ij}$, and in addition we have the scalar correction δ_i which is constant for all j. Clearly, CDM and projection to doubly stochastic matrices modify the distances from a point i to other points j in the same way: by multiplicative correction terms for each j.

The projection to doubly-stochastic matrices suffers from one weakness in the context of image retrieval: the projection takes into account the distances between all pairs of points. However, as high dimensional data—like our visual word frequency vectors—usually live on a (non-linear) manifold of lower dimension embedded in the vector space, only small distances are meaningful in the sense that they tend to correspond to small geodesic distances along the manifold. Large distances, however, are not indicative for the corresponding geodesic distances along the manifold: the geodesic distance may vary greatly for constant distance in the embedding space. For this reason, our CDM method regularizes pairwise distances in smaller neighborhoods instead of regularizing all pairwise square distances. Note that the flavor of the CDM method—global analysis of properties in small overlapping neighborhoods—resembles that used by many recently developed methods for non-linear dimensionality reduction inspired by ISOMAP [18] and LLE [19]. The relevance of this choice is demonstrated in 5.2.1.

3.5 CDM for very large sets

For very large datasets, the bottleneck of the CDM is the distance computation between all frequency vectors, which in theory is of quadratic complexity in the number of images. Fortunately, finding the true neighborhood of frequency vectors is not required to obtain accurate update terms. Suboptimal approximate nearest neighbor search of frequency vectors as proposed in [15] greatly improves the efficiency of the update terms' calculation. We showed that using this strategy to compute the CDM update terms moderately decreases the accuracy of the search while allowing the use of the CDM for a set of 1 million images [15].

Another possible simple method for computing approximate update terms

consists in choosing a fixed set of frequency vectors, not necessarily extracted from the dataset to index, to compute the neighborhood distance. This results in a very fast computation of the update terms. Moreover, the terms do not depend on the dataset to index: they only depend on the chosen fixed set. As a consequence, this avoids having the so-called *out-of-sample extension*, as adding new frequency vectors to the dataset does not modify the other update terms.

4 Exploiting multiple vocabularies using rank aggregation

The search accuracy is improved by using t independently generated vocabularies instead of only one, i.e., by generating t different visual word codebooks. Hence, t distinct image search systems are used, each of which is implemented with an inverted file. The underlying motivation is that it is very unlikely that each system returns the same false positives, while it is very likely that true positives are returned often.

Rank aggregation combines the results of t different subsystems with the method of [12] which was proposed to perform approximate nearest neighbor search of vectors in the spirit of locality sensitive hashing [20]. For each retrieved image we compute its median rank over all ranked lists returned by the t sub-systems. Ties are resolved arbitrarily, but not randomly.

Example: Let us consider $t = 3$ different subsystems and a query for which an image of the dataset is ranked 1^{st} in list 1, 4^{th} in list 2 and 3^{rd} in list 3. The set of ranks obtained for this image is $(1, 3, 4)$, hence its median rank is 3.

Note that this approach can also be applied to other quantiles, i.e., instead of the median one can choose the first quartile. However, we have observed that the median rank consistently provides good results, in contrast to other quantiles.

This approach improves the accuracy, as shown in Section 5. Its main drawback is that the storage requirements are t times higher. Moreover, except for region extraction and descriptor computation, all the other steps of the image search system, i.e., descriptor quantization and inverted file querying are computed t times instead of only one. For very large datasets, where querying the inverted file becomes the bottleneck of the algorithm, the query becomes roughly t times longer.

5 Experiments

5.1 Datasets and evaluation criteria

The evaluation is performed on two datasets:

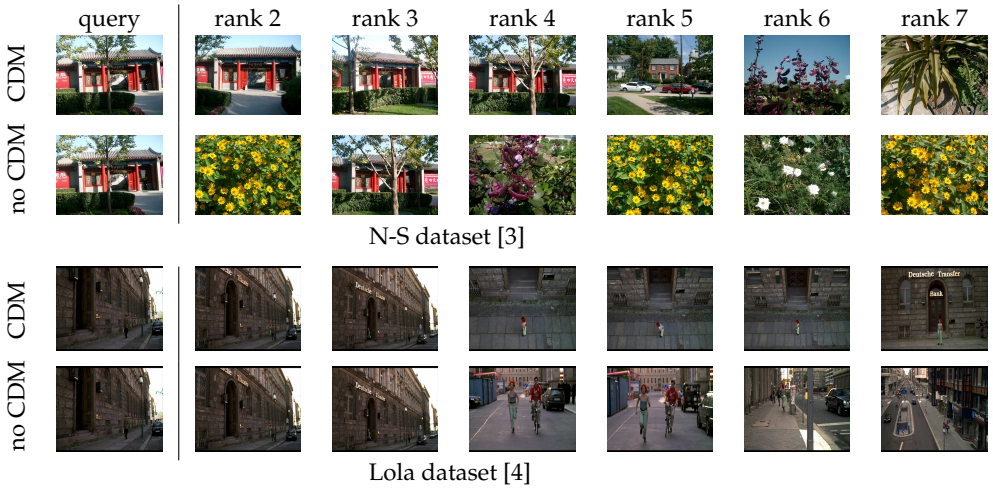


Figure 4: Query examples: ranked lists obtained with and without CDM. We can observe that the CDM significantly improves the ranking.

- the Nistér and Stewénius (N-S) dataset [11],
- a set of frames [4] extracted from the movie "Run, Lola, Run!".

The N-S dataset is composed of 2550 objects or scenes, each of which is imaged from 4 different viewpoints. Hence the dataset contains 10200 images. The Lola dataset is composed of 164 video frames extracted at 19 different locations in the movie.

For all the experiments, we used the Hessian-Affine detector [2]. Except when explicitly specified, the threshold is set to 100, resulting in an average number of 2269 descriptors per image, and the L_1 distance is used.

Three datasets have been used to perform the k-means clustering: the Corel set which is uncorrelated with the evaluation sets, as well as the N-S and Lola datasets used for the evaluation. For this purpose we have extracted from these datasets subsamples of about 1 million SIFT descriptors, except for the Lola dataset where the whole set of descriptors was used.

Two different measures are used to evaluate the impact of the various parameters and variants: the average normalized rank (ANR) and, for the sake of comparison, the measure used by Stewénius and Nistér [11]. For a given query image, the ANR [4] is given by

$$\mathrm{ANR} = \frac{1}{n_q} \sum_{i=1}^{n_q} \frac{1}{n\, n_{\mathrm{rel}(i)}} \left(\sum_{j=1}^{n_{\mathrm{rel}(i)}} \mathrm{rank}(j) - \frac{n_{\mathrm{rel}(i)}(n_{\mathrm{rel}(i)}+1)}{2} \right), \qquad (16)$$

	clustering method	clustering dataset	vocab. size	N-S score no CDM	Sinkhorn	CDM
#1	k-means	corel	1000	2.90	3.05	3.45
#2	k-means	corel	10000	3.10	3.21	3.53
#3	k-means	corel	20000	3.12	3.25	3.54
#4	k-means	corel	30000	**3.14**	**3.28**	**3.55**
#5	k-means	N-S	1000	2.91	3.14	3.49
#6	k-means	N-S	10000	3.16	3.25	**3.57**
#7	k-means	N-S	30000	**3.26**	**3.31**	3.57

Table 1: N-S dataset. Impact of distance regularization (CDM with $n_{\mathcal{N}} = 10$ and Sinkhorn algorithm [5]), of the dataset used for k-means clustering (uncorrelated Corel dataset or the N-S dataset itself) and of the vocabulary size.

where n_q is the number of queries, n is the number of dataset images and $n_{\text{rel}}(i)$ is the number of images which should be retrieved for image i. This measure indicates the average normalized position (between 0 and 1), in which a relevant image appears. For instance, ANR≈0.01 means that the average rank of a relevant image is approximately equal to 1000 for a dataset of 100000 images. Clearly, a lower ANR signifies better accuracy.

The measure proposed in [11] counts the average number of correct images among the four first images returned. This measure is meaningful because there are 4 relevant images per object in the N-S dataset.

5.2 CDM

5.2.1 CDM *vs* Sinkhorn algorithm

All the experiments in Table 1, Table 2 and Table 3 show a significant improvement when using a distance regularization method (CDM or Sinkhorn). Note that the parameters are summarized in the caption. Table 1 shows that the Sinkhorn algorithm improves the results. However, the gain due to the CDM is significantly higher, and this for all the tested parameters. Thus, regularization with local distances only is very important in our context. The relevance of the CDM is also confirmed by experiments on the preprocessed data of [11], as shown in Fig. 7.

Fig. 4 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 next

	clustering dataset	vocab. size	norm	$n_{\mathcal{N}}$	ANR	
					no CDM	CDM
#1	corel	10000	L_1	30	0.0522	0.0148
#2	corel	20000	L_1	10	0.0476	0.0238
#3	corel	20000	L_1	20	0.0476	0.0156
#4	corel	20000	L_1	30	0.0476	0.0145
#5	corel	20000	L_2	30	0.0528	0.0224
#6	corel	30000	L_1	30	0.0468	0.0133
#7	corel	50000	L_1	30	0.0416	**0.0118**
#8	lola	10000	L_1	30	0.0321	0.0063
#9	lola	20000	L_1	30	0.0240	**0.0046**
#10	lola	20000	L_2	30	0.0231	0.0053

Table 2: Lola dataset. Impact of k-means clustering dataset, vocabulary size, norm (Manhattan L_1 or Euclidean L_2) and number of neighbors $n_{\mathcal{N}}$ used in the CDM calculation.

four images seem wrong for both queries, they are in fact correct for the CDM, as the images correspond to the same location (Deutsche Transfer Bank) observed from significantly different viewpoints.

5.2.2 Neighborhood size of the CDM

The only parameter of the CDM is the neighborhood size $n_{\mathcal{N}}$. Fig. 5 shows the impact of this parameter on the performance of the iterative approach. We can observe that the sensitivity to this parameter is moderate: the accuracy increases significantly in the case of very small neighborhoods and decreases gracefully when using large neighborhoods. A small neighborhood also results in lower computational cost. In the rest of this paper, the size $n_{\mathcal{N}}$ is fixed to 10, although better results may be obtained by optimizing this parameter.

5.2.3 too-often-selected and never seen images

The impact of the CDM on the neighborhood reversibility is very important. This has been verified on the N-S dataset with a vocabulary size of 10000. For a neighborhood size of 10 the neighborhood symmetry rate (Eq. 3) increases from 0.37 to 0.62. The percentage of never seen images, see Section 3.1 for the definition, decreases from 9.7% to 0.2%. Similarly, for the 10200 queries of the N-S dataset the most frequent image is returned 54 times in the first 10 positions with the CDM, against 1062 times using the standard L_1 distance.

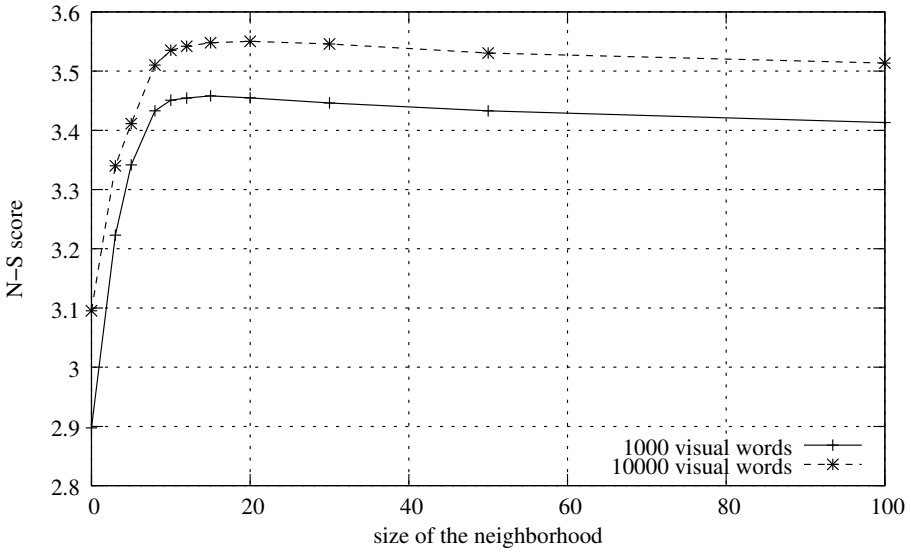


Figure 5: CDM: impact of neighborhood size $n_{\mathcal{N}}$ for the N-S dataset. Parameters: k-means clustering performed on corel, L_1 distance.

5.3 Impact of the parameters and variants

5.3.1 Clustering

We have implemented and evaluated the hierarchical clustering approach [3]. Comparing Table 3 Exp. #1 and Table 1 Exp. #2, we can see that hierarchical clustering reduces the accuracy. However, its significantly reduces the computational cost for assigning SIFT descriptors to visual words, especially for large vocabularies. Note that the concurrent approach of [14] offers similar efficiency as [3], but provides better accuracy (very similar to k-means).

The dataset used for the clustering may have an impact on the accuracy, as shown in Tables 1 and 2. For these two datasets we compare in column "clustering dataset" k-means clustering on an uncorrelated dataset (Corel) with k-means clustering on the evaluation dataset itself (either N-S or Lola). In both cases the results are improved by generating the visual vocabulary with a subset of the dataset on which the experiments are performed.

This confirms the observation made by Nistér and Stéwenius [11], i.e., that using the evaluation set for clustering significantly improves the results. This is particularly true when using a large vocabulary, as shown in Fig. 7 which shows some results obtained with the preprocessed dataset of [11].

When comparing the experiments #1-#4 with #5-#7 in Table 1, we can observe that the CDM is less influenced by the learning set. This remark does not

	clustering method	training set	vocab. size	N-S score no CDM	N-S score CDM	variant
#1	hierarchical	corel	10000	2.90	3.41	[3]
#2	k-means	corel	30000	2.68	3.37	L_2
#3	k-means	corel	30000	3.01	3.49	PCA
#4	k-means	corel	30000	3.12	3.59	MA$\times 2$
#5	k-means	corel	30000	3.07	3.60	MA$\times 3$
#6	k-means	corel	50000	3.21	3.60	MA$\times 2$
#7	k-means	corel	50000	3.18	**3.61**	MA$\times 3$

Table 3: N-S dataset. Impact of the variants and of the following parameters: clustering method (k-means or hierarchical [3]) performed on the corel dataset, vocabulary size, norm (L_1 if not specified or L_2), use of the PCA (36 dimensions), multiple assignment (MA) of descriptors to visual words. Fixed parameter: $n_{\mathcal{N}} = 10$.

hold for the Lola dataset (see Table 2). A possible explanation is that for this set the clustering was performed on the entire set of descriptors and not only on a subsample, hence emphasizing the adaptation of the visual vocabulary to the evaluation dataset.

5.3.2 Number of descriptors

Fig. 6 shows that the number of descriptors extracted for each image has a strong impact on the accuracy. We can observe that the accuracy increases up to a certain point only, i.e., using a too high number of descriptors decreases performance. A possible explanation is that the strongest interest points, i.e., with high cornerness values, are diluted among those with low cornerness, and that this results in noise in the frequency vectors. However, the best number of descriptors to be used depends on the other parameters. In particular, the CDM benefits from a high number of descriptors.

Note that for this experiment, the interest points have been generated using a low threshold and are then filtered based on their cornernesses value. This is slightly different from the standard setup, where the suppression of non-maxima is performed *after* the thresholding.

5.3.3 Vocabulary size

Table 1, Table 2 and Fig. 7 show that bigger vocabularies provide better retrieval accuracy. However, the gain is rather moderate, except when the visual vocabulary is learned on the evaluation set itself (see Fig. 7). In this

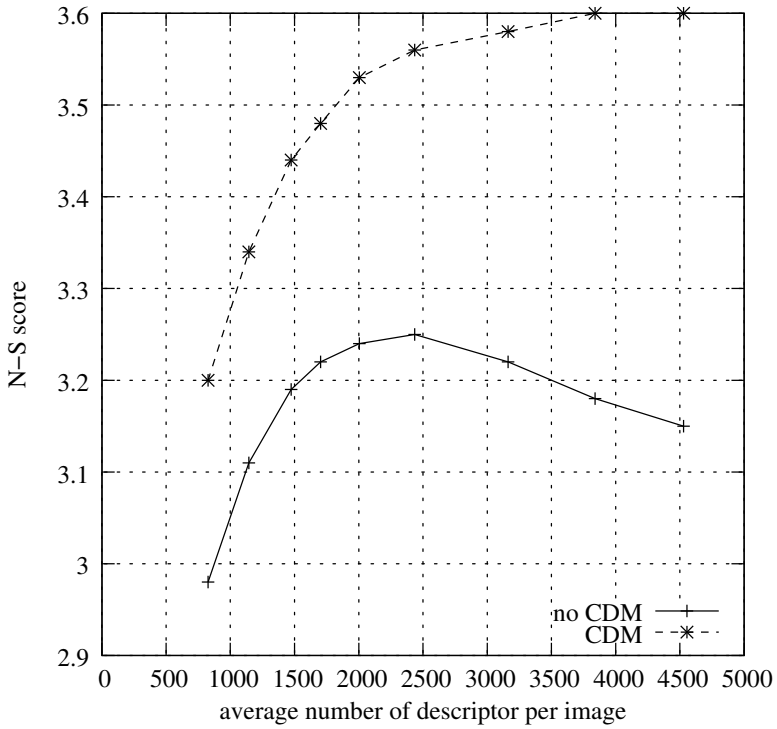


Figure 6: N-S dataset. Impact of the number of descriptors on the retrieval accuracy for a vocabulary of size 30000 learned on the N-S dataset.

case the bag-of-features image search system becomes very similar to an approach which matches individual descriptors, i.e., for very large vocabularies the number of visual words is equal to the number of descriptors in the clustering dataset and all the descriptors are used as centroids. Note that search results obtained when matching individual descriptors outperform those of bag-of-features based search, but significantly increase the search complexity.

5.3.4 Norm

It was observed in [3] that the Manhattan distance provides better results than the Euclidean one. This observation is confirmed in our experiments for the two datasets and also holds when the CDM is used, see Table 3 Exp. #2 and Table 2 Exp. #5 and #10. However, this observation depends on the dataset, as noticed in [14].

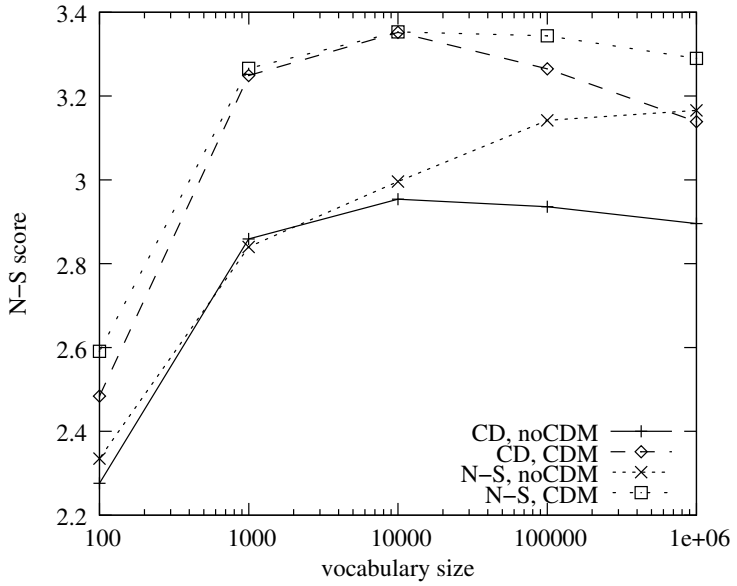


Figure 7: N-S dataset: impact of the vocabulary size and of the clustering dataset (independent dataset "CD" or the N-S set). These curves have been generated using the N-S preprocessed data [11]. Fixed parameter: $n_{\mathcal{N}} = 10$.

5.3.5 Dimensionality reduction of SIFT with PCA

We reduce the dimensionality of the SIFT descriptor with PCA from 128 to 36 dimensions. Table 3 Exp. #3 shows that PCA marginally reduces the accuracy, while decreasing the computational cost of the visual word assignment. However, the hierarchical SIFT assignment of [3] decreases the assignment cost more significantly and at the same time obtains comparable results, see Table 3 Exp. #1 and Exp. #3. Using PCA in this context is, therefore, of limited interest.

5.3.6 Multiple assignment of SIFT descriptors

The MA of SIFT descriptors to visual words slightly improves the accuracy of the search (see Table 3, Exp. #4 to #7) at the cost of an increased search time, due to the impact of the method on the visual word vector sparsity. For instance, for $V = 30000$ visual words the number of multiplications performed is 7 times higher for MA$\times$3 than for the simple assignment. 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.

5.3.7 Rank aggregation

Table 4 presents the results obtained with the rank aggregation method described in Section 4. The visual vocabularies have been generated using distinct SIFT subsamples of the N-S dataset, obtained by modifying the seed of the random number generator. The number of votes required for an image to be added to the ranking list is equal to 2 for 3 visual vocabularies, 3 for 5, 5 for 9 and 10 for 19.

The results show that rank aggregation improves accuracy. The scores are consistently improved for all sets of parameters. The best score obtained with rank aggregation is 3.68 against 3.57 for a single vocabulary. The trade-off between accuracy and efficiency can also be adjusted by choosing a smaller number of distinct visual vocabularies. Hence using only 3 distinct ranking sets is sufficient to obtain a fair improvement.

5.4 Comparison with the state-of-the-art

For the N-S dataset, the CDM approach obtains a N-S score of 3.55 (maximum 4) for a CDM computed with $n_{\mathcal{N}} = 10$ neighbors and 30000 visual words learnt on the Corel dataset. Combining the CDM with the MA improves this results to 3.61 for 50000 visual words. Our best score of 3.68 has been obtained using rank aggregation (see Table 4). The best previous score [11] is 3.29 for their

number of distinct visual vocab.	vocab. size	N-S score without CDM	N-S score with CDM
1	1000	2.91	3.49
	10000	3.16	3.57
	30000	3.26	3.57
3	1000	2.93	3.54
	10000	3.20	3.63
	30000	3.29	3.63
5	1000	2.94	3.55
	10000	3.21	3.64
	30000	3.31	3.65
9	1000	2.95	3.56
	10000	3.22	3.65
	30000	3.32	3.67
19	1000	2.96	3.57
	10000	3.22	3.66
	30000	3.33	**3.68**

Table 4: N-S dataset. Rank aggregation: impact of the number of distinct visual vocabularies (1 to 19), here learned on the N-S dataset itself. Note that the three first rows (1 distinct ranking only) correspond to no rank aggregation. Fixed parameter: $n_{\mathcal{N}} = 10$.

most time consuming approach and a visual vocabulary learned on the N-S dataset itself.

Our best ANR score for the Lola movie is 0.0046, significantly outperforming the previous best score 0.0132 [4]. Note that, by contrast to their work, we use only one type of descriptor (in this case their best score is 0.0196) and no temporal filtering. Our approach is still better (0.0118) if the visual words are learned on uncorrelated data. In [4] the visual vocabulary was learnt on the Lola dataset.

6 Conclusion

This paper introduces the contextual dissimilarity measure to compare frequency vectors of a bag-of-features image representation. This new measure is based on a distance regularization algorithm in the spirit of the Sinkhorn's algorithm, which projects distance matrices on doubly-stochastic matrices. In contrast to this algorithm, our regularization uses local distances only, similar to recently proposed methods for non-linear dimensionality reduction.

The performance of our approach has been demonstrated for a bag-of-features based image search system. A large set of experiments shows that the accuracy is significantly and consistently improved by the CDM for two different datasets. We also analyze several variants and the impact of the main parameters of our image search system. Our final system significantly outperforms the state-of-the-art on both datasets.

7 Acknowledgements

We would like to acknowledge J. Sivic, A. Zisserman, D. Nistér and H. Stewénius for kindly providing their datasets.

References

[1] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[2] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *International Journal of Computer Vision*, vol. 60, no. 1, pp. 63–86, 2004.

[3] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 2161–2168.

[4] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proceedings of the IEEE International Conference on Computer Vision*, 2003, pp. 1470–1477.

[5] R. Sinkhorn, "A relationship between arbitrary positive matrices and double stochastic matrices," *Annals of Mathematics and Statistics*, vol. 35, pp. 876–879, 1964.

[6] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 539–546.

[7] A. Frome, Y. Singer, and J. Malik, "Image retrieval and classification using local distance functions," in *Advances in Neural Information Processing Systems*, 2007, pp. 417–424.

[8] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov, "Neighbourhood components analysis," in *Advances in Neural Information Processing Systems*, 2005, pp. 513–520.

[9] K. Weinberger, J. Blitzer, and L. Saul, "Distance metric learning for large margin nearest neighbor classification," in *Advances in Neural Information Processing Systems*, 2006, pp. 1473–1480.

[10] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988.

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

[12] R. Fagin, R. Kumar, and D. Sivakumar, "Efficient similarity search and classification via rank aggregation," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 2003, pp. 301–312.

[13] J. Zobel, A. Moffat, and K. Ramamohanarao, "Inverted files versus signature files for text indexing," *ACM Transactions on Database Systems*, vol. 23, no. 4, pp. 453–490, 1998.

[14] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[15] H. Jegou, H. Harzallah, and C. Schmid, "A contextual dissimilarity measure for accurate and efficient image search," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[16] E. Hörster, R. Lienhart, and M. Slaney, "Image retrieval on large-scale image databases," in *Proceedings of the ACM international conference on Image and video retrieval*, 2007, pp. 17–24.

[17] C. Johnson, R. Masson, and M. Trosset, "On the diagonal scaling of Euclidean distance matrices to doubly stochastic matrices," *Linear Algebra and its Applications*, vol. 397, no. 1, pp. 253–264, 2005.

[18] J. Tenenbaum, V. de Silva, and J. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[19] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[20] G. Shakhnarovich, T. Darrell, and P. Indyk, *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice*, chapter 3, MIT Press, Mar 2006.