HAL
open science

A review of weighting schemes for bag of visual words image retrieval

Pierre Tirilly, Vincent Claveau, Patrick Gros

To cite this version:

Pierre Tirilly, Vincent Claveau, Patrick Gros. A review of weighting schemes for bag of visual words image retrieval. [Research Report] PI 1927, 2009, pp.47. inria-00380706

HAL Id: inria-00380706

https://inria.hal.science/inria-00380706

Submitted on 4 May 2009

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.

Publications Internes de l'IRISA
ISSN : en cours
PI 1927 – Avril 2009

A review of weighting schemes for bag of visual words image retrieval

Pierre Tirilly[*], Vincent Claveau[**], Patrick Gros[***]
pierre.tirilly@irisa.fr, vincent.claveau@irisa.fr, patrick.gros@inria.fr

Abstract: Current studies on content-based image retrieval mainly rely on bags of visual words. This model of image description allows to perform image retieval in the same way as text retrieval: documents are described as vectors of (visual) word frequencies, and documents are match by computing a distance or similarity measure between the vectors. But instead of raw frequencies, documents can also be described as vectors of word weights, each weight corresponding to the importance of the word in the document. Although the problem of determining automatically such weights, and therefore which words describe well documents, has been widely studied in the case of text retrieval, there is very little litterature applying this idea to the case of image retrieval. In this report, we explore how the use of standard weighting schemes and distance from text retrieval can help to improve the performance of image retrieval systems. We show that there is no distance or weighting scheme that can improve performance on any dataset, but choosing weights or a distance consistent with some properties of a given dataset can improve the performance up to 10%. However, we also show that in the case of very varied and general datasets, the performance gain is not significant.

Key-words: Weighting schemes, content-based image retrieval, bag of visual words, information retrieval, vector space model, probabilistic model, Minkowski distance, divergence from randomness, BM25, TF*IDF

Un aperçu des schémas de pondération pour la recherche d'images par sac de mots visuels

Résumé : *Les travaux actuels en recherche d'image par le contenu se basent sur un modèle en sac de mots visuels. Ce modèle permet de rechercher des images d'une façon très similaire à celle de la recherche textuelle : les documents sont représentés par des vecteurs de fréquences de mots (visuels), et sont appariés en calculant une distance entre ces vecteurs. Au lieu de comparer des fréquences brutes, il est possible d'utiliser des vecteurs de poids, chaque poids représentant l'importance d'un terme dans un document. Bien que de nombreuses méthodes pour déterminer automatiquement ces poids, et donc l'importance des termes, ont été développées pour les documents textuels, il existe peu de travaux équivalents dans le domaine de la recherche d'image. Dans ce rapport, nous étudions l'influence que peuvent avoir les schémas de pondération et les distances classiques de la recherche d'information textuelle sur les performances des systèmes de recherche d'images. Nous montrons qu'il n'existe pas de pondération ni de distance qui soit optimale sur n'importe quelles données, mais aussi que le choix d'une distance ou d'une pondération en adéquation avec les propriétés des données étudiées permet d'augmenter les résultats jusqu'à environ 10%. Cependant, nous montrons aussi que dans le cas de grandes collections d'images variées, les améliorations obtenues ne sont pas significatives.*

Mots clés : *Pondération, recherche d'image par le contenu, sac de mots visuels, recherche d'information, modèle vectoriel, modèle probabiliste, distance de Minkowski, divergence from randomness, BM25, TF*IDF*

---

[*] Texmex Project, CNRS
[**] Texmex Project, CNRS
[***] Texmex Project, INRIA Rennes

©IRISA – Campus de Beaulieu – 35042 Rennes Cedex – France – +33 2 99 84 71 00 – www.irisa.fr

1 Introduction

Acquiring, storing and exchanging image data is becoming easier and easier, thanks to new widespread technologies such as camera cellphones, high-speed internet... As a consequence, personnal as well as professional image databases tend to become very huge (for instance, FlickR[1] hosts more than 3 billions photos), and accessing such data effectively and efficiently is still an open problem. In this context, the fields of image categorization (*i.e.* finding the category an image belongs to, among several categories) and retrieval (*i.e.* finding images that are similar to a query image) have raised in the 90s to handle such databases. Figure 1 shows the principle of image retrieval systems. The two important parts of such systems are the descriptors used to describe images in the database and the way these descriptors are matched. First approaches relied on global image descriptors, that describe a whole image as a vector of numerical values. These values reflect some physical properties of images, such as color (*e.g* color histograms [4]) or texture (*e.g.* cooccurrences, Gabor filters... [11]). These descriptors allow a quite efficient retrieval process, but with a poor precision, because a global descriptor loses most of the information about the different concepts expressed in the image. Moreover, the kind of visual features that is indexed here does not fit the properties of some objects: a car, for instance, can be of any color. A more recent approach relied on the use of local descriptors. The idea is to detect small interesting regions of the images (*e.g.* corners) and consider images as sets of such regions. Images can then be matched by counting the number of similar regions they share. This method has two main drawbacks. First, the number of regions describing an image is usually high (from hundreds to thousands), making the computationnal cost of image retrieval prohibitive for huge databases. Secondly, the exact match between local descriptors makes this technique very effective for copy detection, but provides general-purpose retrieval systems a bad recall.

Recent approaches rely on *bag of visual words* or *bag of features*. Images are described as sets of elementary local features, such elementary features being typically groups of local descriptors. This framework makes the descriptor more robust to changes, by considering groups of local descriptors, and much more efficient, by comparing occurences of local features instead of comparing high-dimensional local descriptors two by two. Moreover, this description is very close to the traditional description of texts in information retrieval: a document is a set of elements (words or visual words). Finding the most relevant words to describe a document (by assigning them specific weights), and then effectively matching documents (through various matching functions), has been widely studied in the field of text retrieval, but this work has been poorly considered in image classification and retrieval.

In this report, we aim at exploring the influence of traditional information retrieval matching functions and weighting schemes for bag-of-visual words image retrieval, to see to which extent text retrieval methods can help to improve image retrieval. The paper is organized as follows: in the next section we present the principle of bag of visual words retrieval, its link to textual retrieval and some related work. In Section 3, we state some conceptual differences between bag of word image and text retrieval. Then in Section 4, we present some classical IR models, and the weighting schemes and distances we chose to test in our experiments. Section 5 presents the experimental settings and results, and, at last, Section 6 proposes a discussion about some of the results.

2 Bag-of-visual words model and related work

The bag of visual words model has been first introduced by Sivic and Zisserman [28] and Csurka *et al.* [8]. It describes images as sets of elementary local features called visual words. The whole visual word set is called the *visual vocabulary.* The description of an image database using bags of visual words relies on two steps:

1. Construction of a visual vocabulary.
2. Description of images using this vocabulary.

The vocabulary is built as follows (see Figure 2):

1. Detection of interest regions on a set of images (detectors: Harris-affine, Hessian-affine, MSER... [20]): some regions with geometric particularities (presence of corners, homogeneity...) are automatically extracted from the image.
2. Description of each interest region as a local descriptor (descriptors: SIFT, SURF... [21]): the interest regions are described as multidimensional numerical vectors, according to their content.
3. Clustering of the local descriptors: the descriptors are grouped using a clustering algorithm. Each resulting cluster (or group) corresponds to a visual word.

[1] www.flickR.com

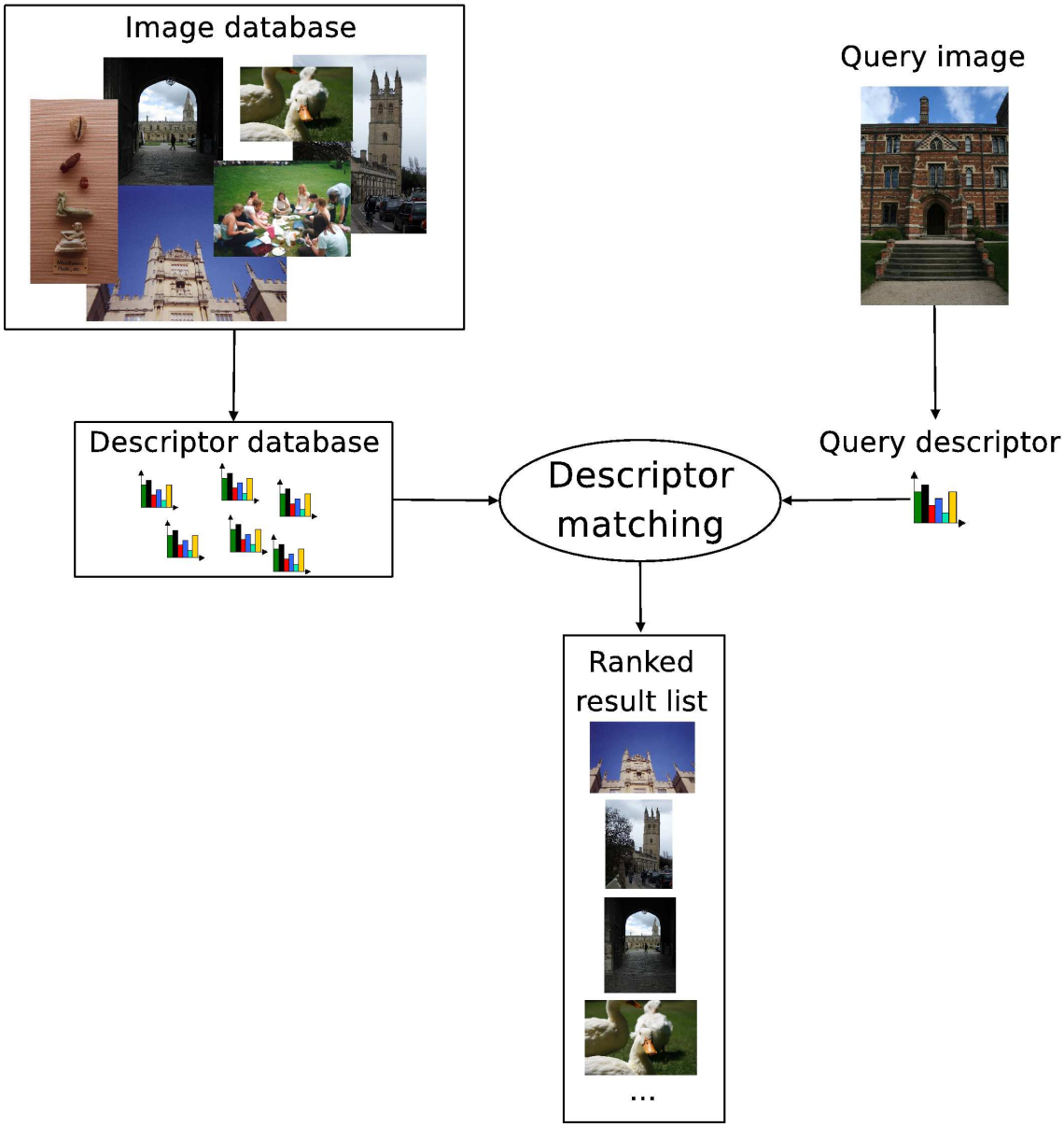


Figure 1: Typical architecture of an image retrieval system

We can then describe any image as a vector of visual words occurrences, as follows (see Figure 3):

1. Detection and description of interest regions in the image.
2. Assignation of each local feature to its nearest visual word in the vocabulary.
3. Description of the image as a vector of visual word frequencies (*i.e.* number of occurrences).

Images can then be matched by computing a distance between the vectors describing them. Classical L1 and L2 distances are generally used. This description of images as high-dimensional sparse vectors of visual word occurrences is very close to a classical text retrieval model: the Vector Space Model (VSM) [31], as stated in Sivic's first paper about this, called *Video-google: a text retrieval approach to object matching in videos* [28]. We can go even further in the analogy with text retrieval: as in the case of text retrieval, where each document is modeled by a set of index terms (*e.g.* a text about information retrieval will be indexed by the terms *information, retrieval, collection, document, index...*), the images are indexed by a set of visual words, visual words being index terms of the image collection. The problem is then to find which index terms are the most relevant to describe documents and how to match documents using these terms. This problem has been widely studied in the case of text retrieval, and several models have been proposed. The most common is the VSM model, as it allows to assign to each index term a specific weight that reflects this term's importance in the document.

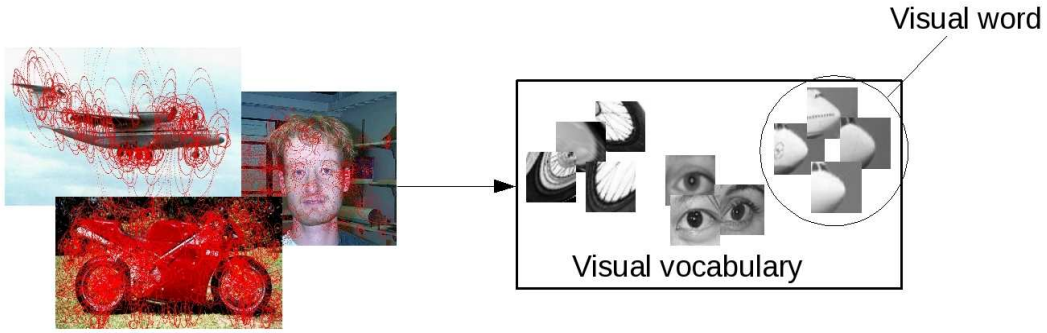


Figure 2: Construction of a visual vocabulary

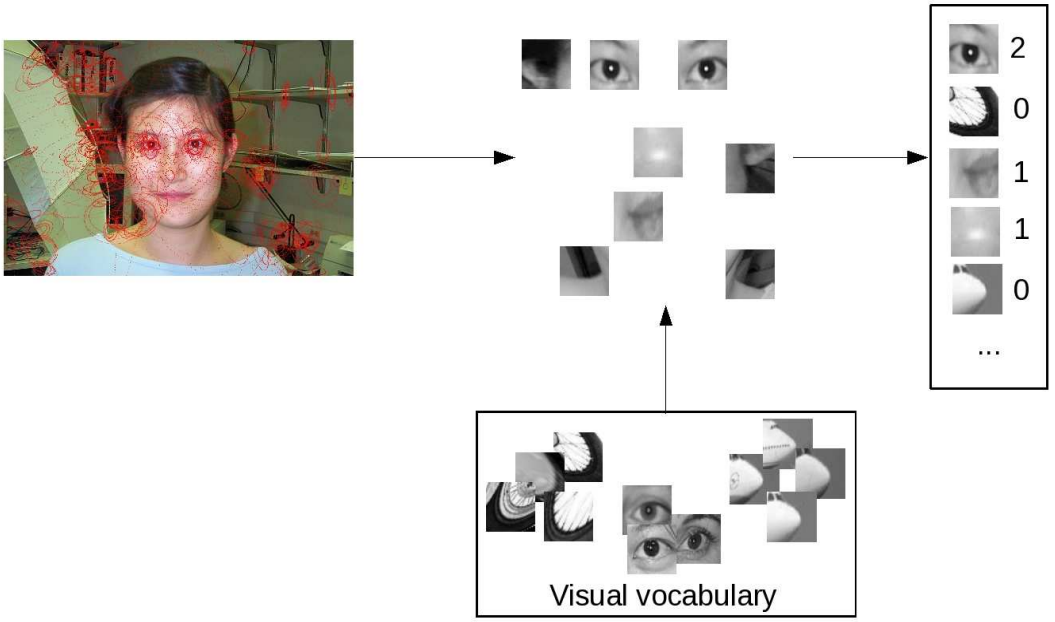


Figure 3: Description of an image as a vector of visual word frequences

Documents are therefore described as vectors of term weights instead of term occurrences. The most common weighting scheme is the TF*IDF weight, that combines the importance of the terms in the document (TF – Term frequency – part) and the importance of the terms in the collection (IDF – Inverse Document Frequency – part). But many other weighting schemes exist, and each of them has specific properties. Another interesting model class is the probabilistic models. They provide a probabilistic modeling of term importance and probabilistically motivated measures to match documents. In this study, we explore the impact of these two classes of models, vectorial and probabilistic, on image retrieval. These models, and the weighting schemes and matching functions we use in our experiments, are described in Section 4. However, there are a few differences between text retrieval and visual word-based image retrieval that we must take into account before using text retrieval techniques in the case of images. These differences are given in Section 3.

Bag of visual words approaches for image classification and retrieval have been intensively studied, however only a few papers go further in the analogy with text than the basic use of VSM and TF*IDF weighting. To our knowledge, only [16] proposes to compare weighting schemes for image retrieval, and only for 3 different weighting schemes: TF*IDF, binary weighting, and a new scheme they proposed, but this new scheme relies on the clustering stage of the system, not on the distribution of visual words over the collection: they use a soft assignment of the local descriptors to visual words. In the case of image classification, only TF*IDF and binary weights have been studied [32]. Moreover, there is little work on the distances used to perform bag of visual words image retrieval. Nister *et al.* compared $L1$ and $L2$ distances on a few experiments, and conclude that $L1$ performed better than $L2$, but they did not explain this behavior [22]. Jegou *et al.* proposed new dissimilarity measures for bag of visual words retrieval: one is based on nearest-neighbor scheme properties [14] and could be generalized to any application, the other relies on properties of the clustering stage of the bag of visual words description [15]. However, to our knowledge, no work have intensively explored the influence of classical distances such as Minkowski distances or the cosine similarity.

3 Fundamental differences between textual words and visual words

Bag of visual words share some properties with the bag of words model commonly used in text retrieval. The name *Video-google: a text retrieval approach to object matching in videos* of Sivic and Zisserman's first paper about this model [28] reflects this. Both bag of words models rely on the vector space model (VSM) of information retrieval. VSM describes documents as vectors of individual features. Text retrieval methods can therefore be used to perform and improve bag of visual words image retrieval. Among text retrieval techniques, the followings have been used in the case of images: *TF*IDF* weigthing [28, 8], inverted index [28], correlation measures, pLSA [3]. Whereas these two representations of images and texts have exactly the same form (a document is described as a set of index terms), some fundamental differences exist between them. The meaning of words, their frequency in documents or the nature of the queries are typically different. We propose, in the following, an overview of the differences between the two models.

3.1 Vocabulary

The first difference is the way the vocabulary is obtained. In the case of text, the vocabulary naturally results from the document collection: it is composed of the words occurring in the documents. Since it is made of words of a given natural language, it is then possible to use some knowledge about this language to improve the retrieval systems. An example of such knowledge is *stop-lists*, which typically contain grammatical words with no meaning for a retrieval system (*e.g.* a, the, you, is...). Such lists can typically be created only by humans to get optimal results. On the contrary, in the case of images, there are several ways to build vocabularies for each collection. The construction of the vocabulary involves several variables. We therefore can obtain lots of different vocabularies for one given image collection. Moreover, we do not have any *a priori* knowledge about the vocabulary, making some tasks such as the definition of stop-lists very difficult. Some of the parameters involved in the construction process of the visual vocabulary are detailed below:

- **Interest region detector**: the interest region detector (or key point detector) has naturally an incidence on the vocabulary. The words are closely related to the regions initially detected on the images. However, We can note that many detectors (Harris-affine, Hessian-affine...) find comparable types of regions, basically regions containing corners or changes in lightning. In contrast, some detectors, typically the Maximally Stable Extremal Region (MSER) detector, detect regions based on their stability, *i.e.* uniform regions (see [20] for details about the properties of the main detectors). Clearly, each kind of detector will produce different visual vocabularies, and different visual words distributions over the collection. We can also note that combining these two kinds of detectors permits to build a better visual vocabulary, as done in [28]. In the rest of this study, we will rely one the use of the Hessian-affine detectors, but most of the results will still be valid with other detectors.

- **Region descriptor**: the interest region descriptor can impact the quality of the visual words because it has an influence on the clusters resulting from the vocabulary construction process.

- **Clustering algorithm**: the clustering algorithm is one of the main bottlenecks of visual word-based retrieval. Many studies about it were published [19, 22, 23], and it has been shown that it can have a significant impact on the results [19, 23]. The choice of the clustering algorithm is essential because the quality of the visual words strongly depends on it. But it is also an awkward issue because descriptors are high-dimensional (typically, 128 dimensions for a SIFT descriptor), and the number of descriptors and clusters is generally high, raising a prohibitive computation time.
- **Size of the vocabulary**: since the vocabulary is built using a clustering algorithm, it is possible to tune the vocabulary size, by setting a specific number of cluster. This is the case most of the time, because clustering algorithms generally do not determine automatically the optimal number of clusters. This stage is critical since it can change the quality of the vocabulary, and so the performance of the retrieval system. If the vocabulary is too small, the visual words will tend not to be discriminant enough to separate relevant images from non-relevant ones. On the contrary, a too large vocabulary will tend to be less robust and can make the computational cost prohibitive. However, recent studies tend to show that larger vocabularies improve the quality of image retrieval [22, 23].

3.2 Semantics of the words

Textual words and visual words have a very different meaning. Whereas one textual word usually corresponds to one given concept (*e.g.* house, computer, democracy), one visual word corresponds to a part of an object, or potentially to parts of several different objects. Hence several visual words are needed to describe an object. This difference is essential because most of the retrieval models of information retrieval rely on the **word independance assumption**, *i.e.* they consider that a word occurs independantly from the others. This assumption is generally acceptable in text retrieval, although incorrect in some cases (*White House* for instance). In the case of images, this assumption seems much less realistic. Table 1 illustrates this: it shows, for the Caltech6 dataset, the quantity of visual words occurring in a given number of categories. We see that only 4 words occur in only one category whereas most of them appear in the 6 image categories. It therefore emphasizes the fact that words taken separatly are poorly significant. This property of the bag of visual words model has been exploited by many authors to improve the retrieval or classification performances [33, 6, 30, 13]. In this report, since we are only interested in studying traditional weighting schemes of information retrieval, we will work under the independance assumption, even if it does not seem very realistic in this context.

Number of categories	1	2	3	4	5	6
Number of words	4	3	11	15	81	6442

Table 1: Number of words according to the number of categories they occur in (Caltech6 dataset)

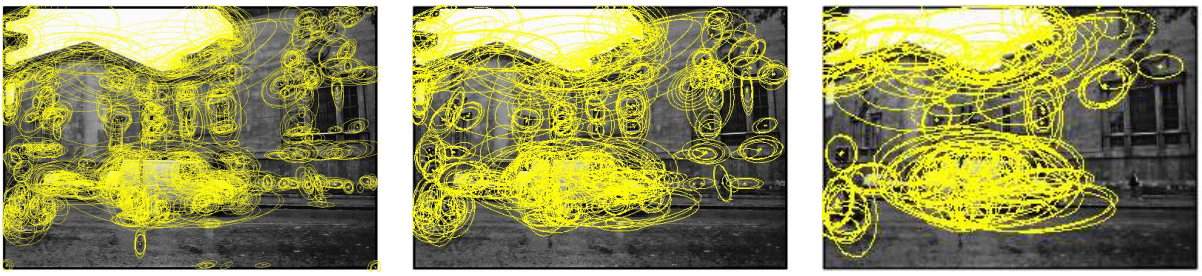
3.3 Document length and intra-document word frequency

In textual information retrieval, it is accepted that:

- the more a word occurs in a document, the better it describes this document;
- the longer a document, the more each word may occur in it.

These two remarks have led to many works about word frequency normalization and document length normalization. However, in the case of image retrieval, these two assumptions are not suited. First of all, the length (*i.e.* the number of visual words) of a document can depend on :

- the size of the input image: the bigger the image, the more regions are detected on it. This is illustrated in Figure 4;
- the appearance of the image: only a very few interest regions are detected on uniform parts of an image, whereas many will be found on heterogenous images. For instance, a street picture, containing people, cars and buildings, will produce much more interest regions than a seaside picture containing the sea, sand and a blue sky. Moreover, dark, light or blurred images will raise much less interest regions. Figure 5 illustrates these properties on images of similar size. Very few regions are detected on the first one because it is very dark, and on the second because of its uniformity. On the contrary, many regions are detected on the two other images, because they contain many details. In the case of the bird image, most of the regions only describe the background rocks, which will generally be considered as irrelevant in retrieval applications;
- the properties of the image detector, since some detectors tend to extract more interest regions than others. Tuning the detector parameters can also make it detect various numbers of interest regions.

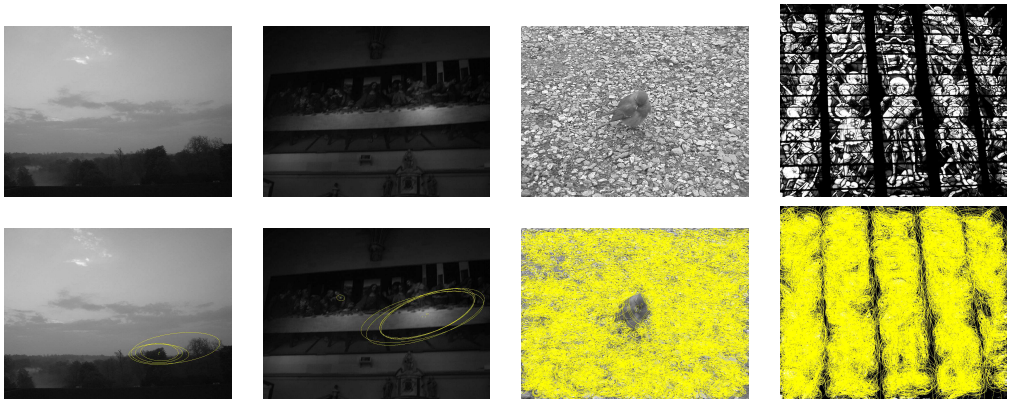


1024 x 720 pixels
2966 regions

512 x 360 pixels
1139 regions

256 x 180 pixels
288 regions

Figure 4: Number of regions detected according to the size of images



6 regions

11 regions

19,003 regions

25,995 regions

Figure 5: Detection of various numbers of regions on images of comparable size

These reasons explain the overall number of regions detected in an image. But we can also make some remarks about the intra-document frequency of a given visual word:

- in the same way the size of an image changes the number of regions detected on it, the size of an object in an image changes the number of visual words describing this object. This naturally affects the frequency of the words describing this object.
- the intra-document frequency also depends on the detector properties, as some detectors tend to detect the same region several times (see Figure 6);
- the intra-document frequency will also depend on the number of objects in the image: if an image contains the same object twice, the visual words belonging to that object will occur twice more. More generally, as different objects can contain the same visual word, this word's frequency will depend on the number of objects containing it in the picture;
- and last but not least, the size of the vocabulary has a strong influence on the intra-document frequency of visual words. The larger the vocabulary, the smaller the word frequency because two regions assigned to one given visual word in a small vocabulary will tend to be assigned to different words if the vocabulary grows.

All these reasons make the problem of term frequency normalization and document length normalization very difficult to manage. Whereas weighting techniques might handle the problem of the image size, the problem of normalizing according to the number of objects is a very awkward issue, as it requires to identify objects in the image.

3.4 Queries

A major difference between text retrieval and visual word-based image retrieval is the query length. In the case of text retrieval, the queries are usually short (a question containing a dozen words, *e.g.* in TREC[2] tasks) or very short (web queries are usually one or two word long). In the case of image retrieval, the queries are full documents or parts of a full document

[2]Text REtrieval Conference: `http://trec.nist.gov`

Figure 6: Example of a region detected several times on a motorbike image

(query per region, as done in [7]), and contain therefore much more index terms. This difference about query lengths has many consequences on the retrieval process and the adaptation of retrieval schemes to the case of images:

- inverted files become unefficient, because of the length of the queries and of the presence of noise in the vocabulary. Some authors tried to improve the inverted file scheme [14], while others overcome this problem by considering region queries [28, 7]. In particular, inverted files were presented as very efficient in [28] as they used very small objects as queries;
- weighting schemes derived from probabilistic retrieval have to be adapted since they do not provide the same weights for document terms and for query terms (see Section 4.2.1).

The query length rises then some issues that are specific to image retrieval. We can (at least partially) overcome these by using region queries. But is region querying a good approximation? It depends on the application: in the case of a query-by-example search system, the user has to provide a picture, so he can provide the good picture region as well. In the case of object discovering, or annotation tasks, or any task where the query is not an image explicitely provided by the user, region querying is not possible.

4 The models of information retrieval

4.1 The Vector Space Model (VSM)

The vector space model is the most popular model of information retrieval, and, to our knowledge, the only one used to perform visual word-based image retrieval. It represents documents as vectors of a vector space whose dimensions are the index terms. The similarity between two documents is computed as the angle between the vectors (cosine distance), or as a distance between vectors (typically, L_1 and L_2). This model makes the assumption of term independancy, as the vector space base is orthogonal. In this framework, the value of a vector component represents the term importance in the description of the document. It can therefore be the term presence or absence, the term frequency or any other weight, with the following assumption: the greater the weight is, the better the index term describes the document. The weight w_{ij} of index term t_i in document d_j is usually divided into three parts, so that $w_{ij} = l_{ij}.g_i.n_j$.

The local weight l_{ij}: the local weight l_{ij} reflects the term importance inside the document. It can aim at emphasizing terms of high frequency, limiting the influence of term frequency, or normalizing term frequency according to the length of the document.

The global weight g_i: the global weight emphasizes the term importance in the collection. The usual assumption is that, the more documents a term appears in, the less discriminative this term is. On the contrary, if a term occurs in a very few documents, then it might be a good descriptor of these documents' content. The classical global weight in IR is the inverse document frequency IDF.

The normalization factor n_j: this normalization factor depends on the document only, since it corresponds to the length of the weighted vector under a given norm. This normalization is necessary to get comparable distances, and therefore rank the documents effectively with respect to a query.

4.1.1 Local weighting

The local weights we considered in this study are shown in Table 2.

Term frequency: This is just the number of occurrences tf_{ij} of term t_i in document d_j.

Frequency logarithm: It aims at reducing the importance of high frequency terms, so that query terms with a small intra-document frequency still play a role in the query-document distance [5].

Augmented normalized frequency: This local weight has been proposed in the SMART retrieval system [27]. It contains two parts:

- a frequency normalization $\frac{\mathrm{tf}_{ij}}{\max_{t_k \in d_j} \mathrm{tf}_{kj}}$: it ajusts the term frequency according to the most frequent term in the document. In text retrieval, it is nearly equivalent to normalizing term frequencies according to the length of the document;
- an augmented frequency: it combines a term-presence based constant score (a) and a weighted ($1-a$) frequency-based score.

Binary weight: It just counts term presence; all frequency information is discarded.

DFR-like normalization: This weight normalizes term frequency according to the document length. We extracted from the definition of DFR matching scores (see Section 4.2.3).

Squared term frequency: It gives more importance to index terms with a high intra-document frequency.

BM25 term frequency: This frequency normalization is extracted from the BM25 formula. It normalizes term frequency according to document length with respect to a specific probabilstic model (see Section 4.2.2 for more details about it).

$l_1(t_i, d_j)$	Term frequency TF	tf_{ij}
$l_2(t_i, d_j)$	Frequency logarithm	$\begin{cases} 1+\log(\mathrm{tf}_{ij}) & \text{if } \mathrm{tf}_{ij} > 0 \\ 0 & \text{otherwise} \end{cases}$
$l_3(t_i, d_j)$	Augmented normalized frequency	$\begin{cases} a+(1-a)\frac{\mathrm{tf}_{ij}}{\max_{t_k \in d_j}(\mathrm{tf}_{kj})} & \text{if } \mathrm{tf}_{ij} > 0 \\ 0 & \text{otherwise} \end{cases}$
$l_4(t_i, d_j)$	Binary	$\begin{cases} 1 & \text{if } \mathrm{tf}_{ij} > 0 \\ 0 & \text{otherwise} \end{cases}$
$l_5(t_i, d_j)$	DFR-like normalization	$\mathrm{tf}_{ij} \cdot \frac{l_{avg}}{l_j}$
$l_6(t_i, d_j)$	Squared TF	tf_{ij}^2
$l_7(t_i, d_j)$	BM25 TF	$\mathrm{tf}_{ij} \cdot \frac{k_1+1}{\mathrm{tf}_{ij}.k1(1-b+b.\frac{l_j}{l_{avg}})}$ with $k_1 = 1.2, b = 0.75$

Table 2: Local weighting schemes of term t_i in document d_j (l_j: d_j length, l_{avg}: average document length)

4.1.2 Global weighting

Table 3 shows the different global weights we studied in this report.

Inverse Document Frequency (IDF): The inverse document frequency emphazises the terms that occur in a small number of documents. The idea is that terms occurring in many documents are more general terms, less discriminative, whereas a rare term will be a more precise document descriptor [31]. In image retrieval, we suppose that these frequent words will correspond to meaningless background visual words.

Probabilistic IDF: This weight has the same principle as the classical IDF, but it is based on probabilistic considerations. It comes from the BM25 formula (see Section 4.2.2 for details).

Squared IDF: It gives more importance to terms with a high IDF. This is supposed to give a better precision to retrieval systems.

(Mean TF)*IDF: We propose this global weight for images. The idea is that the mean frequency of a visual word may be a good clue of its importance. This is based on two remarks (see Section 3.3):

- some detectors tend to detect several times the same region;
- parts of objects that are repeated are generally significant (eyes, wheels, windows...).

Squared (mean TF)*IDF: It makes the (mean TF)*IDF weight more important.

$g_0(t_i)$	No weigth	1
$g_1(t_i)$	Inverse document frequency (IDF)	$\log(\frac{N}{df_i})$
$g_2(t_i)$	Probabilistic IDF	$\max(0, \log(\frac{N-df_i}{df_i}))$
$g_3(t_i)$	Squared IDF	$\log(\frac{N}{df_i})^2$
$g_4(t_i)$	(Mean TF) * IDF	$\overline{tf_i}\log(\frac{N}{df_i})$
$g_5(t_i)$	Squared (mean TF) * IDF	$[\overline{tf_i}\log(\frac{N}{df_i})]^2$

Table 3: Global weighting schemes of term t_i (N: number of documents in the collection, df_i: number of documents containing t_i, $\overline{tf_i}$: mean frequency of t_i in the documents containing it)

4.1.3 Normalization

The normalization factor aims at setting all query-document distances in a similar range, so they can be compared to rank the documents. It has just to be consistent with the distance used. For any Minkowski distance Lk, the associated normalization will then be the inverse of the Lk norm of the document vector:

$$n_j = \frac{1}{||D_j||_{Lk}} = \frac{1}{\left(\sum_i w_{ij}^k\right)^{\frac{1}{k}}}$$

4.1.4 Distances

Once the document vectors $d = (d_1, d_2, \ldots, d_n)$ have been computed, we can compare them to a query $q = (q_1, q_2, \ldots, q_n)$ using a distance function, so we can rank documents from the most similar to the less similar with respect to that query. The following distances are commonly used for textual information retrieval:

- L1 distance: $d_{L1}(d, q) = \sum_i |d_i - q_i|$
- L2 distance: $d_{L2}(d, q) = \sqrt{\sum_i (d_i - q_i)^2}$
- cosine similarity: $d_{cos}(d, q) = \frac{\sum_i d_i . q_i}{\sum_i d_i^2 . \sum_i q_i^2}$

Moreover, when the vectors are normalized, L2 distance and the cosine similarity are linked by the following relation:

$$d_{L2}(d, q)^2 = 2(1 - d_{\cos}(d, q)) \tag{1}$$

Since the functions $f(x) = x^2$ and $g(x) = 2(1 - x)$, are monotonic, they do not change the rankings provided by the original distance. Hence, we see that L2 and the cosine similarity provide the same document rankings for any query. As long as we consider L2, it is therefore not necessary to study the cosine similarity. Furthermore, instead of limiting our study to L1 and L2, which are specific cases of the general Minkowski distance Lk, we can use Minkowski distances with any real value of distance parameter k. Lk distances with non-integer k are called *fractional distances*. Such distances with $k < 1$ yield better results in high-dimensional vector spaces than the classical $L1$ and $L2$ distances [1]. Here are some properties of Minkowski distances in high dimensions:

- As the vector dimension increases, the mean distance from any vector to another tend to be constant. This is an effect of the so-called *curse of dimensionality*. Therefore, the higher the dimension, the less distances between vectors are significative;

- Using low values of k limits the effect of the curse of dimensionality by reducing the importance of the local distance (*i.e.* distance for a given coordinate) on each dimension. It then yields good results for data mining and retrieval applications;
- Lower values of k also allows to deal better with noise in the data. However, when the data is very noisy, Lk distances tend to be equivalent for any value of k.

Although the experiments reported in [1] deal with much less dimensions (up to 168) than in our case, these results may still be interesting for bag of visual word retrieval. We therefore considered several real values of distance parameter k in our experiments (see Section 5.6).

4.2 Probabilistic retrieval

The theory of probabilistic retrieval has risen in the 60s and has been extensively studied in the following years. This theory states the principles of retrieval in a generic way, *i.e.* in any case when we need to provide information to a user, with respect to his information need. It has been studied in the case of text retrieval, but can be used in every context such as image, video or multimedia retrieval. Probabilistic models aim at expressing relevance of documents to a given query as a *probability of relevance*. So, given a query, these probabilities of relevance allow a system to rank the collection documents in order of relevance, according to the *Probability Ranking Principle (PRP)*, as stated by Robertson [25]:

Probability Ranking Principle : If retrieved documents are ordered by decreasing probability of relevance on the data available, then the system's effectiveness is the best to be gotten for the data.

Relying on this principle, we can make two remarks:

- The probabilities of relevance give information on the order the system should return documents. Since we are only interested in getting this order, and not the real probability values, we may build matching functions relying on relevance probabilities, or on any transformation applied to these probabilities, as long as this transformation is order-preserving;
- Once the PRP is stated, the problem is clearly to determine the values of relevance probabilities. To do so, many models can be used. They rely on the type of documents considered, the way the documents are described and assumptions about the meaning and the distribution of documents descriptors. In the case of text retrieval, many models have already been tested, and they often yield good results.

In the rest of this section, we will review matching measures derived from the PRP in the context of text retrieval. We will first show that usual probabilistic matching measures can be seen as a particular case of similarity in the VSM. Then we will review some popular probabilistic models and extract from their matching measures local and global weights to use them in the VSM model.

4.2.1 Relating probabilistic matching scores to the VSM

Probabilistic retrieval usually provides a function to match a query q with a document d_j, and not only term weights that can be handled by any distance in a vector-space context. However, these matching functions can be adapted to provide term weights instead of a global score. A probabilistic matching functions PM usually takes the following form:

$$PM(d_j, q) = \sum_i q_i.w(d_{ij}) \tag{2}$$

This is simply a dot product between vectors q and $w(d_j)$. We can therefore make three remarks:

- $w(d_j)$ is the weighted vectors of occurrences of index terms in document d_j;
- the query vector q is not weighted, as it is often the case in text retrieval, where queries are usually much shorter than documents. This is not the case in image retrieval, as stated in Section 3;
- any distance can be used to compare the vectors, not only the dot product.

So for each probabilistic model considered, we only have to identify the weights w_{ij} to use them in a VSM context. This weights can also be seen as a product $w_{ij} = l_{ij}.g_i$ of local weight l_{ij} and global weight g_i.

4.2.2 Best Match (BM) measures

The Best Match (BM) measures, and especially the BM25 measure, are, to our knowledge, the best matching functions for text retrieval. At least are they considered as the state-of-the-art baseline to evaluate new retrieval schemes. In this paper we focus on the BM25 formula, as it showed better retrieval capabilities than the others on TREC experiments (see [26] for a precise description of BM25 formula and experimental results on TREC collections). This formula is given in Equation 3 (N is the number of documents in the collection, n_i the number of documents containing t_i, R_q the number of documents that are relevant to query q and r_{iq} the number of relevant documents containing t_i). As stated in Section 4.2.1, we can extract from Equation 3 a local weight (Equation 4) and a global weight (Equation 5).

$$\mathrm{BM25}(d_j, q) = \sum_{t_i \in q} \frac{tf_{ij} * (k_1 + 1)}{K + tf_{ij}} \cdot tf_{iq}) \cdot \log \frac{(r_i + 0.5)(N - n_i - R_q + r_{iq} + 0.5)}{(R_q - r_{iq} + 0.5)(n_i - r_{iq} + 0.5)} \tag{3}$$

$$l_BM25(t_i, d_j) = \frac{tf_{ij} * (k_1 + 1)}{K + tf_{ij}} \tag{4}$$

$$g_BM25(t_i, q) = \log \frac{(r_{iq} + 0.5)(N - n_i - R + r_{iq} + 0.5)}{(R - r_{iq} + 0.5)(n_i - r_{iq} + 0.5)} \tag{5}$$

The local weight part contains a normalization of tf_{ij} according to the expected meaning of the word in the document (is it a good document descriptor or not ?) and to the document length. This local weight is derived from a specific word frequency model, *Harter's 2-Poisson model* [10, 18]. It models word frequency as a mixture of two Poisson distributions, the first distribution corresponding to the frequency of relevant descriptor terms, the second one to the frequency of irrelevant document terms. This local weight is the l_7 local weight we presented in Section 4.1.1.

The global weight part is directly derived from Robertson's PRP [17]. It is supposed, given relevance information about the document, to provide optimal global weights to rank documents according to their probability of relevance. When no relevance information is available, it reduces to a IDF-like formulation (the 0.5 are added for estimation reasons). Although relevance information is not available in real applications, this formulation has useful implications:

- it can be used to perform relevance feedback or pseudo-relevance feedback, by injecting the (pseudo-)relevance information to the weights over one or several feedback loops;
- when relevance information is available, as it is usually the case with test collections, such weighting provides a upper bound limit of the performance of a retrieval system, *i.e.* shows the best theoritical results a retrieval system can yield. This property may be particulary useful as an evaluation or optimization protocol for visual word-based systems, since they imply many parameters (region detector and descriptor, clustering algorithms, test queries) and finding the optimal ones is still an awkward issue.

In these experiments, we consider the version of this weight that does not use relevance information, which corresponds to the global weight g_2 we presented in Section 4.1.2.

4.2.3 Divergence from randomness

The Divergence From Randomness (DFR) models are probabilistically motivated matching measures, proposed by Amati and van Rijsbergen [2]. They are based on two assumptions:

- the index terms are randomly distributed over the documents of a collection. Their frequency in each document follows a particular *model of randomness*, whether they are good descriptors of the document or not.
- when a term is a good descriptor of a document, its frequency in the document is higher than in other documents. We can therefore identify good description terms, as their frequency *diverge from the randomness model*.

In such models, the term weights are computed as the product of two information sources, Inf_1 and Inf_2, which are thereselves derived from two probabilities $Prob_1$ and $Prob_2$, as shown in Equation 6. The probability $Prob_1$ corresponds to the probability that the index term t_i has the frequency tf_{ij} in document d_j according to the randomness model. It depends on the whole collection properties. The information source Inf_1 related to $Prob_1$ is called the *informative content* of the term in a document, *i.e.* it measures the amount of information provided by this term in any document. We can therefore see it as a global weight. The second information source Inf_2 describes the *information gain* (or *loss*) due to the term t_i when considering it as a good descriptor of document d_j. It is related only to the documents that contain term t_i, instead of the whole collection, and is computed as the complementary of probability $Prob_2$, which models the risk of taking t_i as a good descriptor for document d_j.

$$\begin{aligned} w(t_i, d_j) &= Inf_1(t_i, d_j).Inf_2(t_i, d_j) \\ &= -\log_2(Prob_1(t_i, d_j)).(1 - Prob_2(t_i, d_j)) \\ &= -\log_2(Prob_1(t_i, d_j)^{1-Prob_2(t_i,d_j)}) \end{aligned} \quad (6)$$

Table 4 shows the different randomness models we tested in the case of image retrieval. The Bernouilli (or binomial) models of randomness (called here P and D), the Bose-Einstein models (G and B) and the inverse document frequency models (In and Ine) are precisely developped in [2]. Moreover, we introduce the hypergeometric randomness model HG.

The hypergeometric model HG: We propose this model that seems, by definition, more suited to the case of bag of visual words image retrieval than the other randomness models. Whereas the Bernouilli models consider term frequency as the result of multiple successive draws of one token, the hypergeometric model considers term frequency as a unique draw of several tokens. As stated in Section 3.2, term occurrences in texts correspond to the presence of a given topic in a document, and the longer the document, or the more focused the document, the more occurrences of an index term appear in it (corresponding to a Bernouilli model). On the contrary, in the case of images, a given number of term occurrences corresponds to a given object (*e.g.*: a motorbike has two wheels), so adding an object to an image corresponds to add several words simultaneously to the document; a hypergeometric model represents such draws of several words among all possible words.

The model is the following: let tf_{ij} be the term t_i frequency in document d_j, l_j the length of document d_j, CF_i the collection frequency of term t_i (the number of occurrences of t_i in the whole collection) and CF^* the total number term occurrences in the collection, the l_j tokens of document d_j are randomly picked from the CF^* tokens of the whole collection. The probability of getting tf_{ij} tokens of index term t_i among the CF_i occurrences of t_i in the collection, according to the hypergeometric law definition, is then:

$$prob_1(\mathrm{tf}_{ij}) = \frac{\begin{pmatrix} \mathrm{CF}_i \\ \mathrm{tf}_{ij} \end{pmatrix} \begin{pmatrix} \mathrm{CF}^* - \mathrm{CF}_i \\ l_j - \mathrm{tf}_{ij} \end{pmatrix}}{\begin{pmatrix} \mathrm{CF}^* \\ l_j \end{pmatrix}}$$

Then, in order to estimate $Inf_1(t_i, d_j) = -\log(prob_1(t_i, d_j))$, we need to compute many factorial logarithms (see Table 4). To do so, we use Ramanujan's approximation [24], which provides a better approximation than Stirling's formula used in [2]. Ramanujan's approximation is:

$$\log(k!) \approx k\log(k) - k + \frac{k(1 + 4k(1 + 2k))}{6} + \frac{\log(\pi)}{2}$$

Table 4.2.3 shows the two information gain models proposed by Amati and van Rijsbergen. They represent the information gain provided by adding one more occurrence of term t_i to document d_j. The first one (refered as L) relies on Laplace's law of succession. The other one (B) models this gain using Bernouilli processes (see [2] for details).

In addition to these two information sources, they add a normalization of intra-document term frequencies according to the document length. The idea is to consider the proportion of the document occupied by the term, instead of the raw number of occurrences, since, in the case of text, longer documents naturally contain more occurrences of each index term than short document. The normalizations proposed by Amati and van Rijsbergen are given in Table 6. According to Section 3.3, this normalization may not be suited to image retrieval, as term frequency can be a fonction of the size or number of objects in an image.

4.2.4 Language models

Language models (LM) have been first developed in the field of speech recognition and machine translation, then adapted to textual information retrieval [29]. They allow us to consider documents not only as sets of independant words (*unigram* LM), but also as sequences of n ordered words (n-gram LM). Given a document D, the probability of relevance of D to a query $Q = (w_1 w_2 \dots w_k)$ is computed as the probability that the model D generates Q, as following:

$$\mathrm{Pr}_D(Q) = \prod_{i=1}^{k} \mathrm{Pr}_D(w_i | w_{i-n+1} \dots w_{i-1})$$

Although unigram LM can be directly used in the case of image retrieval, the n-gram case require some specific adaptation [30]. Moreover, contrary to other probabilistic models, language models cannot be simply reduced to a VSM weighting framework:

Name	Model of $Prob_1$	Approximation for Inf_1
P	$\left[\binom{\mathrm{CF}_i}{\mathrm{tf}_{ij}} \left(\frac{1}{N}\right)^{\mathrm{tf}_{ij}} \left(\frac{N-1}{N}\right)^{\mathrm{CF}_i - \mathrm{tf}_{ij}}\right]$	$\mathrm{tf}_{ij} \,.\, \log_2 \frac{\mathrm{tf}_{ij}}{\lambda} + \left(\lambda + \frac{1}{12.\,\mathrm{tf}_{ij}} - \mathrm{tf}_{ij}\right) . \log_2 e + 0.5 \log_2(2\pi.\,\mathrm{tf}_{ij})$ with $\lambda = \frac{\mathrm{CF}_i}{N}$
D	$\left[\binom{\mathrm{CF}_i}{\mathrm{tf}_{ij}} \left(\frac{1}{N}\right)^{\mathrm{tf}_{ij}} \left(\frac{N-1}{N}\right)^{\mathrm{CF}_i - \mathrm{tf}_{ij}}\right]$	$\mathrm{CF}_i \,.D(\phi, p) + 0.5 \log_2(2\pi.\,\mathrm{tf}_{ij}(1-\phi))$ with $D(\phi, p) = \phi \log_2 \frac{\phi}{p} + (1-\phi).\log_2(\frac{1-\phi}{1-p})$
G	$\frac{(\mathrm{CF}_i - \mathrm{tf}_{ij} + 1)\cdot\ldots\cdot\mathrm{CF}_i \cdot (N-1)}{(N+\mathrm{CF}_i - \mathrm{tf}_{ij} - 1)\cdot\ldots\cdot(N+\mathrm{CF}_i - 1)}$	$-\log_2(\frac{1}{1+\lambda}) - \mathrm{tf}_{ij} \cdot \log_2(\frac{\lambda}{1+\lambda})$ with $\lambda = \frac{\mathrm{CF}_i}{N}$
Be	$\frac{(\mathrm{CF}_i - \mathrm{tf}_{ij} + 1)\cdot\ldots\cdot\mathrm{CF}_i \cdot (N-1)}{(N+\mathrm{CF}_i - \mathrm{tf}_{ij} - 1)\cdot\ldots\cdot(N+\mathrm{CF}_i - 1)}$	$-\log_2(N-1) - \log_2(e) + f(N+\mathrm{CF}_i - 1, N+\mathrm{CF}_i - \mathrm{tf}_{ij} - 2) - f(\mathrm{CF}_i, \mathrm{CF}_i - \mathrm{tf}_{ij})$ where $f(n,m) = (m+0.5)\cdot\log_2\left(\frac{n}{m}\right) + (n-m)\cdot\log_2(n)$
In	$\left(\frac{n_i+0.5}{N+1}\right)^{\mathrm{tf}_{ij}}$	$\mathrm{tf}_{ij} \cdot \log_2 \frac{N+1}{n_i+0.5}$
In_e	$\left(\frac{n_i+0.5}{N+1}\right)^{\mathrm{tf}_{ij}}$	$\mathrm{tf}_{ij} \cdot \log_2 \frac{N+1}{n_e+0.5}$ with $n_e = N \cdot \left(1 - \left(\frac{N-1}{N}\right)^{\mathrm{CF}_i}\right)$
HG	$\frac{\binom{\mathrm{CF}_i}{\mathrm{tf}_{ij}}\binom{\mathrm{CF}^* - \mathrm{CF}_i}{l_j - \mathrm{tf}_{ij}}}{\binom{\mathrm{CF}^*}{l_j}}$	$\log((\mathrm{tf}_{ij})!) + \log((\mathrm{CF}_i - \mathrm{tf}_{ij})!) + \log((l_j - \mathrm{tf}_{ij})!) + \log((\mathrm{CF}^* - \mathrm{CF}_i - l_j + \mathrm{tf}_{ij})!)$ $+ \log(\mathrm{CF}^*!) - \log(\mathrm{CF}_i!) - \log((\mathrm{CF}^* - \mathrm{CF}_i)!) - \log(l!) - \log((\mathrm{CF}^* - l_j)!)$ and $\log(k!) \approx k\log(k) - k + \frac{k(1+4k(1+2k))}{6} + \frac{\log(\pi)}{2}$

Table 4: Models of randomness and their approximation
tf_{ij}: number of occurrences (frequency) of term t_i in document d_j
CF_i: number of occurrences of term t_i in the collection
CF^*: number of occurrences of all terms in the collection
l_j: number of term occurrences in document d_j (length of the document)
N: number of documents in the collection
n_i: number of documents containing t_i

Name	Inf_2
L	$\frac{1}{\mathrm{tf}_{ij}+1}$
B	$\frac{\mathrm{CF}_i+1}{n_i\cdot(\mathrm{tf}_{ij}+1)}$

Table 5: Models of divergence
tf_{ij}: number of occurrences (frequency) of term t_i in document d_j
CF_i: number of occurrences of term t_i in the collection
n_i: number of documents containing t_i

H0	tf_{ij}
H1	$\mathrm{tf}_{ij} \cdot \frac{avg_l}{l_d}$
H2	$\mathrm{tf}_{ij} \cdot \log_2(1 + \frac{avg_l}{l_d})$

Table 6: tf_{ij} normalization for DFR weights
tf_{ij}: number of occurrences (frequency) of term t_i in document d_j
l_j: number of term occurrences in document d_j (length of the document)
avg_l: average document length

as the VSM relies on the term independance assumption, there is no simple way to adapt a n-gram LM to the VSM model, because such LM include information about word dependences. In the unigram case, the model reduces to:

$$\mathrm{Pr}_D(Q) = \prod_{i=1}^{k} \mathrm{Pr}_D(w_i)$$

By considering log-probabilities, we get:

$$\log(\mathrm{Pr}_D(Q)) = \log[\prod_{i=1}^{k} \mathrm{Pr}_D(w_i)] = \sum_{i=1}^{k} \log(\mathrm{Pr}_D(w_i))$$

According to Section 4.2.1, we can then use log-probabilities of word occurrences as weights in a VSM context. However, this approach provided poor results on a few experiments we performed. One last technique would be to consider distances between probability distributions (with similarity measures such as Kullback-Liebler divergence). Language models are not studied further in this report.

5 Experiments

5.1 Datasets

We performed these experiments on several datasets that are commonly used in image retrieval or categorization papers. They have different properties detailled below.

Caltech-6 We built this dataset from images from Caltech4 and Caltech101 datasets. It contains 5,415 images divided into 6 categories. The categories are: airplanes (1,024 images), backgrounds (900), car rears (1155), faces (450), guitars (1,030) and motorbikes (824). Although images from one category can be varied, this is an easy dataset, because it contains very few categories, with many images per category. It is therefore easy to obtain a good vocabulary for this dataset, containing very discriminant words.

Caltech-101 [9] This dataset contains 8,197 images in 101 categories, each category containing from 31 to more than 800 images. This is a very difficult challenge, since it contains lots of categories. In the case of visual word-based retrieval, it is particulary difficult to get a good vocabulary, since some categories contain much less images than the others: it is difficult to get category-specific visual words in such conditions, when using a uniform sampling of images to get the clustering data.

Nister [22] This dataset contains 2,550 objects, with 4 relevant images per object (and so, 10,200 images in the whole collection). This dataset is easier than the previous ones, since the objects within a group of relevant images are very similar: they only vary in position, orientation or illumination, but not in shape as it is the case in Caltech datasets. However, it remains interesting as it contains very few relevant images for a given query. It is therefore well designed for retrieval evaluation, whereas the preceeding sets were more classification-oriented.

Oxford buildings [23] This dataset contains images of Oxford buildings. It aims at retrieving images of a given building according to a query image of this building. As in the case of the Nister dataset, the query objects have little intra-class variations, but they can be partially occluded. The dataset comes with a very complete groundtruth for 55 queries, sorting the relevant documents according to their occlusion degree. In these experiments, we only consider binary relevance judgements: an image is relevant if any part of the query object appears on it, even if this part represents less than 25% of the object (images refered as "junk" in the groudtruth). Whereas the groundtruth provides the precise position of the buildings in the query images (to perform retrieval from short queries), we only considered full images as queries in this study.

5.2 Queries

In these experiments, we only used complete images as queries. Table 7 shows, for each dataset, the number of images from the dataset we used as queries. All query images were randomly chosen, except in the case of Oxford data where queries are provided with the data.

Dataset	Query number
Caltech6	200
Caltech101	200
Nister	300
Oxford	55

Table 7: Number of queries used for each dataset

5.3 Evaluation Measures

We evaluate the search results using standard performance measures of information retrieval: recall, precision and Mean Average Precision (MAP). In the following, N denotes the number of documents of the whole collection, Rl_q the documents that are relevant to query q and Rt_q the documents retrieved by the system when submitting query q.

Precision:

$$P = \frac{|Rl_q \cap Rt_q|}{|Rt_q|}$$

The precision is usually computed only on the first n retrieved documents, since they are the only interesting ones to the user (typically, we consider that a user will watch the 20 or 30 first results to a query, but not the 200th result). The values of n are called Document Cut-off Values (DCV). We computed precision after 10, 20, 50 and 100 documents for the two Caltech datasets, since some categories contain more than 100 images. In the case of the Nister dataset, we only computed the precision after 4 documents, since there are only 4 relevant documents per query. This correponds to the score proposed by Nister and Stewenius to evaluate their dataset [22]. For the Oxford dataset, we computed precisions after 5, 10, 20 and 50 documents, since there are less relevant documents per query than in Caltech datasets.

Mean Average Precision (MAP): The average precision is computed as the average of the precisions computed after each relevant document retrieved. The MAP is then computed as the mean value of the AP over a set of test queries. Whereas the precision gives us only information about the relevance of the first returned documents, the MAP gives us information about the whole ranking: the more relevant document are low ranked, the worse the MAP is.

Recall:

$$R = \frac{|Rl_q \cap Rt_q|}{|Rl_q|}$$

Recall is traditionally computed to check if the retrieval system does not "forget" any relevant documents. In our case, the search is performed exhaustively, by matching each document of a collection to queries, so every relevant document is ranked among the search results. In this context, recall does not provide more information than precision and MAP. For a given DCV, the higher the precision, the higher the recall. Complementary, the MAP indicates whether the last relevant documents we retrieved were well ranked or not, so it somewhat reflects another interest point: the DCV where recall equals 1.

5.4 Statistical tests

Since the results can be very close from one weighting scheme to another, we tested the statistical significance of our results. We used the classical non-parametric Wilcoxon test with a p-value threshold set to 0.1. In the result tables, we show:

- in bold and underlined: the best result(s) for the given evaluation measure and weighting schemes;
- in bold only: every result whose difference from the best one (of the table) is not statistically significant.

5.5 Vocabulary construction

Building an optimal vocabulary is an awkward issue because it depends both on the dataset considered and on the different parameters of the vocabulary building process. Moreover, it generally requires a great computation time, as clustering descriptors is computationally very expensive and has to be repeated several times to get the best possible. However, we can reasonnably make the hypothesis that, in the case of a comparative study, finding the best vocabulary possible is not absolutely necessary: as long as the given vocabulary is representative of the common ways of obtaining such vocabularies, we can consider that the observed results are also valid with comparable vocabularies. So we used a common method to build our vocabularies. They may not be the best, but the observed differences in the results should be the same with other vocabularies.

Key point detector: We used the Hessian-affine key point detector, as it is one of the best [20] and is commonly used in recent visual word-based papers [15, 7].

Interest region descriptor: We used the SIFT descriptors. They yield some of the best results for image matching [21] and are clearly the most used in recent litterature [28, 8, 15, 7].

Clustering algorithm: We used the hierarchical k-means algorithm proposed by Nister and Stewenius [22]. It is not the best one in terms of retrieval performance but it allows to work very efficiently with an acceptable accuracy. Table 8 shows the vocabulary sizes we used for each dataset.

Dataset	Vocabulary size
Caltech-6	6,556
Caltech-101	61,687
Nister	19,545
Oxford	117,151

Table 8: Vocabulary size for each dataset

5.6 Distance experiments

We considered the general form of the Minkowski distance Lk for vector spaces (Equation 7), and studied the impact of distance parameter k on the effectiveness of image retrieval. We tested several values of k from 0.01 to 3 with a standard TF*IDF ($l1g1$ in Tables 2 and 3) weighting scheme. Figure 7 shows the results for each dataset and several distance measures. First, we must make a methodology remark: in the case of the Nister dataset, precision measures with high DCVs are not very significant because there are only 4 relevant documents for each query; considering MAP is therefore more interesting. Then we can make some remarks about these results:

- on Nister and Caltech6 datasets, the results clearly show that the greater the value of k, the worse the results; optimal results are obtained when k value is around 0.75. For smaller values, the performance decreases. These results are commented in Section 6.1;
- on Caltech101, the difference is much less significative, although there is a little improvement when k decreases; this result is also discussed in Section 6.1;
- on Oxford, on the contrary, smaller values of k yield poor results, and best results are obtained when $k \approx 2$. This last result is discussed in Section 6.5.

$$d_{Lk}(d,q) = (\sum_i |d_i - q_i|^k)^{\frac{1}{k}} \tag{7}$$

5.7 Weighting experiments

Results of the weighting experiments are shown on Tables 9 to 28. They are organized as follows:

- Tables 9 to 16 show the results of the weighting schemes obtained by any combination of a local weight from Table 2 and a global weight from Table 3. The combination of local weight l_x and global weight g_y is refered as $l_x g_y$. Each of these weighting schemes has been tested with the four datasets and the two following distances: L1 and L2. For each of these distances, the vectors have been previously normalized with the appropriate norm, as stated in Section 4.1.3;
- Tables 17 to 20 show the results obtained when using DFR matching measures as they are originally defined in [2]. Each DFR matching measure is refered as XYZ, where X corresponds to a randomness model (see Table 4), Y to a information gain model (Table 4.2.3) and Z to a length normalization model (Table 6);
- Tables 21 to 28 contain the results we obtained using DFR-based weights. DFR-based weighting schemes are refered in the same way as DFR matching measures. DFR-based weighting schemes were tested with each dataset and distances L1 and L2 (after an appropriate normalization of the vectors);

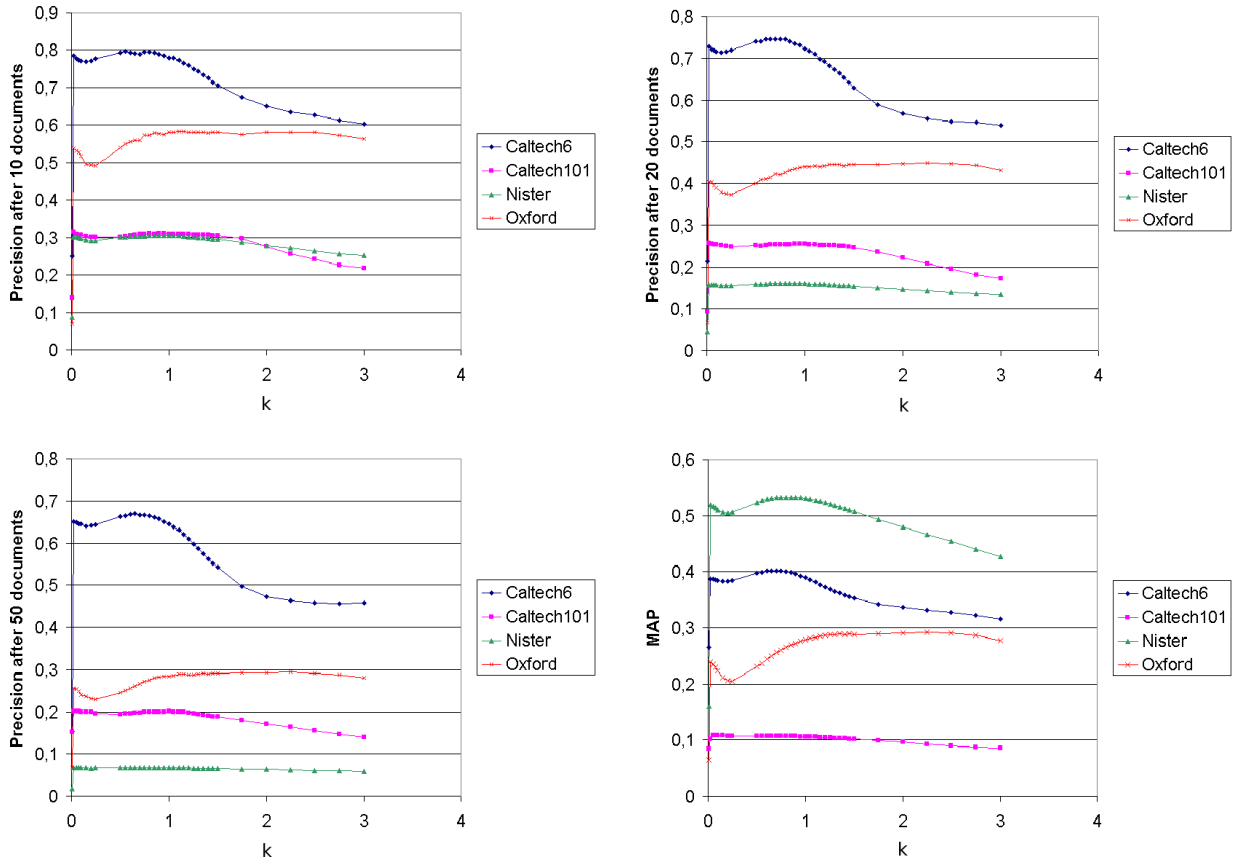


Figure 7: Influence of k on Lk distance's effectiveness

- Figures 8 to 11 give an overview of the results for each weighting scheme and the two distances L1 and L2. The performance measures used in these figures are precision after 10 retrieved documents and MAP. The weighting schemes are in the same order as in the result tables.

For each table, the best result(s) is (are) shown in bold and underlined. The other results in bold are all the results that are not statistically different from the best one(s), according to Wilcoxon's statistical test (Section 5.4). We can make two major remarks about these results:

- **The effectiveness difference from one weighting scheme to another is generally small, and not always significant.** Table 29 shows the best performance improvement obtained compared to the standard weighting scheme $l1g0$ (*i.e.* no specific weighting) when using the L1 distance. The performance is less than 10 percent better than the baseline, except in the case of the Oxford dataset. Moreover, the performance gain is particulary small on the most difficult of our datasets, Caltech101.

- **There is no weighting scheme that performs better than the others on every dataset.** For each dataset and (quite) each performance measure, the best result is given by a different weighting scheme. We therefore cannot expect one weighting scheme to perform better than another on new datasets.

We discuss the (lack of) effect of local weights in Section 6.2 and the one of global weights in Section 6.3. The particular case of the Oxford dataset will be discussed in Section 6.5.

About DFR weights and measures, two things can be noticed:

- the difference between the results using DFR matching scores (corresponding to the original DFR matching measure proposed in [2]) and DFR weights with L2 distance (equivalent to the cosine similarity, see Section 4.1.4) seems to show that queries should not be weighted, contrary to what we stated in Section 3.4. However, this is due to the fact that L2, which is equivalent to the cosine similarity, computed as products of the query's weights and the document's weights. When using other Minkowski-like distances, that rely on weight differences, we checked on a few experiments that query normalization is mandatory, as weights have to stand in comparable ranges;

- Results obtained using DFR weights are similar to the one obtained with standard weights. Moreover, they seem generally equivalent and the difference between the best DFR weights and the others is most of the time not significant. These results are discussed in Section 6.4.

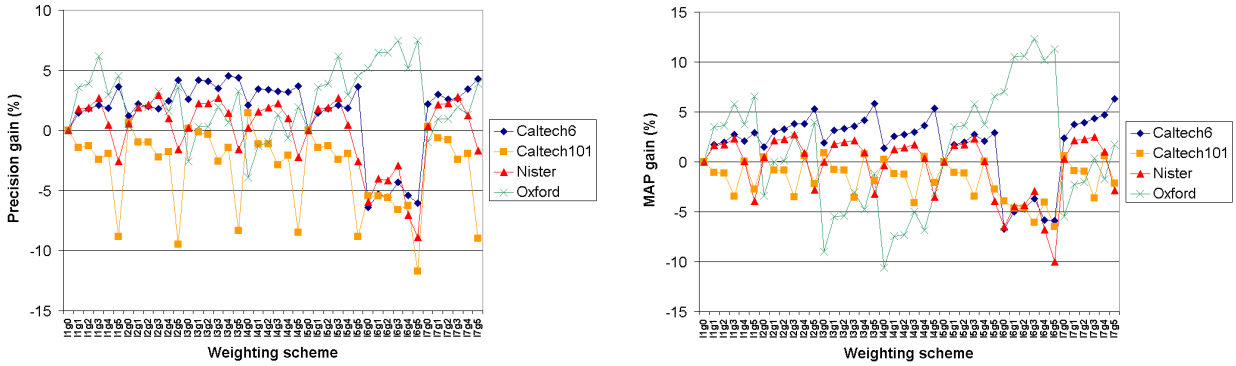


Figure 8: Results using L1 distance and the weighting schemes from Tables 2 and 3

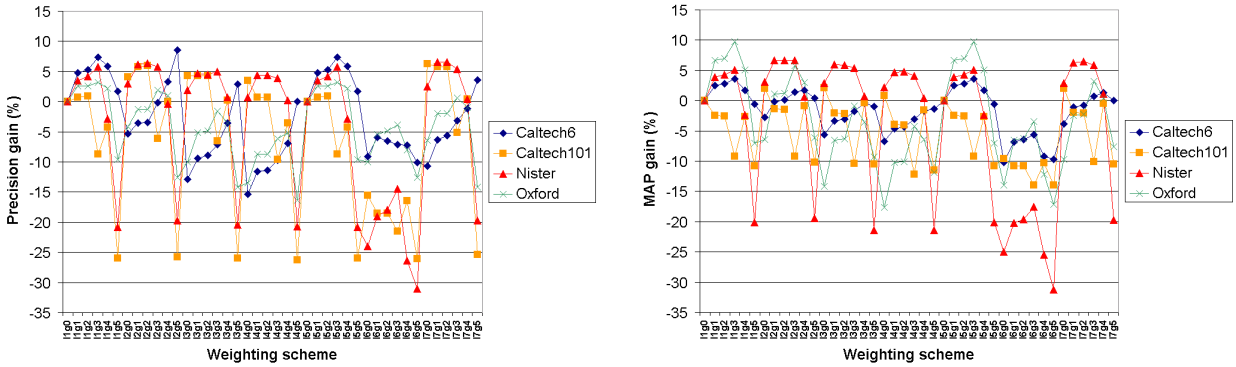


Figure 9: Results using L2 distance and the weighting schemes from Tables 2 and 3

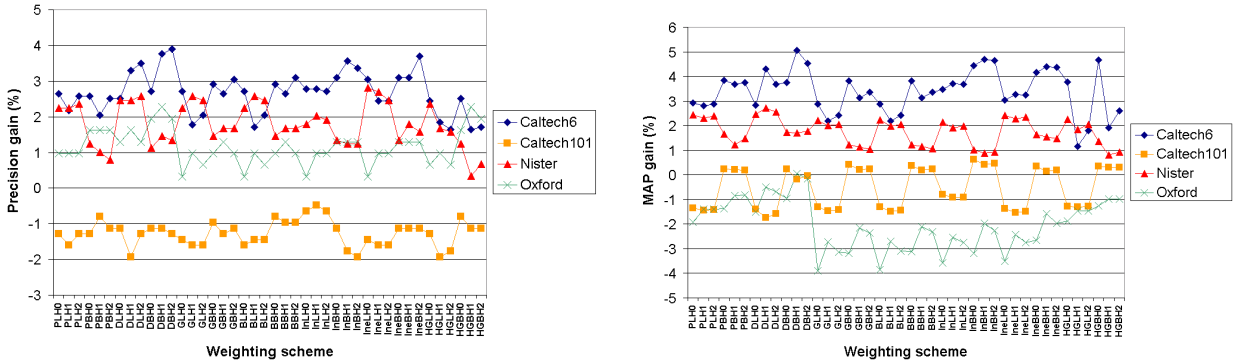


Figure 10: Results using L1 distance and the DFR weighting schemes

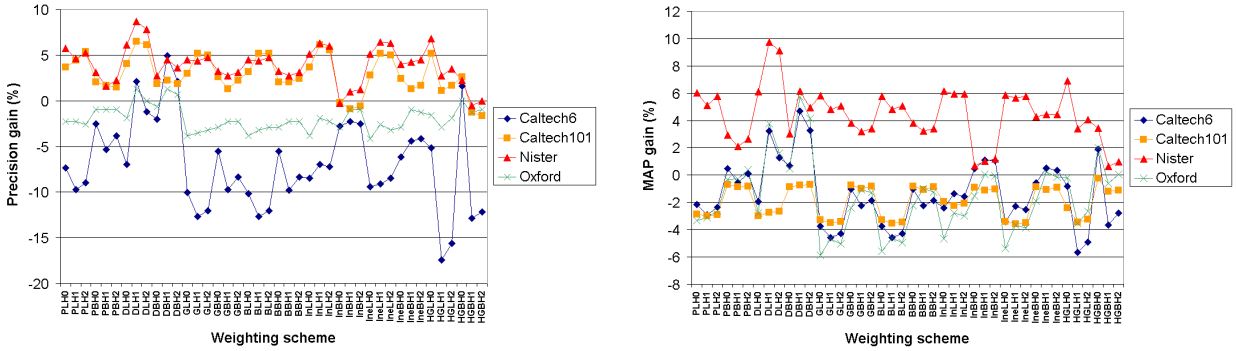


Figure 11: Results using L2 distance and the DFR weighting schemes

	P@10	P@20	P@50	P@100	MAP
l1g0	0.7685	0.7102	0.6318	0.5712	0.3827
l1g1	0.7797	0.7228	0.6445	0.5833	0.3893
l1g2	0.7822	0.7261	0.6458	0.5855	0.3903
l1g3	0.7848	0.7284	0.6538	0.5919	0.3932
l1g4	0.7827	0.7244	0.6481	0.5867	0.3908
l1g5	**0.7964**	0.7332	0.6455	0.5882	0.3938
l2g0	0.7777	0.7211	0.6426	0.5823	0.3884
l2g1	0.7853	0.7315	0.6556	0.5945	0.3943
l2g2	0.7838	0.7345	0.6576	0.5973	0.3951
l2g3	0.7822	0.7353	0.6617	0.6005	0.3972
l2g4	0.7873	0.7419	0.6605	0.6004	0.3973
l2g5	**0.8005**	0.7434	0.6692	0.6104	0.4030
l3g0	0.7883	0.7332	0.6562	0.5911	0.3900
l3g1	**0.8005**	0.7431	0.6681	0.5994	0.3948
l3g2	**0.8000**	0.7434	0.6677	0.5995	0.3954
l3g3	**0.7954**	0.7442	0.6718	0.6022	0.3965
l3g4	**0.8036**	0.7444	0.6740	0.6065	0.3986
l3g5	**0.8020**	**0.7548**	**0.6825**	**0.6193**	**0.4049**
l4g0	0.7848	0.7282	0.6522	0.5896	0.3880
l4g1	**0.7949**	0.7363	0.6642	0.5957	0.3925
l4g2	**0.7944**	0.7373	0.6659	0.5975	0.3931
l4g3	**0.7934**	0.7409	0.6658	0.5995	0.3940
l4g4	0.7929	0.7416	0.6695	0.6051	0.3966
l4g5	**0.7970**	0.7482	**0.6818**	**0.6174**	0.4032
l5g0	0.7685	0.7102	0.6318	0.5712	0.3827
l5g1	0.7797	0.7228	0.6445	0.5833	0.3893
l5g2	0.7822	0.7261	0.6458	0.5855	0.3903
l5g3	0.7848	0.7284	0.6538	0.5919	0.3932
l5g4	0.7827	0.7244	0.6481	0.5867	0.3908
l5g5	**0.7964**	0.7332	0.6455	0.5882	0.3938
l6g0	0.7193	0.6503	0.5735	0.5128	0.3569
l6g1	0.7279	0.6589	0.5866	0.5262	0.3636
l6g2	0.7259	0.6657	0.5897	0.5284	0.3648
l6g3	0.7355	0.6739	0.5948	0.5338	0.3685
l6g4	0.7269	0.6622	0.5807	0.5199	0.3603
l6g5	0.7218	0.6596	0.5791	0.5219	0.3601
l7g0	0.7853	0.7254	0.6519	0.5898	0.3917
l7g1	0.7914	0.7406	0.6641	0.6003	0.3971
l7g2	0.7883	0.7406	0.6660	0.6008	0.3978
l7g3	0.7888	0.7393	0.6688	0.6038	0.3993
l7g4	**0.7949**	0.7444	0.6712	0.6065	0.4007
l7g5	**0.8015**	**0.7536**	0.6791	**0.6198**	**0.4068**

Table 9: Performance of Tables 2 and 3 weighting schemes with L1 distance on Caltech6

	P@10	P@20	P@50	P@100	MAP
l1g0	0.6208	0.5518	0.4678	0.4150	**0.3287**
l1g1	0.6503	0.5680	0.4728	0.4212	**0.3368**
l1g2	**0.6533**	0.5695	0.4739	0.4224	**0.3378**
l1g3	**0.6665**	**0.5832**	0.4823	0.4259	**0.3406**
l1g4	**0.6574**	**0.5797**	**0.4877**	**0.4316**	**0.3342**
l1g5	0.6315	**0.5685**	**0.4931**	**0.4387**	**0.3268**
l2g0	0.5878	0.4959	0.4066	0.3644	0.3195
l2g1	0.5985	0.5094	0.4189	0.3709	0.3280
l2g2	0.5995	0.5124	0.4214	0.3722	0.3291
l2g3	0.6198	0.5332	0.4380	0.3853	0.3335
l2g4	0.6411	0.5574	0.4535	0.4041	0.3343
l2g5	**0.6741**	**0.5906**	**0.4902**	0.4276	**0.3301**
l3g0	0.5406	0.4614	0.3678	0.3275	0.3101
l3g1	0.5624	0.4751	0.3755	0.3339	0.3176
l3g2	0.5655	0.4723	0.3787	0.3349	0.3185
l3g3	0.5766	0.4848	0.3974	0.3514	0.3228
l3g4	0.5985	0.5140	0.4209	0.3708	0.3275
l3g5	0.6386	0.5627	0.4743	0.4066	0.3253
l4g0	0.5254	0.4447	0.3540	0.3137	0.3064
l4g1	0.5487	0.4538	0.3616	0.3213	0.3135
l4g2	0.5503	0.4548	0.3608	0.3233	0.3143
l4g3	0.5604	0.4690	0.3828	0.3380	0.3185
l4g4	0.5777	0.4952	0.4028	0.3552	0.3233
l4g5	0.6208	0.5541	0.4640	0.4037	0.3241
l5g0	0.6208	0.5518	0.4678	0.4150	**0.3287**
l5g1	0.6503	0.5680	0.4728	0.4212	**0.3368**
l5g2	**0.6533**	0.5695	0.4739	0.4224	**0.3378**
l5g3	**0.6665**	**0.5832**	0.4823	0.4259	**0.3406**
l5g4	**0.6574**	**0.5797**	**0.4877**	**0.4316**	**0.3342**
l5g5	0.6315	**0.5685**	**0.4931**	**0.4387**	**0.3268**
l6g0	0.5645	0.5175	0.4636	0.4171	0.2952
l6g1	0.5838	0.5302	0.4776	**0.4358**	**0.3062**
l6g2	0.5802	0.5289	0.4770	**0.4371**	**0.3077**
l6g3	0.5766	0.5312	0.4755	**0.4392**	0.3102
l6g4	0.5761	0.5140	0.4645	**0.4233**	**0.2985**
l6g5	0.5579	0.4997	0.4496	0.4133	**0.2967**
l7g0	0.5543	0.4744	0.3870	0.3463	0.3161
l7g1	0.5812	0.4952	0.4038	0.3568	0.3251
l7g2	0.5858	0.4982	0.4063	0.3594	0.3263
l7g3	0.6010	0.5114	0.4242	0.3747	0.3309
l7g4	0.6137	0.5299	0.4348	0.3878	0.3331
l7g5	0.6431	0.5718	0.4759	0.4129	**0.3287**

Table 10: Performance of Tables 2 and 3 weighting schemes with L2 distance on Caltech6

	P@10	P@20	P@50	P@100	MAP
l1g0	**0.3146**	**0.2596**	**0.2030**	0.1656	0.1071
l1g1	**0.3101**	0.2553	**0.2003**	0.1597	**0.1060**
l1g2	**0.3106**	0.2553	**0.2003**	0.1597	**0.1060**
l1g3	**0.3071**	0.2457	0.1910	0.1541	0.1034
l1g4	**0.3086**	0.2535	0.1982	**0.1644**	**0.1072**
l1g5	0.2869	0.2396	0.1864	0.1577	0.1042
l2g0	**0.3167**	0.2604	**0.2048**	**0.1667**	**0.1077**
l2g1	**0.3116**	0.2548	**0.2008**	0.1609	**0.1063**
l2g2	**0.3116**	0.2548	**0.2005**	0.1607	**0.1062**
l2g3	**0.3076**	0.2465	0.1917	0.1547	0.1034
l2g4	**0.3091**	**0.2571**	0.2006	**0.1651**	**0.1078**
l2g5	0.2848	0.2409	0.1889	0.1584	0.1048
l3g0	**0.3152**	**0.2624**	**0.2040**	**0.1663**	**0.1081**
l3g1	**0.3141**	0.2545	0.1982	0.1604	0.1063
l3g2	**0.3136**	0.2538	0.1983	0.1603	0.1062
l3g3	**0.3066**	0.2439	0.1919	0.1539	0.1033
l3g4	**0.3101**	**0.2561**	0.1997	**0.1640**	**0.1080**
l3g5	0.2884	0.2422	0.1888	0.1580	0.1051
l4g0	**0.3192**	**0.2609**	0.2009	**0.1660**	**0.1074**
l4g1	**0.3111**	0.2528	0.1976	0.1587	0.1059
l4g2	**0.3111**	0.2525	0.1975	0.1586	0.1058
l4g3	0.3056	0.2427	0.1911	0.1529	0.1027
l4g4	0.3081	0.2558	0.1978	**0.1627**	**0.1077**
l4g5	0.2879	0.2417	0.1881	0.1569	0.1049
l5g0	**0.3146**	**0.2596**	**0.2030**	0.1656	0.1071
l5g1	**0.3101**	0.2553	**0.2003**	0.1597	**0.1060**
l5g2	**0.3106**	0.2553	**0.2003**	0.1597	0.1060
l5g3	**0.3071**	0.2457	0.1910	0.1541	0.1034
l5g4	**0.3086**	0.2535	0.1982	**0.1644**	**0.1072**
l5g5	0.2869	0.2396	0.1864	0.1577	0.1042
l6g0	0.2975	0.2371	0.1887	0.1558	0.1030
l6g1	0.2975	0.2369	0.1869	0.1541	0.1022
l6g2	0.2970	0.2369	0.1873	0.1540	0.1022
l6g3	0.2939	0.2348	0.1812	0.1501	0.1006
l6g4	0.2949	0.2356	0.1862	0.1556	0.1028
l6g5	0.2778	0.2237	0.1754	0.1513	0.1002
l7g0	0.3157	0.2614	0.2037	0.1661	0.1078
l7g1	**0.3126**	0.2538	0.1990	**0.1604**	0.1062
l7g2	**0.3121**	0.2540	0.1989	**0.1603**	0.1061
l7g3	**0.3071**	0.2434	0.1908	0.1542	0.1033
l7g4	**0.3086**	**0.2578**	0.1989	**0.1643**	**0.1078**
l7g5	0.2864	0.2427	0.1886	0.1581	0.1048

Table 11: Performance of Tables 2 and 3 weighting schemes with L1 distance on Caltech101

	P@10	P@20	P@50	P@100	MAP
l1g0	0.2727	0.2237	**0.1823**	**0.1524**	0.0985
l1g1	0.2747	0.2222	0.1713	0.1426	0.0961
l1g2	0.2753	0.2227	0.1709	0.1422	0.0960
l1g3	0.2490	0.1985	0.1483	0.1239	0.0895
l1g4	0.2611	0.2091	0.1697	0.1435	0.0959
l1g5	0.2020	0.1636	0.1374	0.1228	0.0879
l2g0	0.2838	0.2336	**0.1848**	**0.1542**	**0.1005**
l2g1	0.2884	0.2280	0.1729	0.1428	0.0971
l2g2	0.2889	0.2278	**0.1731**	0.1425	0.0971
l2g3	0.2561	0.1972	0.1488	0.1204	0.0894
l2g4	0.2727	0.2210	0.1710	0.1459	0.0977
l2g5	0.2025	0.1636	0.1369	0.1216	0.0885
l3g0	0.2843	0.2333	**0.1837**	**0.1533**	**0.1006**
l3g1	0.2843	0.2283	**0.1708**	0.1407	0.0964
l3g2	0.2843	0.2278	**0.1705**	0.1404	0.0964
l3g3	0.2551	0.1912	0.1435	0.1155	0.0883
l3g4	0.2732	0.2210	0.1717	0.1444	0.0981
l3g5	0.2020	0.1654	0.1346	0.1194	0.0882
l4g0	0.2823	0.2273	**0.1819**	0.1501	0.0993
l4g1	0.2747	0.2182	0.1674	0.1366	0.0946
l4g2	0.2747	0.2179	0.1672	0.1363	0.0945
l4g3	0.2465	0.1866	0.1406	0.1110	0.0866
l4g4	0.2631	0.2136	0.1683	0.1415	0.0970
l4g5	0.2010	0.1626	0.1312	0.1165	0.0872
l5g0	0.2727	0.2237	**0.1823**	**0.1524**	0.0985
l5g1	0.2747	0.2222	0.1713	0.1426	0.0961
l5g2	0.2753	0.2227	0.1709	0.1422	0.0960
l5g3	0.2490	0.1985	0.1483	0.1239	0.0895
l5g4	0.2611	0.2091	0.1697	0.1435	0.0959
l5g5	0.2020	0.1636	0.1374	0.1228	0.0879
l6g0	0.2303	0.1871	0.1476	0.1294	0.0891
l6g1	0.2222	0.1780	0.1443	0.1261	0.0879
l6g2	0.2222	0.1780	0.1446	0.1262	0.0879
l6g3	0.2141	0.1679	0.1333	0.1156	0.0848
l6g4	0.2278	0.1790	0.1445	0.1262	0.0884
l6g5	0.2015	0.1578	0.1320	0.1173	0.0848
l7g0	**0.2899**	**0.2356**	0.1835	0.1526	**0.1005**
l7g1	0.2884	0.2273	**0.1709**	0.1406	0.0966
l7g2	0.2884	0.2275	0.1710	0.1403	0.0965
l7g3	0.2586	0.1929	0.1442	0.1162	0.0885
l7g4	0.2737	0.2227	0.1730	0.1449	0.0980
l7g5	0.2035	0.1636	0.1352	0.1206	0.0882

Table 12: Performance of Tables 2 and 3 weighting schemes with L2 distance on Caltech-101

	P@4	MAP
l1g0	0.6970	0.5229
l1g1	0.7104	0.5316
l1g2	0.7113	0.5318
l1g3	0.7121	**0.5349**
l1g4	0.6970	0.5232
l1g5	0.6540	0.5024
l2g0	0.7029	0.5254
l2g1	**0.7180**	**0.5341**
l2g2	**0.7180**	**0.5347**
l2g3	**0.7197**	**0.5371**
l2g4	0.7045	0.5277
l2g5	0.6658	0.5083
l3g0	0.6886	0.5228
l3g1	0.7088	**0.5324**
l3g2	0.7096	**0.5332**
l3g3	**0.7172**	**0.5340**
l3g4	0.6995	0.5277
l3g5	0.6582	0.5060
l4g0	0.6852	0.5209
l4g1	0.6995	**0.5294**
l4g2	0.6995	**0.5303**
l4g3	0.7104	**0.5320**
l4g4	0.6886	0.5249
l4g5	0.6591	0.5044
l5g0	0.6970	0.5229
l5g1	0.7104	0.5316
l5g2	0.7113	0.5318
l5g3	0.7121	**0.5349**
l5g4	0.6970	0.5232
l5g5	0.6540	0.5024
l6g0	0.6279	0.4891
l6g1	0.6557	0.4996
l6g2	0.6566	0.5003
l6g3	0.6675	0.5075
l6g4	0.6355	0.4875
l6g5	0.6103	0.4705
l7g0	0.6995	0.5243
l7g1	**0.7146**	**0.5340**
l7g2	**0.7138**	**0.5347**
l7g3	**0.7180**	**0.5361**
l7g4	0.7029	0.5280
l7g5	0.6675	0.5079

Table 13: Performance of Tables 2 and 3 weighting schemes with L1 distance on Nister

	P@4	MAP
l1g0	0.5985	0.4611
l1g1	0.6254	0.4792
l1g2	0.6305	**0.4811**
l1g3	**0.6414**	**0.4847**
l1g4	0.5766	0.4502
l1g5	0.4579	0.3682
l2g0	0.6271	0.4753
l2g1	**0.6507**	0.4918
l2g2	**0.6524**	**0.4920**
l2g3	**0.6549**	**0.4917**
l2g4	0.6019	0.4641
l2g5	0.4731	0.3715
l3g0	0.6313	0.4744
l3g1	**0.6557**	**0.4884**
l3g2	**0.6557**	0.4884
l3g3	**0.6524**	0.4861
l3g4	0.6103	0.4644
l3g5	0.4638	0.3626
l4g0	0.6296	0.4714
l4g1	0.6481	0.4829
l4g2	0.6490	0.4833
l4g3	0.6397	0.4798
l4g4	0.6086	0.4632
l4g5	0.4714	0.3625
l5g0	0.5985	0.4611
l5g1	0.6254	0.4792
l5g2	0.6305	**0.4811**
l5g3	**0.6414**	**0.4847**
l5g4	0.5766	0.4502
l5g5	0.4579	0.3682
l6g0	0.4184	0.3460
l6g1	0.4428	0.3678
l6g2	0.4461	0.3705
l6g3	0.4604	0.3801
l6g4	0.4125	0.3436
l6g5	0.3763	0.3169
l7g0	0.6288	0.4746
l7g1	**0.6498**	**0.4899**
l7g2	**0.6549**	**0.4910**
l7g3	**0.6498**	**0.4881**
l7g4	0.6103	0.4665
l7g5	0.4739	0.3704

Table 14: Performance of Tables 2 and 3 weighting schemes with L2 distance on Nister

	P@5	P@10	P@20	P@50	MAP
l1g0	0.7200	0.5600	0.4218	0.2764	0.2698
l1g1	0.7200	0.5800	0.4400	0.2844	0.2792
l1g2	0.7200	0.5818	0.4400	0.2855	0.2795
l1g3	**0.7236**	**0.5945**	**0.4518**	0.2920	0.2853
l1g4	0.7164	0.5764	0.4327	0.2855	0.2798
l1g5	**0.7273**	0.5855	**0.4464**	0.2887	0.2874
l2g0	0.7091	0.5618	0.4255	0.2691	0.2606
l2g1	0.7127	0.5709	0.4327	0.2804	0.2697
l2g2	0.7127	0.5709	0.4327	0.2804	0.2701
l2g3	**0.7236**	0.5782	0.4418	0.2836	0.2769
l2g4	0.7127	0.5691	0.4345	0.2756	0.2705
l2g5	0.7055	0.5800	**0.4464**	0.2855	0.2803
l3g0	0.6945	0.5455	0.4082	0.2604	0.2455
l3g1	0.7018	0.5618	0.4182	0.2695	0.2549
l3g2	0.7018	0.5618	0.4191	0.2698	0.2552
l3g3	0.7055	0.5709	0.4264	0.2756	0.2615
l3g4	0.7018	0.5636	0.4236	0.2691	0.2570
l3g5	**0.7055**	0.5782	0.4300	0.2760	0.2665
l4g0	0.6800	0.5382	0.4064	0.2585	0.2412
l4g1	0.6873	0.5527	0.4136	0.2662	0.2497
l4g2	0.6873	0.5545	0.4136	0.2669	0.2500
l4g3	0.6982	0.5673	0.4236	0.2724	0.2563
l4g4	0.6909	0.5564	0.4155	0.2665	0.2513
l4g5	0.7018	0.5709	0.4273	0.2727	0.2615
l5g0	0.7200	0.5600	0.4218	0.2764	0.2698
l5g1	0.7200	0.5800	0.4400	0.2844	0.2792
l5g2	0.7200	0.5818	0.4400	0.2855	0.2795
l5g3	**0.7236**	**0.5945**	**0.4518**	0.2920	0.2853
l5g4	0.7164	0.5764	0.4327	0.2855	0.2798
l5g5	**0.7273**	0.5855	**0.4464**	0.2887	0.2874
l6g0	0.7309	0.5891	0.4455	0.2865	0.2886
l6g1	**0.7418**	**0.5964**	**0.4527**	0.2949	0.2981
l6g2	**0.7382**	**0.5964**	**0.4536**	0.2956	0.2983
l6g3	**0.7382**	**0.6018**	**0.4609**	**0.3040**	**0.3030**
l6g4	**0.7345**	0.5891	**0.4527**	**0.2949**	0.2973
l6g5	**0.7236**	**0.6018**	**0.4609**	**0.2964**	**0.3003**
l7g0	0.7018	0.5545	0.4182	0.2640	0.2550
l7g1	0.7127	0.5655	0.4282	0.2738	0.2636
l7g2	**0.7164**	0.5655	0.4300	0.2745	0.2643
l7g3	**0.7236**	0.5709	0.4345	0.2782	0.2707
l7g4	**0.7200**	0.5673	0.4291	0.2735	0.2651
l7g5	0.7055	0.5818	0.4382	0.2804	0.2743

Table 15: Performance of Tables 2 and 3 weighting schemes with L1 distance on Oxford

	P@5	P@10	P@20	P@50	MAP
l1g0	0.6800	**0.5655**	0.4209	0.2782	0.2730
l1g1	**0.7055**	**0.5800**	0.4482	**0.2931**	0.2911
l1g2	**0.7055**	**0.5800**	0.4482	**0.2938**	0.2921
l1g3	**0.7018**	**0.5836**	**0.4600**	**0.2982**	**0.2997**
l1g4	0.6836	**0.5782**	**0.4491**	0.2905	0.2868
l1g5	0.5855	0.5109	0.4027	0.2644	0.2539
l2g0	0.6655	0.5418	0.4091	0.2622	0.2555
l2g1	0.6909	0.5582	0.4300	0.2778	0.2760
l2g2	0.6909	0.5582	0.4300	0.2782	0.2764
l2g3	**0.6982**	**0.5764**	0.4464	0.2825	0.2888
l2g4	0.6764	**0.5709**	0.4373	0.2822	0.2811
l2g5	0.6109	0.4945	0.4018	0.2625	0.2562
l3g0	0.6291	0.5091	0.3945	0.2531	0.2343
l3g1	0.6509	0.5364	0.4127	0.2640	0.2553
l3g2	0.6509	0.5382	0.4136	0.2640	0.2558
l3g3	0.6691	0.5564	0.4300	0.2702	0.2706
l3g4	0.6618	0.5455	0.4227	0.2716	0.2634
l3g5	0.5855	0.4855	0.3927	0.2549	0.2477
l4g0	0.6109	0.4891	0.3809	0.2455	0.2248
l4g1	0.6400	0.5164	0.4045	0.2589	0.2451
l4g2	0.6400	0.5164	0.4045	0.2596	0.2455
l4g3	0.6618	0.5309	0.4236	0.2662	0.2614
l4g4	0.6364	0.5364	0.4191	0.2669	0.2555
l4g5	0.5818	0.4727	0.3845	0.2484	0.2405
l5g0	0.6800	**0.5655**	0.4209	0.2782	0.2730
l5g1	**0.7055**	**0.5800**	0.4482	**0.2931**	0.2911
l5g2	**0.7055**	**0.5800**	0.4482	**0.2938**	0.2921
l5g3	**0.7018**	**0.5836**	**0.4600**	**0.2982**	**0.2997**
l5g4	0.6836	**0.5782**	**0.4491**	0.2905	0.2868
l5g5	0.5855	0.5109	0.4027	0.2644	0.2539
l6g0	0.6145	0.5091	0.3791	0.2480	0.2348
l6g1	0.6291	0.5345	0.4073	0.2640	0.2551
l6g2	0.6291	0.5382	0.4100	0.2647	0.2559
l6g3	0.6291	0.5436	0.4282	0.2738	0.2636
l6g4	0.6073	0.5200	0.3955	0.2531	0.2397
l6g5	0.5782	0.4945	0.3745	0.2436	0.2261
l7g0	0.6473	0.5291	0.4000	0.2593	0.2466
l7g1	0.6618	0.5545	0.4236	0.2691	0.2664
l7g2	0.6618	0.5545	0.4236	0.2687	0.2670
l7g3	0.6836	**0.5691**	0.4455	0.2756	0.2818
l7g4	0.6764	0.5600	0.4327	0.2796	0.2742
l7g5	0.6073	0.4855	0.3982	0.2593	0.2521

Table 16: Performance of Tables 2 and 3 weighting schemes with L2 distance on Oxford

	P@5	P@10	P@20	P@100	MAP
PLH0	0.4020	0.3376	0.2845	0.2264	0.2832
PLH1	0.6640	0.5812	0.5008	0.3852	0.3282
PLH2	0.6081	0.5061	0.4350	0.3256	0.3136
PBH0	0.4051	0.3396	0.2914	0.2351	0.2849
PBH1	0.6792	0.5934	0.5155	0.3985	0.3291
PBH2	0.6193	0.5183	0.4462	0.3360	0.3151
DLH0	0.4020	0.3365	0.2848	0.2265	0.2832
DLH1	**0.8112**	**0.7538**	**0.6805**	**0.5409**	**0.3583**
DLH2	0.7249	0.6218	0.5576	0.4217	0.3353
DBH0	0.4051	0.3396	0.2911	0.2354	0.2849
DBH1	**0.8010**	**0.7431**	**0.6744**	**0.5386**	**0.3547**
DBH2	0.7228	0.6289	0.5599	0.4301	0.3347
GLH0	0.4122	0.3299	0.2843	0.2258	0.2836
GLH1	0.5980	0.5056	0.4330	0.3175	0.3119
GLH2	0.5553	0.4563	0.3924	0.2860	0.3032
GBH0	0.4162	0.3401	0.2942	0.2320	0.2851
GBH1	0.6000	0.5091	0.4393	0.3256	0.3128
GBH2	0.5543	0.4629	0.4003	0.2938	0.3046
BLH0	0.4122	0.3299	0.2843	0.2257	0.2836
BLH1	0.5980	0.5056	0.4330	0.3177	0.3119
BLH2	0.5553	0.4569	0.3924	0.2860	0.3032
BBH0	0.4162	0.3401	0.2942	0.2320	0.2851
BBH1	0.6010	0.5091	0.4393	0.3257	0.3128
BBH2	0.5543	0.4629	0.4005	0.2938	0.3047
InLH0	0.4112	0.3365	0.2878	0.2280	0.2844
InLH1	0.6782	0.5782	0.5048	0.3819	0.3268
InLH2	0.5929	0.4970	0.4325	0.3182	0.3119
InBH0	0.4162	0.3416	0.2947	0.2352	0.2860
InBH1	0.6853	0.5827	0.5183	0.3916	0.3270
InBH2	0.5959	0.5010	0.4378	0.3288	0.3129
IneLH0	0.4122	0.3305	0.2845	0.2265	0.2838
IneLH1	0.6772	0.5782	0.5008	0.3784	0.3265
IneLH2	0.5888	0.4964	0.4302	0.3148	0.3113
IneBH0	0.4162	0.3406	0.2954	0.2325	0.2855
IneBH1	0.6772	0.5848	0.5109	0.3881	0.3279
IneBH2	0.5929	0.4985	0.4388	0.3251	0.3133
HGLH0	0.6162	0.5117	0.4353	0.3196	0.3130
HGLH1	0.5025	0.4234	0.3977	0.3932	0.3097
HGLH2	0.4934	0.4127	0.3863	0.3769	0.3256
HGBH0	0.6294	0.5320	0.4490	0.3366	0.3155
HGBH1	0.5005	0.4254	0.4038	0.3928	0.3065
HGBH2	0.5015	0.4173	0.3893	0.3808	0.3225

Table 17: Performance of DFR matching measures on Caltech6

	P@4	MAP
PLH0	0.3409	0.2819
PLH1	0.6675	0.4981
PLH2	0.6094	0.4642
PBH0	0.3434	0.2808
PBH1	0.6338	0.4860
PBH2	0.5892	0.4542
DLH0	0.3392	0.2815
DLH1	**0.6953**	**0.5237**
DLH2	0.6465	0.4905
DBH0	0.3426	0.2799
DBH1	0.6625	0.5084
DBH2	0.6212	0.4788
GLH0	0.3577	0.2906
GLH1	0.6431	0.4823
GLH2	0.5951	0.4482
GBH0	0.3653	0.2962
GBH1	0.6263	0.4751
GBH2	0.5816	0.4438
BLH0	0.3577	0.2905
BLH1	0.6439	0.4830
BLH2	0.5960	0.4483
BBH0	0.3653	0.2961
BBH1	0.6263	0.4753
BBH2	0.5816	0.4439
InLH0	0.3603	0.2921
InLH1	0.6582	0.4900
InLH2	0.6052	0.4549
InBH0	0.3653	0.2964
InBH1	0.6397	0.4825
InBH2	0.5825	0.4476
IneLH0	0.3586	0.2907
IneLH1	0.6591	0.4918
IneLH2	0.6069	0.4566
IneBH0	0.3662	0.2960
IneBH1	0.6507	0.4872
IneBH2	0.5985	0.4525
HGLH0	0.5859	0.4449
HGLH1	0.6254	0.4825
HGLH2	0.6498	0.4989
HGBH0	0.5707	0.4391
HGBH1	0.5707	0.4484
HGBH2	0.6237	0.4799

Table 18: Performance of DFR matching measures on Nister

	P@10	P@20	P@50	P@100	MAP
PLH0	0.1631	0.1381	0.1252	0.1126	0.0853
PLH1	0.2586	0.2242	0.1800	0.1489	0.0953
PLH2	0.2677	0.2253	0.1804	0.1499	0.0967
PBH0	0.1687	0.1394	0.1283	0.1171	0.0869
PBH1	0.2551	0.2250	0.1819	0.1514	0.0965
PBH2	0.2662	0.2263	0.1813	0.1519	0.0978
DLH0	0.1636	0.1376	0.1253	0.1123	0.0853
DLH1	0.2884	**0.2389**	0.1826	**0.1498**	0.0999
DLH2	**0.3005**	**0.2399**	**0.1834**	**0.1510**	**0.1011**
DBH0	0.1682	0.1396	0.1285	0.1170	0.0869
DBH1	0.2894	**0.2381**	**0.1852**	**0.1519**	0.1012
DBH2	**0.2995**	**0.2391**	**0.1868**	**0.1531**	**0.1022**
GLH0	0.1662	0.1399	0.1254	0.1117	0.0852
GLH1	**0.3015**	**0.2412**	**0.1839**	**0.1513**	**0.1012**
GLH2	**0.3045**	**0.2379**	**0.1829**	**0.1516**	**0.1008**
GBH0	0.1682	0.1399	0.1290	0.1164	0.0869
GBH1	**0.2995**	**0.2437**	**0.1872**	**0.1542**	**0.1023**
GBH2	**0.3015**	**0.2386**	**0.1847**	**0.1528**	**0.1019**
BLH0	0.1662	0.1399	0.1254	0.1117	0.0852
BLH1	**0.3005**	**0.2409**	**0.1841**	**0.1513**	**0.1012**
BLH2	**0.3051**	**0.2374**	**0.1828**	**0.1516**	**0.1008**
BBH0	0.1682	0.1399	0.1291	0.1163	0.0869
BBH1	**0.3010**	**0.2434**	**0.1869**	**0.1543**	**0.1023**
BBH2	**0.3020**	**0.2386**	**0.1846**	**0.1528**	**0.1019**
InLH0	0.1677	0.1391	0.1269	0.1129	0.0857
InLH1	**0.3030**	**0.2427**	0.1848	**0.1525**	**0.1016**
InLH2	**0.3035**	**0.2384**	**0.1838**	**0.1525**	**0.1012**
InBH0	0.1682	0.1417	0.1296	0.1170	0.0873
InBH1	**0.2985**	0.2417	**0.1873**	**0.1544**	**0.1024**
InBH2	**0.2990**	0.2374	0.1853	0.1529	0.1020
IneLH0	0.1662	0.1402	0.1253	0.1118	0.0852
IneLH1	**0.3015**	**0.2412**	**0.1838**	**0.1513**	**0.1012**
IneLH2	**0.3051**	**0.2376**	**0.1828**	**0.1513**	**0.1007**
IneBH0	0.1682	0.1402	0.1290	0.1163	0.0869
IneBH1	**0.2995**	**0.2432**	**0.1869**	**0.1541**	**0.1023**
IneBH2	**0.3020**	**0.2389**	**0.1843**	**0.1527**	**0.1019**
HGLH0	0.2414	0.2038	0.1600	0.1349	0.0936
HGLH1	0.2581	0.2172	0.1705	0.1405	0.0954
HGLH2	0.2737	0.2313	0.1800	0.1488	0.0980
HGBH0	0.2404	0.2053	0.1621	0.1388	0.0949
HGBH1	0.2576	0.2167	0.1729	0.1442	0.0968
HGBH2	0.2763	0.2301	0.1828	0.1515	0.0992

Table 19: Performance of DFR matching measures on Caltech101

	P@5	P@10	P@20	P@50	MAP
PLH0	0.0633	0.0889	0.1163	0.1720	0.1450
PLH1	0.1292	0.1801	0.2368	0.3073	0.2992
PLH2	0.1231	0.1689	0.2112	0.2825	0.2718
PBH0	0.0658	0.0905	0.1172	0.1749	0.1501
PBH1	0.1282	0.1747	0.2359	0.3066	0.2966
PBH2	0.1205	0.1659	0.2136	0.2785	0.2726
DLH0	0.0633	0.0889	0.1157	0.1720	0.1451
DLH1	**0.1343**	**0.1905**	**0.2511**	**0.3178**	**0.3165**
DLH2	0.1286	0.1787	0.2243	0.3028	0.2936
DBH0	0.0658	0.0908	0.1163	0.1746	0.1507
DBH1	**0.1299**	**0.1886**	0.2437	0.3139	0.3075
DBH2	0.1279	0.1739	0.2242	0.2983	0.2892
GLH0	0.0655	0.0872	0.1127	0.1712	0.1446
GLH1	0.1303	0.1774	0.2223	0.2935	0.2896
GLH2	0.1201	0.1587	0.2034	0.2623	0.2567
GBH0	0.0658	0.0906	0.1188	0.1759	0.1499
GBH1	0.1296	0.1728	0.2204	0.2952	0.2894
GBH2	0.1189	0.1633	0.2030	0.2670	0.2603
BLH0	0.0655	0.0872	0.1127	0.1712	0.1446
BLH1	0.1303	0.1774	0.2223	0.2935	0.2898
BLH2	0.1201	0.1587	0.2040	0.2632	0.2570
BBH0	0.0658	0.0908	0.1188	0.1759	0.1499
BBH1	0.1296	0.1744	0.2221	0.2952	0.2897
BBH2	0.1189	0.1633	0.2030	0.2670	0.2605
InLH0	0.0655	0.0868	0.1159	0.1721	0.1460
InLH1	**0.1326**	0.1790	0.2231	0.3001	0.2931
InLH2	0.1214	0.1638	0.2070	0.2689	0.2618
InBH0	0.0658	0.0909	0.1182	0.1763	0.1512
InBH1	0.1310	0.1742	0.2252	0.2952	0.2910
InBH2	0.1203	0.1655	0.2079	0.2692	0.2648
IneLH0	0.0655	0.0873	0.1127	0.1712	0.1450
IneLH1	**0.1326**	0.1793	0.2249	0.2987	0.2940
IneLH2	0.1220	0.1626	0.2050	0.2673	0.2615
IneBH0	0.0658	0.0908	0.1185	0.1767	0.1503
IneBH1	0.1310	0.1758	0.2247	0.3011	0.2930
IneBH2	0.1217	0.1638	0.2056	0.2702	0.2650
HGLH0	0.1003	0.1391	0.1857	0.2402	0.2249
HGLH1	**0.1369**	**0.1916**	**0.2498**	**0.3110**	**0.3178**
HGLH2	**0.1422**	**0.1955**	**0.2490**	**0.3165**	**0.3241**
HGBH0	0.1026	0.1412	0.1856	0.2442	0.2311
HGBH1	0.1295	0.1856	**0.2372**	**0.3069**	0.3060
HGBH2	0.1348	**0.1946**	**0.2437**	**0.3154**	0.3146

Table 20: Performance of DFR matching measures on Oxford

	P@10	P@20	P@50	P@100	MAP
PLH0	**0.7888**	0.7338	0.6583	0.5952	0.3939
PLH1	**0.7853**	0.7307	0.6546	0.5919	0.3934
PLH2	**0.7883**	0.7325	0.6557	0.5925	0.3937
PBH0	0.7883	**0.7449**	0.6612	0.6008	0.3974
PBH1	0.7843	0.7360	**0.6622**	0.5983	**0.3968**
PBH2	0.7878	0.7391	**0.6622**	0.5988	0.3971
DLH0	**0.7878**	0.7338	0.6569	0.5944	0.3936
DLH1	**0.7939**	**0.7424**	**0.6668**	**0.6072**	0.3991
DLH2	**0.7954**	0.7345	0.6638	0.6013	0.3968
DBH0	**0.7893**	**0.7419**	0.6608	0.6003	0.3971
DBH1	**0.7975**	**0.7470**	**0.6718**	**0.6116**	**0.4020**
DBH2	**0.7985**	**0.7467**	**0.6680**	0.6076	0.4001
GLH0	**0.7893**	**0.7442**	**0.6643**	0.5934	0.3937
GLH1	0.7822	0.7371	0.6572	0.5909	0.3911
GLH2	0.7843	0.7368	0.6592	0.5920	0.3920
GBH0	**0.7909**	**0.7449**	**0.6689**	0.6018	**0.3974**
GBH1	**0.7888**	0.7371	**0.6622**	0.5975	0.3947
GBH2	**0.7919**	0.7406	**0.6632**	0.5984	0.3956
BLH0	**0.7893**	**0.7442**	**0.6641**	0.5933	0.3937
BLH1	0.7817	0.7365	0.6570	0.5908	0.3911
BLH2	0.7843	0.7371	0.6591	0.5920	0.3920
BBH0	**0.7909**	**0.7452**	**0.6688**	0.6017	**0.3974**
BBH1	**0.7888**	0.7365	**0.6620**	0.5975	0.3947
BBH2	**0.7924**	0.7409	**0.6632**	0.5984	0.3956
InLH0	**0.7898**	**0.7442**	**0.6646**	0.5993	0.3960
InLH1	**0.7898**	0.7409	**0.6641**	0.5986	0.3969
InLH2	**0.7893**	**0.7424**	**0.6640**	0.5994	0.3968
InBH0	**0.7924**	**0.7475**	**0.6710**	**0.6051**	0.3997
InBH1	**0.7959**	**0.7447**	**0.6696**	**0.6070**	**0.4006**
InBH2	**0.7944**	**0.7452**	**0.6712**	**0.6073**	**0.4005**
IneLH0	**0.7919**	0.7411	**0.6632**	0.5942	0.3943
IneLH1	0.7873	0.7376	**0.6619**	0.5965	0.3952
IneLH2	**0.7873**	**0.7401**	**0.6639**	0.5962	0.3951
IneBH0	**0.7924**	**0.7472**	**0.6690**	**0.6050**	**0.3986**
IneBH1	**0.7924**	**0.7434**	**0.6684**	0.6041	**0.3995**
IneBH2	**0.7970**	**0.7457**	**0.6691**	**0.6054**	**0.3994**
HGLH0	**0.7873**	**0.7396**	**0.6611**	0.5976	**0.3972**
HGLH1	0.7827	0.7274	0.6533	0.5842	0.3871
HGLH2	0.7812	0.7338	0.6537	0.5878	0.3896
HGBH0	**0.7878**	**0.7452**	**0.6659**	**0.6032**	0.4006
HGBH1	0.7812	0.7340	0.6571	0.5922	0.3900
HGBH2	0.7817	0.7371	**0.6605**	0.5946	0.3927

Table 21: Performance of DFR weighting with L1 distance on Caltech6

	P@10	P@20	P@50	P@100	MAP
PLH0	0.5751	0.4881	0.3944	0.3494	0.3216
PLH1	0.5604	0.4711	0.3750	0.3329	0.3191
PLH2	0.5650	0.4827	0.3859	0.3421	0.3209
PBH0	0.6051	0.5218	0.4275	0.3809	0.3302
PBH1	0.5878	0.4997	0.4085	0.3629	0.3270
PBH2	0.5970	0.5089	0.4169	0.3737	0.3290
DLH0	0.5777	0.4924	0.3976	0.3518	0.3223
DLH1	0.6340	0.5515	0.4656	0.4138	0.3393
DLH2	0.6132	0.5287	0.4411	0.3922	0.3329
DBH0	0.6081	0.5231	0.4305	0.3845	0.3309
DBH1	**0.6513**	**0.5711**	**0.4840**	**0.4338**	**0.3441**
DBH2	0.6340	0.5536	0.4660	0.4159	0.3395
GLH0	0.5584	0.4726	0.3749	0.3317	0.3164
GLH1	0.5421	0.4594	0.3620	0.3207	0.3137
GLH2	0.5462	0.4622	0.3656	0.3238	0.3146
GBH0	0.5863	0.5056	0.4083	0.3625	0.3253
GBH1	0.5604	0.4835	0.3901	0.3431	0.3213
GBH2	0.5690	0.4883	0.3958	0.3475	0.3225
BLH0	0.5579	0.4726	0.3751	0.3318	0.3164
BLH1	0.5421	0.4594	0.3618	0.3206	0.3137
BLH2	0.5462	0.4622	0.3656	0.3239	0.3146
BBH0	0.5863	0.5058	0.4084	0.3626	0.3253
BBH1	0.5599	0.4835	0.3901	0.3431	0.3213
BBH2	0.5690	0.4883	0.3958	0.3477	0.3225
InLH0	0.5680	0.4810	0.3891	0.3436	0.3207
InLH1	0.5777	0.4898	0.4012	0.3540	0.3242
InLH2	0.5761	0.4886	0.3986	0.3525	0.3236
InBH0	0.6036	0.5269	0.4296	0.3790	0.3302
InBH1	0.6066	0.5234	0.4313	0.3848	0.3324
InBH2	0.6051	0.5239	0.4308	0.3845	0.3321
IneLH0	0.5624	0.4706	0.3765	0.3336	0.3175
IneLH1	0.5645	0.4820	0.3894	0.3452	0.3211
IneLH2	0.5680	0.4817	0.3873	0.3433	0.3204
IneBH0	0.5827	0.5020	0.4085	0.3625	0.3269
IneBH1	0.5934	0.5099	0.4183	0.3708	0.3304
IneBH2	0.5949	0.5107	0.4177	0.3682	0.3299
HGLH0	0.5888	0.5033	0.4080	0.3634	0.3260
HGLH1	0.5127	0.4312	0.3408	0.2983	0.3101
HGLH2	0.5239	0.4401	0.3468	0.3033	0.3125
HGBH0	**0.6310**	0.5371	0.4454	0.3958	0.3349
HGBH1	0.5411	0.4591	0.3668	0.3239	0.3167
HGBH2	0.5452	0.4693	0.3759	0.3308	0.3195

Table 22: Performance of DFR weighting with L2 distance on Caltech6

	P@10	P@20	P@50	P@100	MAP
PLH0	**0.3106**	**0.2548**	**0.1990**	0.1595	0.1057
PLH1	**0.3096**	**0.2540**	**0.1984**	**0.1600**	0.1056
PLH2	**0.3106**	**0.2543**	**0.1984**	**0.1599**	0.1056
PBH0	**0.3106**	**0.2551**	**0.1998**	**0.1635**	**0.1074**
PBH1	**0.3121**	**0.2556**	0.1993	**0.1631**	**0.1074**
PBH2	**0.3111**	**0.2553**	**0.1997**	**0.1632**	**0.1073**
DLH0	**0.3111**	**0.2545**	**0.1991**	0.1594	0.1056
DLH1	0.3086	**0.2528**	**0.1970**	0.1586	0.1053
DLH2	**0.3106**	**0.2533**	**0.1975**	**0.1594**	0.1054
DBH0	**0.3111**	**0.2563**	**0.2000**	**0.1632**	**0.1074**
DBH1	**0.3111**	**0.2540**	**0.1979**	**0.1624**	**0.1069**
DBH2	**0.3106**	**0.2553**	**0.1987**	**0.1631**	**0.1071**
GLH0	**0.3101**	**0.2545**	**0.1985**	0.1594	0.1057
GLH1	0.3096	0.2538	**0.1981**	**0.1596**	0.1056
GLH2	0.3096	0.2540	**0.1982**	0.1595	0.1056
GBH0	**0.3116**	**0.2556**	0.1995	0.1633	**0.1076**
GBH1	**0.3106**	0.2551	**0.1993**	0.1631	0.1074
GBH2	**0.3111**	0.2556	**0.1992**	0.1632	**0.1074**
BLH0	**0.3096**	**0.2548**	**0.1985**	0.1595	0.1057
BLH1	0.3101	0.2538	**0.1980**	**0.1596**	0.1056
BLH2	0.3101	0.2540	0.1979	0.1594	0.1056
BBH0	**0.3121**	**0.2558**	**0.1993**	**0.1634**	**0.1076**
BBH1	**0.3116**	0.2553	0.1988	0.1630	0.1073
BBH2	**0.3116**	0.2556	0.1987	**0.1633**	0.1074
InLH0	**0.3126**	**0.2548**	**0.1988**	**0.1607**	0.1063
InLH1	**0.3131**	**0.2540**	**0.1987**	**0.1602**	0.1062
InLH2	**0.3126**	**0.2548**	**0.1990**	**0.1603**	0.1062
InBH0	**0.3111**	**0.2566**	**0.1997**	**0.1641**	**0.1078**
InBH1	**0.3091**	**0.2571**	0.1992	**0.1639**	0.1076
InBH2	**0.3086**	**0.2571**	0.1989	**0.1638**	0.1076
IneLH0	**0.3101**	**0.2543**	**0.1983**	0.1593	0.1057
IneLH1	0.3096	**0.2540**	**0.1978**	**0.1595**	0.1055
IneLH2	0.3096	0.2538	**0.1981**	**0.1595**	0.1056
IneBH0	**0.3111**	**0.2561**	**0.1995**	**0.1631**	**0.1075**
IneBH1	**0.3111**	0.2553	**0.1993**	0.1628	0.1073
IneBH2	**0.3111**	0.2553	**0.1991**	0.1628	**0.1074**
HGLH0	**0.3106**	**0.2543**	**0.1992**	**0.1599**	0.1058
HGLH1	0.3086	**0.2553**	**0.1987**	**0.1601**	0.1057
HGLH2	0.3091	**0.2543**	**0.1989**	0.1598	0.1058
HGBH0	**0.3121**	**0.2551**	**0.2003**	**0.1635**	**0.1075**
HGBH1	**0.3111**	**0.2571**	**0.1998**	**0.1634**	**0.1075**
HGBH2	**0.3111**	**0.2566**	**0.2001**	**0.1633**	**0.1075**

Table 23: Performance of DFR weighting with L1 distance on Caltech101

	P@10	P@20	P@50	P@100	MAP
PLH0	0.2828	**0.2258**	**0.1699**	**0.1396**	**0.0957**
PLH1	0.2848	**0.2258**	**0.1695**	**0.1390**	**0.0956**
PLH2	**0.2874**	**0.2268**	**0.1698**	**0.1390**	**0.0957**
PBH0	0.2783	**0.2255**	**0.1710**	**0.1439**	0.0979
PBH1	0.2773	**0.2268**	0.1708	**0.1435**	0.0977
PBH2	0.2768	**0.2265**	**0.1713**	**0.1439**	0.0977
DLH0	0.2838	**0.2247**	**0.1692**	**0.1386**	**0.0956**
DLH1	**0.2904**	**0.2258**	**0.1685**	**0.1384**	**0.0958**
DLH2	**0.2894**	**0.2270**	**0.1698**	**0.1394**	**0.0959**
DBH0	0.2778	**0.2242**	0.1706	0.1428	0.0977
DBH1	0.2788	**0.2268**	**0.1707**	**0.1440**	0.0978
DBH2	0.2778	**0.2273**	**0.1722**	**0.1440**	**0.0978**
GLH0	0.2808	0.2242	**0.1690**	**0.1387**	**0.0953**
GLH1	**0.2869**	0.2237	**0.1677**	**0.1386**	**0.0951**
GLH2	**0.2864**	**0.2245**	**0.1681**	**0.1386**	**0.0952**
GBH0	0.2798	**0.2245**	**0.1709**	**0.1435**	0.0978
GBH1	0.2763	**0.2237**	**0.1725**	**0.1433**	0.0975
GBH2	0.2788	**0.2245**	**0.1720**	**0.1433**	**0.0977**
BLH0	0.2813	0.2242	**0.1685**	**0.1384**	**0.0953**
BLH1	**0.2869**	0.2240	0.1673	**0.1380**	**0.0950**
BLH2	**0.2869**	**0.2240**	**0.1679**	**0.1383**	**0.0951**
BBH0	0.2783	**0.2250**	0.1710	**0.1433**	0.0977
BBH1	0.2783	**0.2235**	**0.1720**	**0.1428**	0.0975
BBH2	0.2793	**0.2247**	**0.1717**	**0.1431**	0.0977
InLH0	0.2828	**0.2280**	**0.1720**	**0.1413**	**0.0966**
InLH1	**0.2894**	**0.2270**	**0.1707**	**0.1406**	**0.0963**
InLH2	**0.2879**	**0.2273**	**0.1708**	**0.1404**	**0.0965**
InBH0	0.2722	**0.2202**	0.1693	**0.1434**	0.0976
InBH1	0.2702	**0.2205**	0.1702	**0.1433**	0.0974
InBH2	0.2712	**0.2205**	0.1702	**0.1436**	0.0975
IneLH0	0.2803	0.2242	**0.1690**	**0.1383**	**0.0952**
IneLH1	**0.2869**	**0.2245**	**0.1680**	**0.1382**	**0.0950**
IneLH2	**0.2864**	**0.2240**	**0.1679**	**0.1382**	**0.0951**
IneBH0	0.2793	**0.2250**	**0.1710**	**0.1432**	0.0977
IneBH1	0.2763	**0.2232**	**0.1725**	**0.1429**	**0.0975**
IneBH2	0.2773	**0.2250**	**0.1724**	**0.1431**	**0.0976**
HGLH0	**0.2869**	**0.2280**	**0.1711**	**0.1405**	**0.0961**
HGLH1	0.2758	0.2217	**0.1674**	**0.1387**	**0.0951**
HGLH2	0.2773	0.2237	**0.1679**	**0.1390**	**0.0953**
HGBH0	0.2798	**0.2270**	**0.1720**	**0.1443**	**0.0983**
HGBH1	0.2692	**0.2235**	0.1693	**0.1429**	0.0973
HGBH2	0.2682	**0.2235**	0.1699	0.1428	0.0974

Table 24: Performance of DFR weigthing with L2 distance on Caltech101

	P@4	MAP
PLH0	**0.7197**	**0.5356**
PLH1	**0.7205**	**0.5349**
PLH2	**0.7197**	**0.5354**
PBH0	0.7146	0.5316
PBH1	0.7113	0.5292
PBH2	0.7130	0.5306
DLH0	**0.7197**	**0.5357**
DLH1	**0.7214**	**0.5371**
DLH2	**0.7197**	**0.5362**
DBH0	**0.7146**	0.5319
DBH1	0.7130	0.5318
DBH2	**0.7163**	0.5321
GLH0	0.7113	**0.5344**
GLH1	0.7113	0.5333
GLH2	0.7104	0.5336
GBH0	0.7037	0.5293
GBH1	0.7045	0.5288
GBH2	0.7029	0.5283
BLH0	0.7121	**0.5345**
BLH1	0.7113	0.5333
BLH2	0.7104	0.5336
BBH0	0.7037	0.5293
BBH1	0.7045	0.5289
BBH2	0.7029	0.5284
InLH0	0.7130	0.5341
InLH1	0.7096	0.5329
InLH2	0.7113	0.5332
InBH0	0.7003	0.5282
InBH1	0.7012	0.5275
InBH2	0.6995	0.5278
IneLH0	0.7130	**0.5355**
IneLH1	0.7121	**0.5348**
IneLH2	0.7138	**0.5351**
IneBH0	0.7113	0.5315
IneBH1	0.7113	0.5309
IneBH2	0.7121	0.5306
HGLH0	**0.7180**	**0.5347**
HGLH1	0.7138	0.5325
HGLH2	**0.7180**	0.5336
HGBH0	0.7121	0.5299
HGBH1	0.7029	0.5271
HGBH2	0.7054	0.5277

Table 25: Performance of DFR weighting with L1 distance on Nister

	P@4	MAP
PLH0	0.6532	0.4889
PLH1	0.6473	0.4846
PLH2	0.6498	0.4877
PBH0	0.6187	0.4746
PBH1	0.6170	0.4708
PBH2	0.6195	0.4733
DLH0	0.6524	0.4894
DLH1	**0.6751**	**0.5061**
DLH2	**0.6726**	0.5031
DBH0	0.6187	0.4751
DBH1	0.6431	0.4895
DBH2	0.6347	0.4840
GLH0	0.6515	0.4879
GLH1	0.6465	0.4834
GLH2	0.6465	0.4845
GBH0	0.6279	0.4787
GBH1	0.6263	0.4759
GBH2	0.6279	0.4768
BLH0	0.6515	0.4879
BLH1	0.6465	0.4834
BLH2	0.6465	0.4846
BBH0	0.6271	0.4787
BBH1	0.6263	0.4761
BBH2	0.6279	0.4768
InLH0	0.6540	0.4894
InLH1	0.6490	0.4886
InLH2	0.6481	0.4886
InBH0	0.6044	0.4642
InBH1	0.6103	0.4659
InBH2	0.6120	0.4667
IneLH0	0.6524	0.4881
IneLH1	0.6490	0.4873
IneLH2	0.6507	0.4877
IneBH0	0.6347	0.4808
IneBH1	0.6347	0.4817
IneBH2	0.6330	0.4816
HGLH0	0.6574	0.4930
HGLH1	0.6355	0.4768
HGLH2	0.6397	0.4798
HGBH0	0.6237	0.4770
HGBH1	0.6086	0.4640
HGBH2	0.6136	0.4657

Table 26: Performance of DFR weighting with L2 distance on Nister

	P@5	P@10	P@20	P@50	MAP
PLH0	**0.7091**	**0.5655**	**0.4309**	**0.2778**	0.2646
PLH1	**0.7200**	**0.5655**	0.4300	0.2760	0.2660
PLH2	**0.7200**	**0.5655**	**0.4309**	0.2764	0.2660
PBH0	**0.7127**	**0.5691**	**0.4309**	0.2753	0.2661
PBH1	**0.7164**	**0.5691**	**0.4327**	**0.2756**	0.2675
PBH2	**0.7164**	**0.5691**	**0.4327**	**0.2760**	0.2675
DLH0	0.7127	**0.5673**	**0.4309**	**0.2782**	0.2657
DLH1	0.7200	**0.5691**	**0.4318**	**0.2767**	0.2684
DLH2	**0.7236**	**0.5673**	**0.4318**	**0.2767**	0.2679
DBH0	**0.7127**	**0.5709**	**0.4318**	**0.2767**	0.2672
DBH1	**0.7200**	**0.5727**	**0.4355**	**0.2764**	**0.2699**
DBH2	**0.7164**	**0.5709**	**0.4327**	**0.2771**	0.2693
GLH0	0.7018	0.5618	0.4218	0.2731	0.2592
GLH1	**0.7127**	**0.5655**	0.4282	0.2727	0.2624
GLH2	**0.7091**	0.5636	0.4264	0.2724	0.2613
GBH0	**0.7091**	**0.5655**	0.4236	0.2716	0.2612
GBH1	**0.7127**	**0.5673**	0.4273	0.2724	0.2640
GBH2	**0.7164**	0.5655	0.4264	0.2716	0.2634
BLH0	0.7018	0.5618	0.4218	0.2735	0.2594
BLH1	**0.7127**	**0.5655**	0.4282	0.2735	0.2624
BLH2	**0.7091**	0.5636	0.4264	0.2727	0.2614
BBH0	**0.7091**	**0.5655**	0.4255	0.2716	0.2613
BBH1	**0.7127**	**0.5673**	0.4282	0.2724	0.2641
BBH2	**0.7164**	0.5655	0.4273	0.2716	0.2635
InLH0	**0.7091**	0.5618	0.4227	0.2724	0.2601
InLH1	**0.7127**	**0.5655**	0.4273	0.2742	0.2629
InLH2	**0.7127**	**0.5655**	0.4264	0.2727	0.2623
InBH0	**0.7018**	**0.5673**	0.4255	0.2705	0.2612
InBH1	**0.7200**	**0.5673**	0.4273	0.2724	0.2645
InBH2	**0.7164**	**0.5673**	0.4273	0.2731	0.2637
IneLH0	0.7055	0.5618	0.4227	0.2742	0.2603
IneLH1	**0.7127**	**0.5655**	0.4291	0.2745	0.2632
IneLH2	**0.7091**	**0.5655**	0.4273	0.2735	0.2624
IneBH0	**0.7127**	**0.5673**	0.4245	0.2724	0.2626
IneBH1	**0.7236**	**0.5673**	**0.4300**	**0.2749**	0.2655
IneBH2	**0.7164**	**0.5673**	0.4273	0.2738	0.2644
HGLH0	**0.7091**	0.5636	0.4300	0.2742	0.2647
HGLH1	**0.7236**	0.5655	0.4282	0.2753	0.2658
HGLH2	**0.7200**	0.5636	0.4291	0.2738	0.2658
HGBH0	**0.7164**	**0.5691**	**0.4318**	**0.2764**	0.2664
HGBH1	**0.7164**	**0.5727**	**0.4336**	**0.2745**	0.2671
HGBH2	**0.7164**	**0.5709**	**0.4336**	**0.2753**	0.2671

Table 27: Performance of DFR weighting with L1 distance on Oxford

	P@5	P@10	P@20	P@50	MAP
PLH0	0.6691	0.5527	0.4209	0.2680	0.2639
PLH1	0.6691	0.5527	0.4200	0.2698	0.2646
PLH2	0.6727	0.5509	0.4218	0.2698	0.2659
PBH0	0.6727	**0.5600**	0.4264	0.2775	0.2722
PBH1	0.6836	0.5600	0.4291	0.2785	0.2723
PBH2	0.6800	0.5600	0.4309	0.2793	0.2741
DLH0	**0.6691**	0.5545	0.4218	0.2691	0.2655
DLH1	**0.6982**	**0.5727**	**0.4409**	0.2800	0.2834
DLH2	**0.6945**	**0.5655**	**0.4336**	0.2771	0.2774
DBH0	**0.6764**	**0.5618**	0.4300	0.2785	0.2743
DBH1	**0.6982**	**0.5727**	**0.4436**	**0.2920**	**0.2887**
DBH2	**0.6909**	**0.5691**	**0.4409**	0.2862	0.2844
GLH0	0.6473	0.5436	0.4145	0.2658	0.2570
GLH1	0.6618	0.5455	0.4173	0.2651	0.2603
GLH2	0.6618	0.5473	0.4173	0.2647	0.2593
GBH0	0.6655	0.5491	0.4218	0.2738	0.2665
GBH1	0.6691	0.5527	0.4227	0.2749	0.2701
GBH2	0.6691	0.5527	0.4245	0.2742	0.2695
BLH0	0.6509	0.5436	0.4155	0.2658	0.2578
BLH1	0.6618	0.5473	0.4173	0.2651	0.2604
BLH2	0.6618	0.5491	0.4173	0.2647	0.2596
BBH0	0.6655	0.5491	0.4218	0.2735	0.2669
BBH1	0.6691	0.5527	0.4227	0.2749	0.2703
BBH2	0.6691	0.5527	0.4245	0.2742	0.2696
InLH0	0.6545	0.5436	0.4173	0.2673	0.2603
InLH1	0.6618	0.5545	0.4200	0.2684	0.2654
InLH2	0.6618	0.5527	0.4209	0.2680	0.2649
InBH0	0.6582	0.5491	0.4282	0.2756	0.2689
InBH1	0.6764	0.5600	0.4300	0.2800	0.2732
InBH2	0.6727	0.5600	0.4291	0.2778	0.2728
IneLH0	0.6509	0.5418	0.4136	0.2647	0.2584
IneLH1	0.6655	0.5509	0.4182	0.2662	0.2630
IneLH2	0.6655	0.5473	0.4182	0.2665	0.2625
IneBH0	0.6655	0.5491	0.4218	0.2753	0.2678
IneBH1	0.6764	0.5600	0.4273	0.2767	0.2737
IneBH2	**0.6764**	0.5582	0.4264	0.2764	0.2726
HGLH0	**0.6873**	0.5564	0.4282	0.2720	0.2725
HGLH1	0.6691	0.5491	0.4200	0.2698	0.2634
HGLH2	0.6691	0.5545	0.4209	0.2713	0.2658
HGBH0	**0.6873**	**0.5655**	0.4364	0.2833	0.2786
HGBH1	0.6727	0.5582	0.4291	0.2793	0.2714
HGBH2	0.6727	**0.5600**	0.4309	0.2793	0.2732

Table 28: Performance of DFR weighting with L2 distance on Oxford

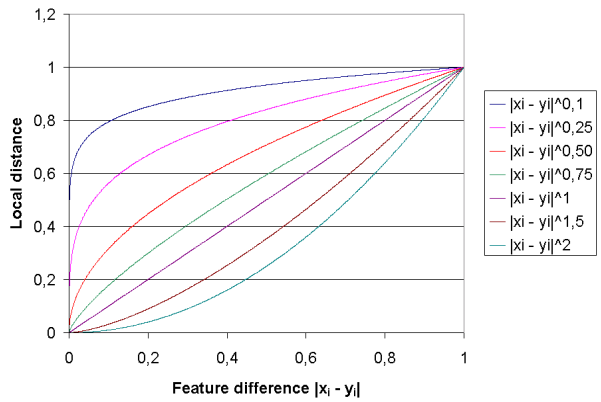


Figure 12: Importance of local distances according to the value of k

6 Discussion

6.1 The effect of k in Lk distances

On Caltech6 and Nister, the effect of distance parameter k is consistent with the observations of [1] and especially [12]: values smaller than one improve the retrieval performance, but below a given threshold, the performance begins to decrease. This behavior is not discussed in these papers. In our case, the optimal value is $k = 0.75$. The influence of k in Lk distance is the following: large values of k give more importance to local distances (*i.e.* distance for a given coordinate of the vectors) whereas small values emphasize the simple fact that the values of each vector for a coordinate are equal or not (see Figure 12). The fact that the performance decreases when k becomes too small shows that a trade-off between considering the difference between frequencies or the only fact that frequencies are differents must be found.

On Caltech101, using $k < 1$ provides much less improvement compared to Caltech6 and Nister. One difference between Caltech101 and these datasets is that the vocabulary used in Caltech101 is much larger: this results in more sparse vectors of higher dimension. However, experiments in [12] show that fractional distances tend to perform better on sparse vectors than on dense vectors. The other difference with Caltech6 and Nister is that data vectors contain much more noise. Compared to Caltech6 data, Caltech101 vocabulary contains much more visual words, so the probability that a given local descriptor falls into one word (*i.e* cluster) instead of another is much larger. Compared to Nister data, there is much more variations in local descriptors found on images relevant to one query, because relevant images in Nister contain the same objects than the query, seen from a different point of view, whereas there are many intra-class variations in Caltech101: the probability for a local descriptor to fall in the wrong visual word is then also greater in Caltech101 than in Nister. The noise in the data can explain the stability of the results over the values of k, as Aggrawal *et. al* showed that in the presence of noise, Lk distances tend to be equivalent for any k [1].

6.2 Local weighting

The best local weight to use depends strongly on the dataset considered. In the case of the L1 distance, the results are:

- on Caltech6, the best local weights are $l3$, $l7$ and then $l4$;
- on Caltech101, the performance differences are not statistically significant, except for $l6$ that performs worse than the others;
- on Nister, $l2$ and $l7$ perform best;
- on Oxford, $l6$ yields the best results, $l5$ gives similar results than $l1$ (*i.e.* standard tf), and the others decrease the performance.

Figure 13 shows, for each dataset, mean and standard error of visual word frequency in the images they occur in. For Caltech6 and Nister datasets, the mean frequency of the visual words lies between 1 and 2 for most of the words, and quickly increase for a small portion of them. Moreover, there are a few words with mean frequency equal to 1, but most of them are not hapax, *i.e.* words occuring only once in the whole collection (see Figure 30). With such a frequency model, we can understand the effectiveness of local weights such as $l3$, $l7$ or $l2$. As they reduce the higher frequency values, they make local distances more reduced, and then give local distances due to high frequency differences less importance in the overall distance.

	p@4	p@5	p@10	p@20	p@50	p@100	MAP
Caltech6	N/A	N/A	+4.3%	+6.3%	+8.0%	+8.5%	+6.3%
Caltech101	N/A	N/A	+1.5%	+1.1%	+0.9%	+0.7%	+0.9%
Nister	+3.3%	N/A	N/A	N/A	N/A	N/A	+2.7%
Oxford	N/A	+3.0%	+7.5%	+9.3%	+10.0%	N/A	+12.3%

Table 29: Best performance improvement compared to baseline weight l1g0 when using L1 distance

	hapax	mean frequency = 1
Caltech-6	0	17
Caltech-101	3021	9790
Nister	56	358
Oxford	1434	4206

Table 30: Comparison of hapax and visual words with mean frequency equal to 1

This effect is consistent with the results observed with Lk distances: giving less importance to local differences (using small k values) improve the overall retrieval performance. On Caltech101, most of the visual words have a frequency between 1 and 1.2, and a few visual words have higher (but still quite low) mean frequencies. Moreover, there are proportionnally much more words with a frequency equal to 1, and about one third of them are *hapax* (see Table 30). This frequency distribution can easily be explained: increasing the number of visual words (*i.e.* clusters) reduce the number of descriptors assigned to each visual word, hence their mean frequency in the documents. This frequencies explain the lack of effect of local weights, as weighted frequencies tend to be the same as original frequencies when tf_{ij} is small. In particular, when $tf_{ij} = 1$, $l1$, $l2$, $l4$ and $l6$ are strictly the same. $l6$ yields the worst results however, as it has a similar effect than $L2$ distance: it gives more importance to local distances when $tf_{ij} > 2$. Oxford frequencies follow a similar model, although they are a little higher and have a more important standard deviation. As it yields contradictory results, they are discussed in Section 6.5.

6.3 Global weighting

The best global weight is not clearly defined:

- On Caltech6, the best results are obtained using mean frequency based global weights ($g4$ and$g5$), but simple IDF-based weights also work well;
- On Nister, squared IDF $g3$ yields the best results, but the difference with $g1$ and $g2$ is not very important;
- On Caltech101, all global weighting schemes give similar result, although using no global weight at all ($g0$) is often the best;
- On Oxford, squared IDF $g3$ performs best but $g5$ (squared IDF*mean TF) also performs very well.

There is no global weighting scheme that outperforms the others, but squared IDF $g3$ seems to provide good performances in many cases. Using mean word frequency in global weights does not work in every case (*e.g.* Nister), but can provide a small improvement on some datasets (Caltech6, Oxford): it probably depends on some specific properties of the dataset. For instance, Nister shall not contain query objects that can be described by repeated visual words, contrary to Caltech6 (eyes from faces, wheels from motorbikes typically provide repeated visual words) or Oxford (see Section 6.5). We can notice that, for the most general and difficult of our datasets, Caltech101, all global weights yields similar results, and best results are provided when no global weight is used. So in the case of very varied and general datasets, using a global weights might not help. This is probably due to the presence of noise in the data, that makes considering words importance by their document frequency awkward.

6.4 DFR-based weighting

The results of the DFR weights are very close from one to another and the difference is generally not statistically significant. The best results are obtained by using D, In or Ine randomness models. $DBH1$ performs particulary well on Nister, whereas the B divergence model seems more suited to the other datasets. About the hyper-geometric model we proposed, we can make two remarks:

- best results are generally obtained when using normalization $H0$, *i.e.* no normalization. This is because document length is already handled by the randomness model;

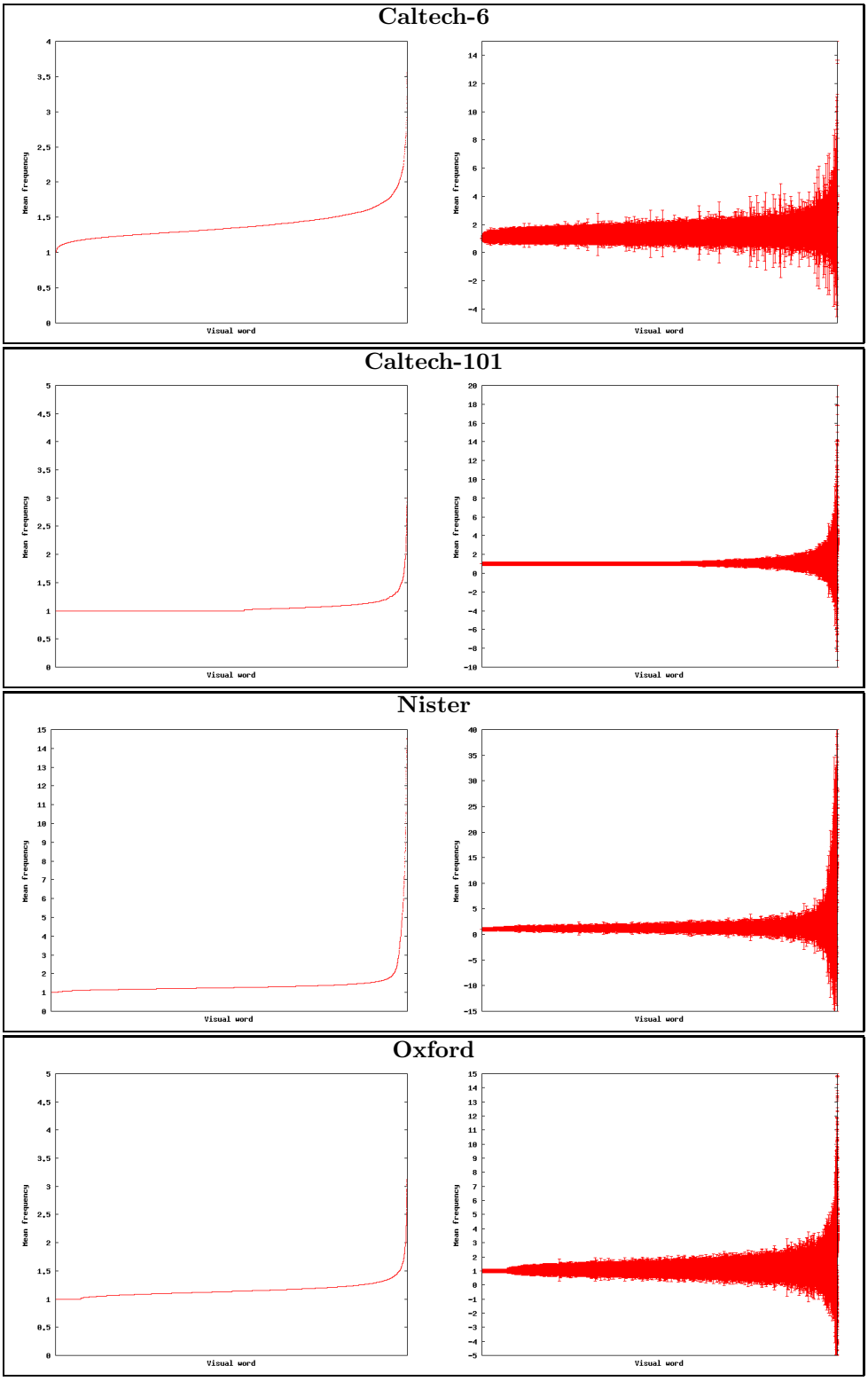


Figure 13: Mean frequency and mean frequency error for each dataset

- although it seemed theoritically more suited than a binomial randomness model, it yields similar results. This may be because under some conditions hyper-geometric models reduce to binomial models.

6.5 The case of the Oxford dataset

The results on the Oxford dataset are generally opposed to the results on the other datasets :

- Lk distances yield the best results with high p values;
- $l6$ local weight performs better than the others with L1 distance.

These results show that this dataset has different properties from the others. On standard datasets, although reducing high word frequencies, and therefore local distances, improves the performances, giving more importance to words with high mean frequency seems to improve the results. This is consistent with the fact that $l6$ weight combined with L1 yields good results, as well as lk distance measures with high k values, or the generally good results observed with global weights $g4$ and $g5$ (that emphasize words with high mean frequency). The fact that binary local weight ($l4$) provides the worse results confirms this idea. This property is probably due to the nature of the query objects: buildings. Figure 14 shows the different query objects available in Oxford. It is clear that many repeated visual words will be found on such images, as they contain many repeated parts (windows, doors, archs...), and that these words will be very relevant to describe these images. Figure 13 also reflects this, as the words have a higher standard deviation with respect to their frequency than in other datasets. Emphasizing the importance of these words in the matching functions naturally improves the system's performance. However, in this case also a trade-off between considering word frequencies and considering word presence in the distance has to be found: we see for instance that using too high distance parameter values, or combining $l6$ weight with an $L2$ distance worsen the results. In a general way, we can consider that this dataset, due to its properties, is different than the others, which provide all coherent results. We therefore can consider that results obtained on Oxford (or similar datasets) may not be systematically generalized to any dataset.

7 Conclusion and future work

Recent image retrieval studies rely on the use of the bag of visual words descriptor. This descriptor is quite similar to the standard bag of words descriptor used in text retrieval. Text retrieval systems' performance can usually be improved by the use of appropriate weights that give more importance to the most relevant words of the documents. In this paper we investigated the effect of these weights on the effectiveness of image retrieval. We also tested the role of the distance used to match documents. The main conclusions of this study are:

- matching effectively documents using visual words requires to find a trade-off between matching word frequencies or word presence only. This trade-off can be different due to some properties of the data, although two datasets performed similarly. Increasing or decreasing the importance of word frequence can be done by using an appropriate weight or by choosing the appropriate distance parameter of Minkowski distance.
- when the dataset becomes too varied and general, the presence of noise in the data (due to wrong assignement of local descriptors to visual words) makes the effect of weights insignificant. The choice of the distance also influences the performance only very slightly.

Future work in this way includes studying the influence of the vocabulary size on the effectiveness of weighting schemes. It is known that increasing the vocabulary size usually improve the performance of bag of visual words retrieval [7, 22], but it also makes the computational cost of image retrieval very high. We also observed in this study that using very large vocabularies would make the use of weights insignificant. It would be interesting to see to what extent the use of different vocabulary sizes can change the weights' effect, and whether the use of an appropriate weighting scheme can limit the vocabulary size with no lost of retrieval effectiveness. We could also check if the consistence of these results when using dense sampling of descriptors (*i.e.* dividing the image according to a grid) instead of using an interest point detector.

References

[1] Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. On the surprising behavior of distance metrics in high dimensional space. In *Lecture Notes in Computer Science*, pages 420–434. Springer, 2001.

[2] G. Amati and C. J. Van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Transactions on Information Systems*, 20(4):357–389, 2002.

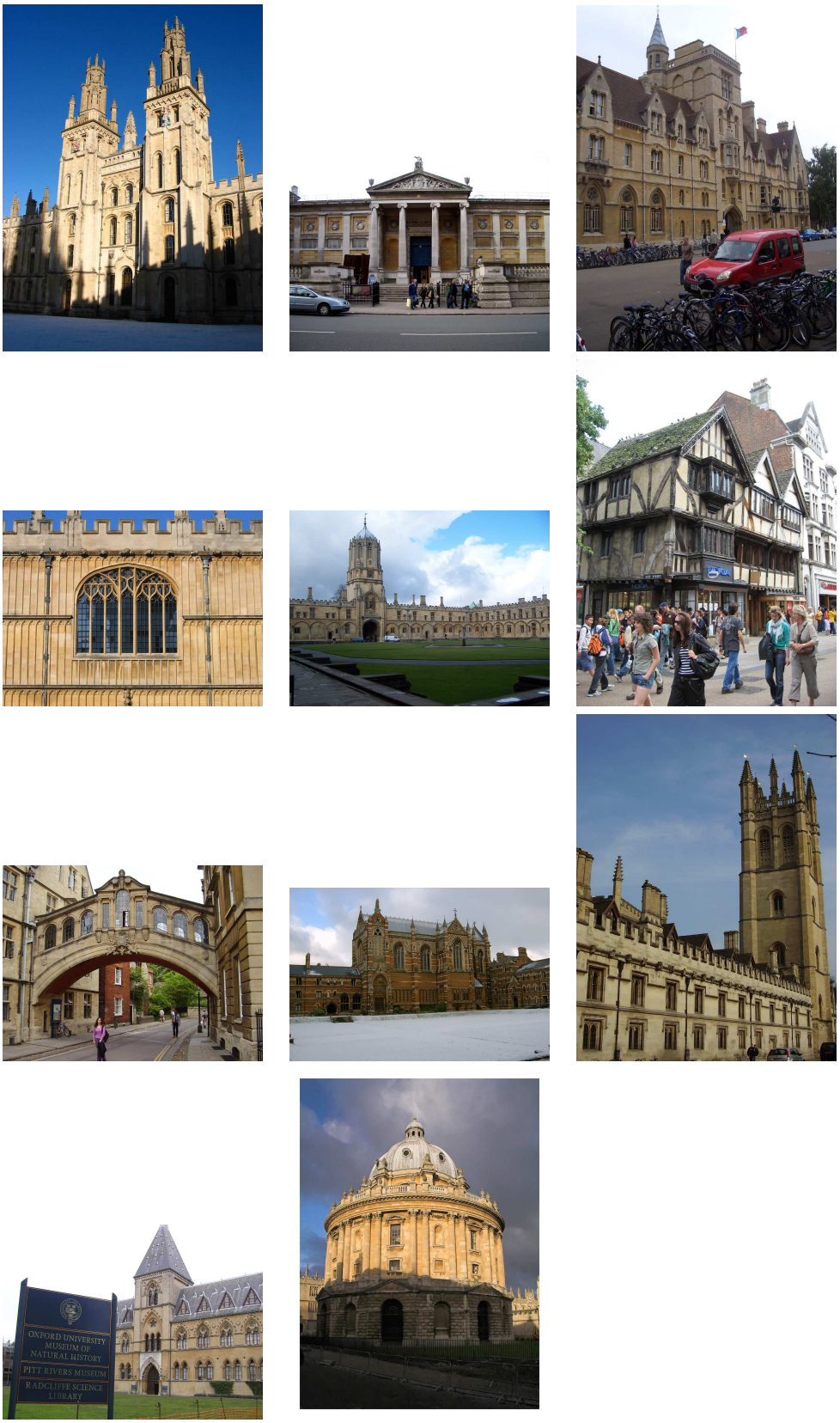


Figure 14: Query objects from Oxford dataset

[3] A. Bosch, A. Zisserman, and X. Munoz. Scene classification via pLSA. In *Proceedings of the European Conference on Computer Vision*, 2006.

[4] Sabri Boughorbel, Nozha Boujemaa, and Constantin Vertan. Histogram-based color signatures for image indexing. In *Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU2002*, 2002.

[5] C. Buckley, G. Salton, and J. Allan. Automatic retrieval with locality information using SMART. In *Proceedings of the first Text Retrieval Conference*, Gaithersburg, USA, 1992.

[6] G. Carneiro and A.D. Jepson. Flexible spatial models for grouping local image features. *Proceedings of CVPR*, 2:II–747–II–754 Vol.2, 27 Jun-2 Jul 2004.

[7] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *Proceedings of the 11th International Conference on Computer Vision, Rio de Janeiro, Brazil*, 2007.

[8] Gabriela Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cedric Bray. Visual categorization with bags of keypoints. In *ECCV: Workshop on Statistical Learning in Computer Vision*, Prague, Czech Republic, May 2004.

[9] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In *IEEE CVPR 2004, Workshop on Generative-Model Based Vision*, 2004.

[10] S. P. Harter. A probabilistic approach to automatic keyword indexing. Part 1: on the distribution of specialty words in the litterature. *Journal of the American Society for Information Science*, pages 197–216, 1975.

[11] Peter Howarth and Stefan Rüger. Evaluation of texture features for content-based image retrieval. In *Proceedings of the third international Conference on Image and Video Retrieval : CIVR 2004*, pages 326–334, Dublin, Irland, 2004.

[12] Peter Howarth and Stefan Rüger. Fractional distance measures for content-based image retrieval. In *In 27th European Conference on Information Retrieval*, pages 447–456. Springer, 2005.

[13] M. Jamieson, S. Dickinson, S. Stevenson, and S. Wachsmuth. Using language to drive the perceptual grouping of local image features. *Proceedings of CVPR*, 2:2102–2109, 2006.

[14] H. Jegou, H. Harzallah, and C. Schmid. A contextual dissimilarity measure for accurate and efficient image search. In *Computer Vision and Pattern Recognition, 2007*, pages 1–8, 2007.

[15] Hervé Jegou, Matthijs Douze, and Cordelia Schmid. Hamming embedding and weak spatial consistency for large scale image search. In *Proceedings of ECCV*, 2008.

[16] Yu-Gang Jiang, Chong-Wah Ngo, and Jun Yang. Towards optimal bag-of-features for object categorization and semantic video retrieval. In *CIVR'07: Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 494–501, New York, NY, USA, 2007. ACM.

[17] K. Sparck Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: development and comparative experiments (part 1). *Information Processing and Management*, 36(6):779–808, 2000.

[18] K. Sparck Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: development and comparative experiments (part 2). *Information Processing and Management*, 36(6):809–840, 2000.

[19] Diane Larlus and Frederic Jurie. Latent mixture vocabularies for object categorization. In *British Machine Vision Conference*, 2006.

[20] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1-2):43–72, 2005.

[21] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[22] David Nister and Henrik Stewenius. Scalable recognition with a vocabulary tree. In *Proceedings of CVPR*, pages 2161–2168, Washington, DC, USA, 2006. IEEE Computer Society.

[23] 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.

[24] Srinivasa Ramanujan. *The lost notebook and other unpublished papers.* Springer-Verlag, 1988.

[25] S.E. Robertson. The probability ranking principle in information retrieval. *Journal of documentation*, 33:294 – 304, 1977.

[26] S.E. Robertson, S. Walker, S. Jones, M.M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. In *3rd Text Retrieval Conference*, pages 109–126, 1995.

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

[28] Josef Sivic and Andrew Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of ICCV*, volume 2, pages 1470–1477, Nice, France, 2003.

[29] Fei Song. A general language model for information retrieval. In *In Proceedings of the 1999 ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 279–280, 1999.

[30] Pierre Tirilly, Vincent Claveau, and Patrick Gros. Language modeling for bag-of-visual words image categorization. In *CIVR '08: Proceedings of the 2008 international conference on Cont ent-based Image and Video Retrieval*, pages 249–258, Niagara Falls, Canada, 2008. ACM.

[31] C. J. Van Rijsbergen. *Information retrieval.* Butterworths, 1979.

[32] Jun Yang, Yu-Gang Jiang, Alexander G. Hauptmann, and Chong-Wah Ngo. Evaluating bag-of-visual-words representations in scene classification. In *MIR '07: Proceedings of the international workshop on multimedia information retrieval*, pages 197–206, New York, NY, USA, 2007. ACM.

[33] Qing-Fang Zheng, Wei-Qiang Wang, and Wen Gao. Effective and efficient object-based image retrieval using visual phrases. In *Proceedings of ACM Multimedia*, pages 77–80, New York, USA, 2006. ACM.