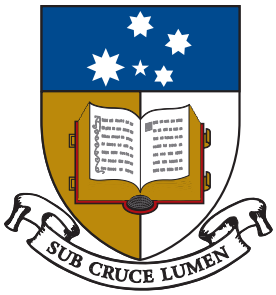


THE UNIVERSITY OF ADELAIDE

School of Computer Science

Efficient and Robust Image Ranking for Object Retrieval

Yanzhi Chen

December, 2013

SUBMITTED FOR THE DEGREE OF DOCTOR OF PHILOSOPHY IN THE FACULTY OF ENGINEERING, COMPUTER & MATHEMATICAL SCIENCES

IMPROVING THE IMAGE SIMILARITY MEASURE

CHAPTER IV

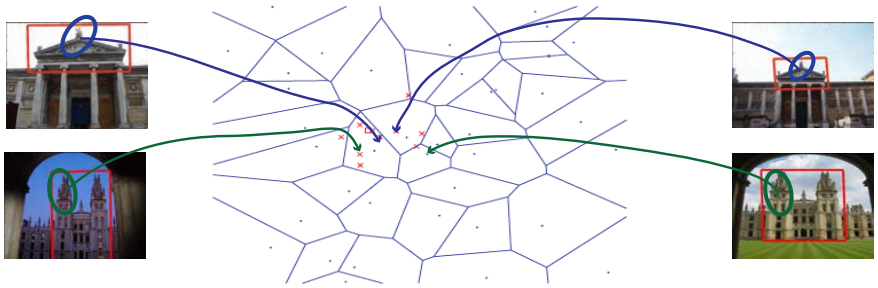
Chapter 3 has proposed a spatial expansion method to improve the BoW representation of the original query, where spatially related words are combined with the query words and the images are ranked in the same way as the standard retrieval system [130]. Therefore, the improvement relies on the refinement of BoW representation of the query.

Image similarity plays an important role in a retrieval system. It determines the distance of a dataset image to a given query. Compared to improving the image representation, it is more challenging to improve the image similarity measure because it should accurately describe the visual similarity between a pair of query/dataset images with imperfect knowledge, *e.g.* the quantised visual words. In this chapter, we aim to improve the image similarity measure in order that the BoW model more accurately ranks dataset images. For a query image q and one of the dataset image d, the standard method measures the similarity between them by the normalised dot product of tf-idf vectors $\mathbf{q}$ and $\mathbf{d}$ vectors corresponding to:

$$\Psi(q,d) = \frac{\mathbf{q} \cdot \mathbf{d}}{\| \mathbf{q} \|_2 \| \mathbf{d} \|_2} \tag{4.1}$$

However, Eq. (4.1) is based on the quantised visual word IDs, and is word-to-word matching. Thus, only co-occurrence of the same visual word in both images contributes to similarity, while different words are considered infinitely distant even though they may be neighbors in feature space. Therefore, the dot product similarity is intolerant to quantisation errors introduced by previous retrieval modules (*e.g.* image representation or vocabulary building), as illustrated in Figure 4.1:

- Relevant features maybe be located in different cluster centres.
- Irrelevant features maybe be located in the same cluster centres.



(a) (b)

Figure 4.1: Example of quantisation errors in feature space: (a) Relevant features might be located in different cluster centres. (b) Irrelevant features might be located in the same cluster centres. The illustration of visual word quantisation in feature space is taken from [63].

This causes inaccuracy in retrieval results. However, two factors have not been well captured in Eq. (4.1):

- The importance of visual words. The tf-idf scheme weights each visual word according to its frequency in the individual image as well as in the corpus. However, tf-idf weights do not distinguish between word appearances in the foreground or background of an image, and therefore can assign high weights to words that are not informative when searching for an object, or vice versa.

- The relatedness of visual words. In the standard BoW model, each feature is mapped to its closest cluster centre (visual word) in the feature space. However, the word similarity should be based not just on proximity in feature space, but also on their association with the same object.

These properties are important in a similarity measure between two BoW vectors, and lead to two enhanced visual distance measures for image features. Firstly, we develop a word re-weighting scheme that is more directly based on how often, and in what range of conditions, a word is correctly matched when it appears as part of the foreground object (Section 4.1). This aims to address the importance of visual word. The visual words occurring in the foreground usually are robust matches between images. Under the standard tf-idf weighting, these visual words are not necessarily weighted strongly – for example they may occur in many images in the database, and therefore have a low idf weight. Intuitively, these foreground visual words can be captured by an object-based thesaurus, as

described in Chapter 3. We use entropy to measure the importance of a visual word according to its spatial co-occurrence distribution, which has been recorded in the object-based thesaurus. A re-weighting scheme is proposed to encourage these foreground visual words (Section 4.1.1). The re-weighting scheme can also combine with spatial expansion presented in Chapter 3, in order that the re-weighting scheme not only considers the visual word importance but also includes visual word relatedness (Section 4.2).

Secondly, visual words also need to incorporate relatedness of visual words, because some matching features might be mapped to different visual words. We use a cross-word matching component in the image similarity score. It is able to consider inter-word distance measured by the enhanced visual word distance (Section 4.3). The similarity measure is related to the Earth Mover's Distance (EMD), proposed in [67] in the context of image retrieval, but is computationally simpler, reducing its query time overhead.

With these improved similarity measurements, the retrieval system is able to more accurately rank the dataset images.

4.1 Visual word re-weighting based on an object-based thesaurus

We show the framework of visual word re-weighting in Figure 4.2, which aims to refine the tf-idf weighting (Section 4.1). After re-weighting, the similarity measure is the same as the standard retrieval system (normalized dot product).

The standard tf-idf does not always capture the importance of visual words, as shown in Eq. (4.1). In contrast, we investigate the importance of those words by the spatial consistency information from an object-based thesaurus, as introduced in Chapter 3. In order to examine word importance, our method uses an information theory based measurement to re-weight the visual words that have been detected as spatially consistent.

4.1.1 The visual word re-weighting scheme

As in Section 3.5, we use an object-based thesaurus to collect the foreground words appearing as inliers $\mathbf{W}_S$. The re-weighting function $\alpha(w_i)$ is applied to adjust the

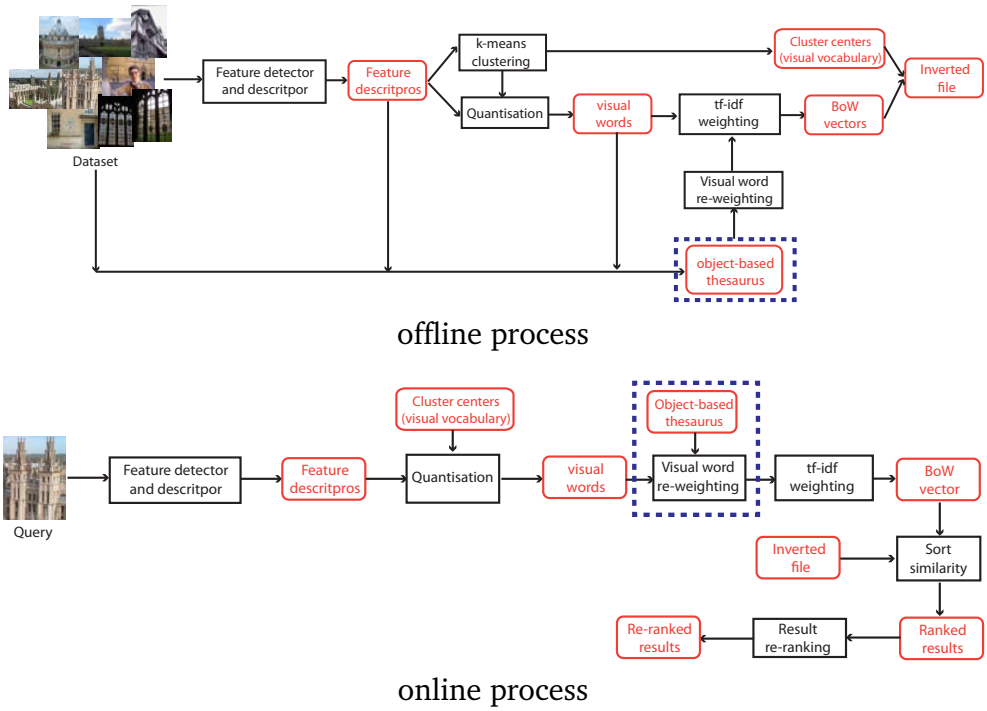


Figure 4.2: System framework of visual word re-weighting. This is the standard BoW retrieval system with new steps (indicated in dash box) introduced by our method.

tf-idf weight $\tau(w_i)$ of each inlier visual word $w_i \in \mathbf{W}_S$:

$$\tau^*(w_i) = \begin{cases} \alpha(w_i) \cdot \tau(w_i) & \text{if } w_i \in \mathbf{W}_S \\ \tau(w_i) & \text{otherwise} \end{cases} \tag{4.2}$$

Note that the similarity function $\Psi(q,d)$ is calculated between all pairs of query q and dataset image d. Therefore, the visual word re-weighting is applied to all query/dataset pairs as well. We now describe the visual word re-weighting scheme with different forms of prior knowledge: i) without prior knowledge; ii) with word frequency; iii) with spatial neighbourhood; iv) with query words.

Re-weighting without prior knowledge Intuitively, the simplest re-weighting function is to multiply the inlier visual words by a constant scale factor, $\alpha(w_i) = c, c > 1$. This treats all inlier visual words as being equally important, and more important than words that do not occur as inliers, but neglects the difference among them.

Re-weighting with word frequency Alternatively, each visual word can be re-weighted according to the frequency with which it occurs as an inlier during the training stage:

$$\alpha(w_i) = P_f(w_i) = \frac{\#\text{ of } w_i \text{ is inliers}}{\#\text{ of } w_i \text{ in training data}} \tag{4.3}$$

The intuition here is that a word that commonly occurs on an object of interest and is correctly matched is likely to be a good indicator of the object. However, this fails to detect words that commonly occur as inliers, but are largely redundant because they are strongly associated with other inlier words. In this case, the object-based thesaurus will include all such words in any query vectors that contains one of them. Weighting all of these words strongly overestimates the confidence of the match and can lead to false positives.

Re-weighting with spatial neighbourhood To avoid these cases, we devise a measure to strongly weight those words that occur as inliers despite a wide variety of words appearing in their spatial neighbourhood. This implies that the word itself is a strong indicator of the presence of the object. In order to choose reliable visual words in this way, we introduce an information theory based method to measure the importance of a visual word based on its neighbourhood diversity, which to our knowledge has not been studied in previous methods. As illustrated in first row of Figure 3.8, it is hard to distinguish true from false results based solely on word matches. Our method is motivated by the following observations on spatial neighborhoods. **Observation 1**: Since the objects of interest are rigid, the spatial structure of these objects are geometrically consistent across images—for example, visual words on the objects satisfy a common epipolar constraint. **Observation 2**: During the random sampling in RANSAC, mismatched features are also likely to be selected as inliers. Such visual words are less informative, and should not be weighted higher than the visual words observed above. As shown in Figure 4.3, these words are often either isolated from other inliers (*i.e.* word D has no neighbors) or, in the case of word C, co-occur with few neighbors in their spatial neighbourhood.

As a result, the diversity of inliers occurring in their spatial neighborhood is essential in measuring the importance of the inlier visual words. Note that we are aiming to encourage visual words that have a diverse nearest neighbors in various spatial region ($\mathbf{D}_i$) in the training data, while discouraging visual words that have

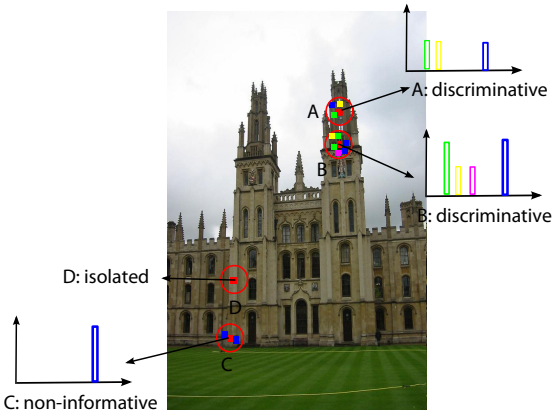


Figure 4.3: Examples of the distribution of the neighborhood of visual words. The red points show examples of the foreground words $\mathbf{W}_S$ appears in the image domain. Three cases of visual words are shown here. Words A and B: discriminative; Word C: redundant; Word D: isolated. Words A and B are located on the corner of the window, while words C and D are located on the wall of the building.

single or few nearest neighbors. Mathematically, the diversity of a distribution can be characterized as *entropy*, which is a classical measurement of uncertainty in information theory [61]. A higher entropy indicates that a variable is more uniformly distributed. Such words occur as inliers despite a wide variety of spatial neighbors, and are the ones we need to heavily weight. Figure 4.4 shows examples of a visual word whose spatial neighborhood has a high entropy. As it occurs frequently in the corpus, its tf-idf weight is low. However, in practice it is a reliable visual word for matching.

To measure entropy, for each visual word $w_i \in \mathbf{W}_S$, we obtain its nearest neighbors $\mathbf{D}_i$, with $\mathbf{D}_i \subset \mathbf{W}_S$. The diversity of the distribution of $\mathbf{D}_i$ is measured by a relative entropy $H(\mathbf{D}_i)$:

$$H(\mathbf{D}_i) = -\sum_j P_f(w_j) \log P_f(w_j) \tag{4.4}$$

where $w_j \in \mathbf{D}_i$ and $P_f(w_j)$ is the frequency that a visual word w_j that occurs in a fixed spatial region, similar to Eq. (4.3). As a result, the re-weighting function can be defined as:

$$\alpha(w_i) = \begin{cases} 1 & \text{if } \mathbf{D}_i = \emptyset \\ \exp(\delta * H(\mathbf{D}_i)) & \text{otherwise} \end{cases} \tag{4.5}$$

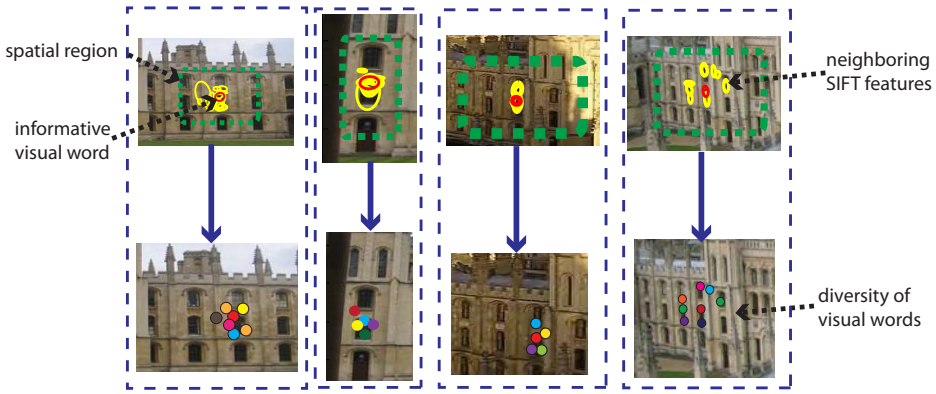


Figure 4.4: A visual word with a high entropy should be strongly weighted, as happens in different images of All souls. The first row shows various features (indicated in yellow) detected in neighbourhoods of a certain visual word (indicated in red). The second row shows close-up of some nearest neighbors (identified by color), which form a diverse nearest neighbor distribution. Note that all the visual words are found as inliers at least once in the training stage.

where the scaling factor δ is used to control the influence of $\alpha(w_i)$, $\delta \in (0,1)$ and is fixed in the experiments. We use an object-based thesaurus to store the spatial co-occurrence found during the training stage, and for further calculation of the entropy $H(\mathbf{D}_i)$.

The histograms in the object-based thesaurus indicate the co-occurrence of inlier words. The histograms used in this section are slightly different from those described in Section 3.5. We use a Gaussian mask to calculate the co-occurrence of visual words in the thesaurus histogram $h_i \in \mathbf{H}$, in which each entry $h_i(j)$ is incremented as follows:

$$h_i(j) = h_i(j) + \exp(- \parallel \mathrm{loc}(w_i) - \mathrm{loc}(w_j) \parallel / \sigma) \tag{4.6}$$

where $w_j \in \mathbf{D}_i$, σ is the scaling function and the function $\mathrm{loc}(w_i)$ recovers the (x,y) location of feature assigned to visual word w_i in image domain. Thus, the entropy of each word $w_i \in \mathbf{W}_S$ is calculated by the function $H(\mathbf{D}_i)$, where $\mathbf{D}_i$ can be found from the thesaurus histogram h_i.

Re-weighting with query words Furthermore, the inlier words in query are essential in retrieval. We propose a query specification mechanism to encourage

Algorithm 6 Visual word re-weighing scheme.

1: **Input**: Query words $\mathbf{Q}$, the foreground words $\mathbf{W}_S$, query vector $\mathbf{q}$ and candidate vector $\mathbf{d}$.
2: **Output**: Re-weighted similarity score $\Psi(\mathbf{q}^*, \mathbf{d}^*)$.

3: **for all** $w_i \in \mathbf{Q} \bigcap \mathbf{W}_S$ **do**
4: Adjust the tf-idf score of w_i according to Eq. (4.7).
5: $\mathbf{q}^* \leftarrow \mathbf{q}$ and $\mathbf{d}^* \leftarrow \mathbf{d}$.
6: **end for**
7: Calculate the similarity score $\Psi(\mathbf{q}^*, \mathbf{d}^*)$.

the words that appear both in the query words $\mathbf{Q}$ and the foreground words $\mathbf{W}_S$. Such words are important for a query since they can be shared by other images in the dataset, which also contain the same query objects. Thus, we slightly increase the weights of such words in computing the similarity:

$$\tau^*(w_i) = \begin{cases} \lambda \cdot \alpha(w_i) \cdot \tau(w_i) & \text{if } w_i \in \mathbf{Q}^* \\ \alpha(w_i) \cdot \tau(w_i) & \text{if } w_i \in \mathbf{W}_S, \text{ and } w_i \notin \mathbf{Q}^* \\ \tau(w_i) & \text{otherwise} \end{cases} \tag{4.7}$$

where $\mathbf{Q}^* = \mathbf{Q} \bigcap \mathbf{W}_S$, λ is a scaling factor, and the re-weighting function $\alpha(w_i)$ is predefined. Since $\mathbf{Q}^*$ is obtained online according to the given query, this step requires a slight cost in similarity computation. Usually, the length of $\mathbf{Q}^*$ is much smaller than the length of query vector $\mathbf{q}$. The computation complexity will increase very little. Algorithm 6 describes the details of our re-weighting method.

4.1.2 Experimental results

In this section, we investigate the effects of visual word re-weighting on three public datasets: Oxford 5K, Paris 6K and Oxford 105K datasets. We use $\mathbf{F}'_{15}$ in building the object-based thesaurus, where the number of nearest neighbors is fixed, the distance threshold $\rho = 30$ pixels and the scaling factor for Gaussian mask $\sigma = 15$ (mentioned in Eq. (4.6)). The setting of object-based thesaurus considers the wide variety of visual words co-occurring as inliers. The effects are examined in the following aspects:

Training data size		mAP: Oxford 5K				
# image pair	# inliers	α_1	α_2	α_3	α_4	α_5
Baseline(0K)	0K	0.612	0.612	0.612	0.612	0.612
2.0%L	5K	0.620	0.623	0.622	0.622	0.622
5.0%L	14K	0.623	0.630	0.625	0.629	0.628
10%L	21K	0.626	0.624	0.632	0.635	0.636
20%L	35K	0.642	0.638	0.644	0.643	0.650
40%L	53K	0.643	0.644	0.651	0.652	0.659
60%L	68K	0.642	0.647	0.650	0.653	0.658
80%L	83K	0.639	0.639	0.651	0.653	0.657
100%L	96K	0.642	0.642	0.650	0.655	**0.660**

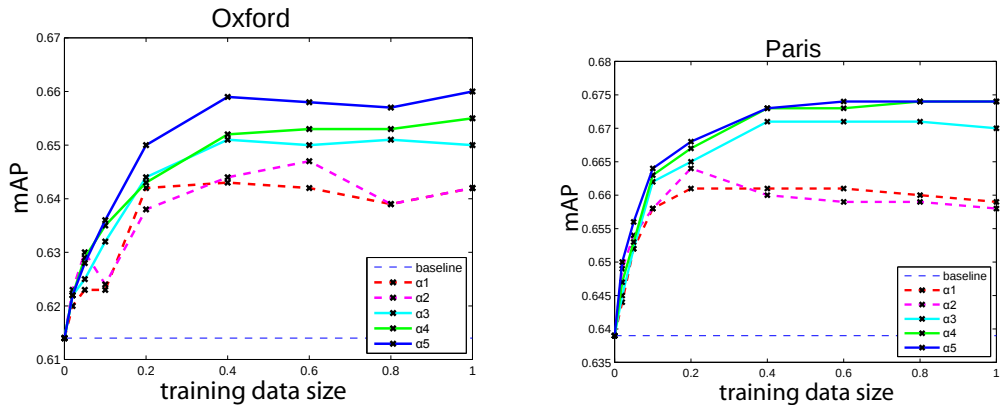
Table 4.1: Evaluation of different training data size on the Oxford 5K dataset. We compare the results with different types of α. The total number of training pairs is $L = 6.4K$.

Training data size		mAP: Paris 6K				
# image pair	# inliers	α_1	α_2	α_3	α_4	α_5
Baseline(0K)	0K	0.639	0.639	0.639	0.639	0.639
2.0%L	5K	0.644	0.649	0.645	0.647	0.650
5.0%L	9K	0.652	0.654	0.652	0.653	0.656
10%L	16K	0.658	0.658	0.662	0.663	0.664
20%L	31K	0.661	0.664	0.665	0.667	0.668
40%L	53K	0.661	0.660	0.671	0.673	0.673
60%L	67K	0.661	0.659	0.671	0.673	**0.674**
80%L	85K	0.660	0.659	0.671	**0.674**	**0.674**
100%L	102K	0.659	0.658	0.670	**0.674**	**0.674**

Table 4.2: Evaluation of different training data size on the Paris 6K dataset. We compare the results with different types of α. The total number of training pairs is $L = 7.6K$.

Effects of training data size Tables 4.1 and 4.2 investigate the effects of training data size on the Oxford 5K and Paris 6K datasets. To simplify the comparison, we start with the re-weighting function as $\alpha(w_i) = 2$, and then generate training data with the method presented in Section 3.5 in Chapter 3. After collecting 10% of visual words from the vocabulary (detected as inliers), we stop the process of training data generation. During this step, the mAPs for both the Oxford and Paris datasets rise for small amount of training data, and then plateau if given more training data, as the red dashed curves show in Figure 4.5. This is because our method relies on the spatial correspondence between image pairs. Once there is sufficient data to estimate this, retrieval performance is stable.

Effects of re-weighting function We examine the performance of visual word re-weighting with five different re-weighting functions $\alpha_{i,i=1:5}$, which can be



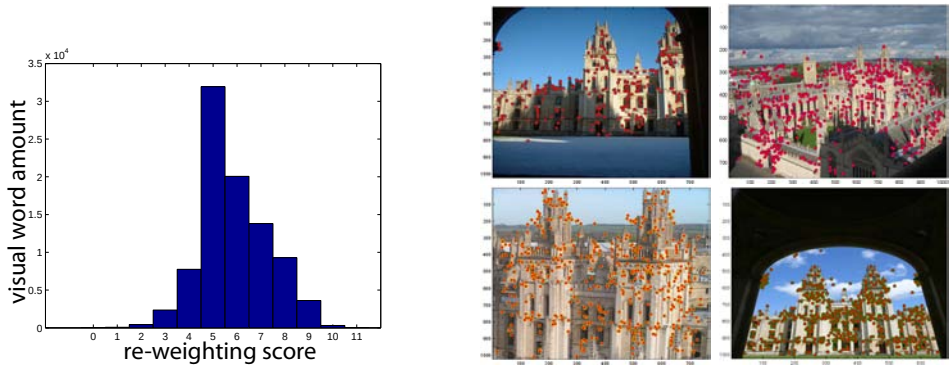
(a) Oxford 5K (b) Paris 6K

Figure 4.5: Evaluation of the retrieval performance on the Oxford 5K and Paris 6K datasets: i): training data size and ii) six different types of re-weighting function $\alpha(w_i)$.

grouped into two categories: non-entropy (α_1 and α_2) and entropy (α_3, α_4 and α_5). These re-weighting functions incrementally introduce knowledge of the foreground visual words.

- $\alpha_1(w_i) = 2$, there is no difference between the importance of the foreground words $\mathbf{W}_S$.
- $\alpha_2(w_i) = \exp(P_f(w_i))$, the importance of each visual word $w_i \in \mathbf{W}_S$ is a function of $P_f(w_i)$, the number of times it appears as inliers in the training data, as a fraction of the total number of times it appears. Therefore, $P_f(w_i) \in (0,1]$ and $\alpha_2 \in (1,e]$.
- $\alpha_3(w_i) = \exp(\delta \cdot H(\mathbf{D}_i))$, the importance of each visual word $w_i \in \mathbf{W}_S$ is proportional to the entropy in its neighborhood word set $\mathbf{D}_i$.
- $\alpha_4(w_i) = \exp(\delta \cdot P_f(w_i) H(\mathbf{D}_i))$, α_4 incorporates the prior information into the entropy. Thus, α_4 is a weighted version of α_3.
- α_5: query specification on α_4 (Section 4.1, Eq. (4.7)), it slightly increases the weight of the words $\mathbf{Q}^*$, based on the results of α_4.

In our experiments, the scaling factor δ for the re-weighting functions is set to 0.5. This is based on the highest entropy we observe from the experiment. Ideally, the maximum entropy happens if a nearest neighbor $\mathbf{D}_i$ include all the words $\mathbf{W}_S$, and each word $w_j \in \mathbf{D}_i$ has the same probability to appear ($\frac{1}{m}$, where $m = 10^5$ is the size of the foreground words $\mathbf{W}_S$). Therefore, the maximum entropy



(a) (b)

Figure 4.6: Illustration of tf-idf score adaption. (a) The frequency of α_4 re-weighting scores assigned to visual words. (b) The visual words for which $\alpha_5 > 4$. Note that most occur on the foreground object. The query specification is based on α_4 shown in (a).

is $H(\mathbf{D}_i) = -\sum_{j=1}^{m} \frac{1}{m} \cdot \log \frac{1}{m} = -m \cdot \frac{1}{m} \cdot \log \frac{1}{m} = 5$. The scaling factor λ for query specification is set to 1.15. Note that these re-weighting functions start with no spatial relationship (α_1), to weak spatial information (α_2), and to more strong spatial information using entropy (α_3, α_4, and α_5). The corresponding retrieval performances of these re-weighting functions are reported in Figure 4.5. As is seen in Figure 4.5, we obtain two groups of curve results, according to the non-entropy functions and the entropy functions. The mAPs for all the re-weighting functions rise rapidly for small amounts of training data (from zero to 20%L). The entropy functions lead to superior results over the non-entropy functions when the size of training data increases (from 20%L to 100%L). The mAPs for the entropy functions can still rise after using more than 20% training data, and then plateau with enough training data (about 40%L). On the contrary, the retrieval performance of the non-entropy function does not significantly improve after above 20% of the training data. The best results are obtained by α_5, and this will be used in the following experiments.

Effects of tf-idf score adaptation Figure 4.6 illustrates the distribution of re-weighting scores across different visual words. Note that the tf-idf score adaption does not need to be applied to all the visual words. Instead, it is applied to a subset of visual words in the vocabulary (10% of the visual vocabulary that have occurred in the training data). According to Figure 4.6 (a), many visual words in $\mathbf{W}_S$ will

be heavily weighted if they appear either in query and dataset image (computed offline by α_4). During the run time, query specification (α_5) increases the tf-idf scorer based on α_4 to re-weight the visual words. These highly re-weighted visual words ($\alpha_5 > 4$) are shown in Figure 4.6 (b), which mainly correspond to the geometric information of the buildings. We also notice that the number of the nearest neighbors influences the performance of visual word re-weighting, but the difference is small.

4.1.3 Discussion

Figure 4.7 shows the top ranked retrieval results of the visual word re-weighting method. With the refinement of weighting scheme, the standard similarity measure (normalized dot product) is able to more accurately describe the visual distance between a pair of query/dataset images. The retrieval accuracy is thereby increased. The comparison to state-of-the-art will be discussed in next section, together with an associated scheme of spatial expansion and visual word re-weighting.

4.2 Spatially aware feature selection and re-weighting

In this section, we consider both relatedness and importance of the visual words learnt from an object-based thesaurus. This is proceeded by two steps. Firstly, we recover spatially related words from an object-based thesaurus online. This is the spatial expansion method described in Chapter 3. Secondly, the weights of these spatially related words, together with the query words, are adjusted according to their importance learnt offline. The objective of each step is different. Like query expansion, spatial expansion is designed to improve image representation by adding extra relevant words to a query vector. The visual word re-weighting is designed to improve the precision in image ranking. Both are based on object-based thesaurus, the visual words to be expanded or re-weighted are from foreground and thus reduce the error illustrated in Figure 3.8.

Therefore, we present a **total association** scheme to combine the benefits of spatial expansion and visual word re-weighting. The total association scheme is illustrated in Figure 4.8. It has two stages:

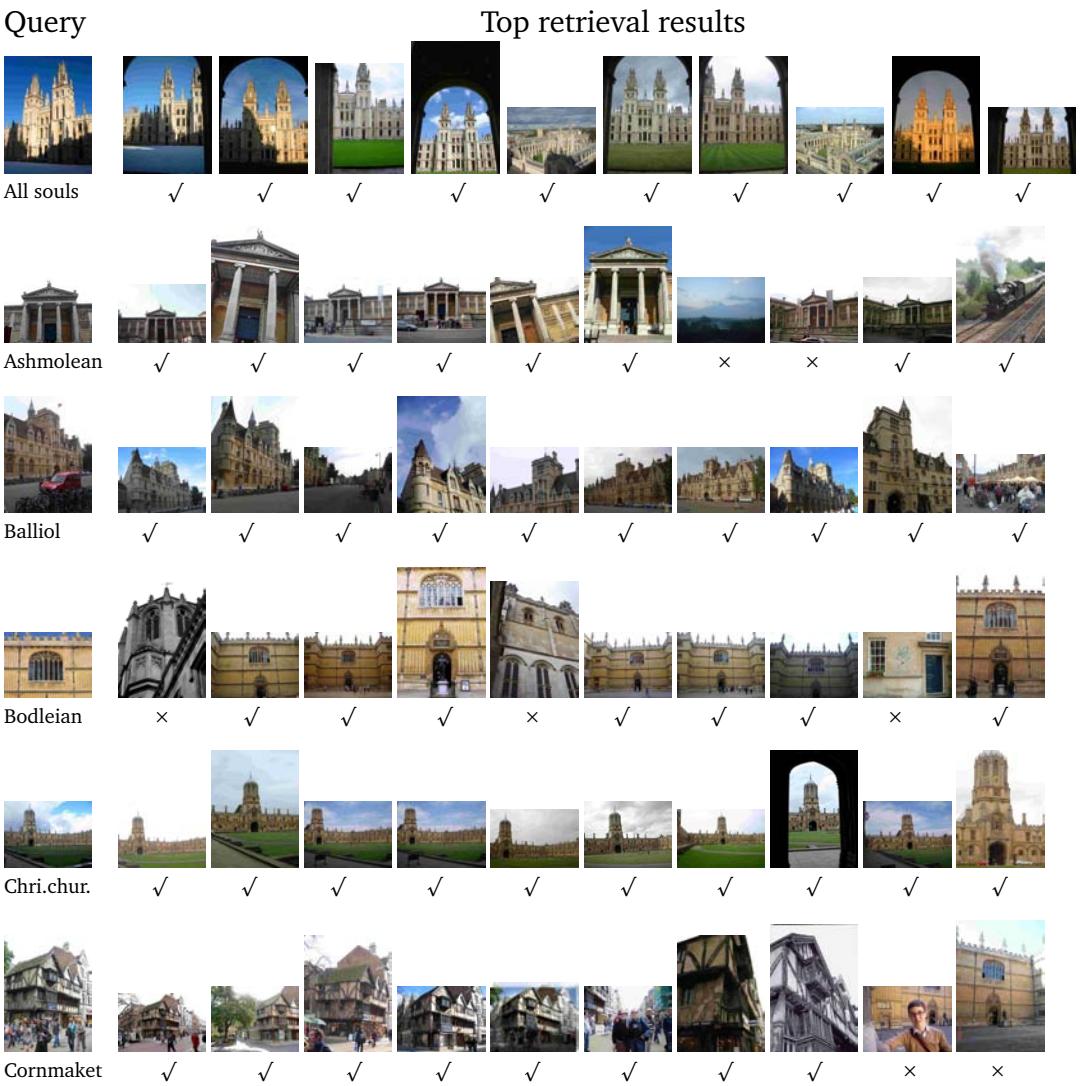
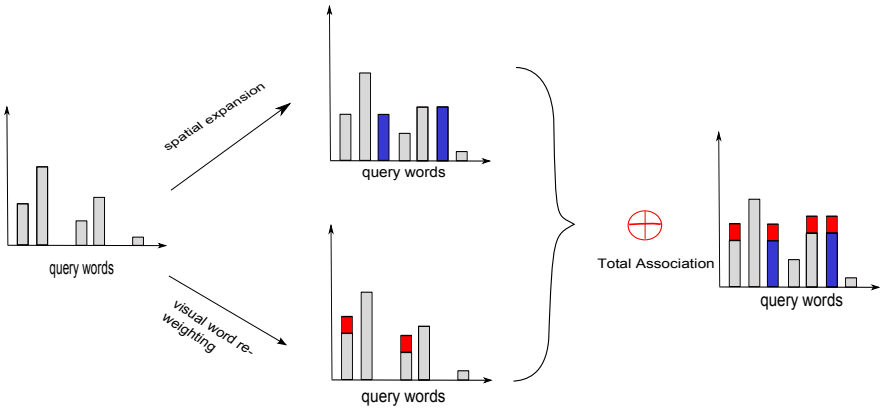


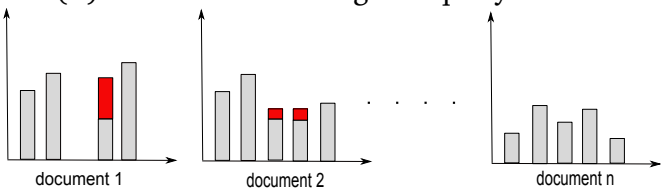
Figure 4.7: Top retrieval results of the visual word re-weighting method.

- offline: re-weight the visual words $w_i \in \mathbf{W}_S$ in all the documents according to the entropy $H(\mathbf{D}_i)$, as shown in Figure 4.8 (b).
- online: given a set of query words $\mathbf{Q}$, it will be first expanded with latent visual words to be $\mathbf{Q}' \leftarrow [\mathbf{Q}, \mathbf{W}_T]$ and then $\mathbf{Q}'$ will be re-weighted according to the entropy $H(\mathbf{D}_i)$ as well, as shown in Figure 4.8 (a).

Algorithm 7 describes the outline of our method. The offline process aims to collect foreground visual words and weights them in terms of their occurrence in the whole dataset. In this stage, the weights of the visual words are adjusted according to their importance in the images. The online process stage aims to



(a) Online: refine the given query words.



(b) Offline: refine the visual words in all the documents.

Figure 4.8: Illustration of the total association scheme. Gray bins: original word weights. Blue bins: refinement with spatial expansion. Red bins: refinement with visual word re-weighting. (a): online re-weighting process. (b): offline re-weighting process. The offline process firstly collects foreground words in an object-based thesaurus, as described in Chapter 3. It then computes the re-weighting scores of all visual words. The online process expands the given query words (in blue bins) and then re-weights them (gray and blue bins together) to a higher weight in the vector space (red bins).

retrieve the relevant images by matching the visual words vectors. In this stage, the query vector is expanded and re-weighted with the knowledge learned from the offline stage. Figure 4.8 describes the work flow of our association method. This combined scheme can cover more than one situation in which tf-idf based image retrieval fails. Firstly, latent visual words can be added into query if they are found; secondly, discriminative words can be weighted effectively.

4.2.1 Experimental results

We investigate the effects of total association on three public datasets: Oxford 5K, Paris 6K and Oxford 105K as follows:

Algorithm 7 Outline of total association method.

Require: Image dataset and queries.

Build an object-based thesaurus (offline)

1. Automatic training data collection via RANSAC [108] (Section 3.5).
2. Build an object-based thesaurus from the training data, which only considers the points that were detected as *inliers* in RANSAC.

Visual word re-weighting (offline)

1. Calculate the entropy H from object-based visual thesaurus (Section 4.1).
2. Re-weight the visual words in dataset images according to their entropy (Section 4.1).

Spatial expansion based on object-based thesaurus (online)

1. Expand the query visual words from object-based thesaurus. The expansion is based on frequent co-occurrence of visual words (Section 3.5.3).
2. Re-weight the expanded query words according to their entropy (Section 4.1).
3. Compute the similarity of dataset images to the query by the re-weighted BoW vectors.

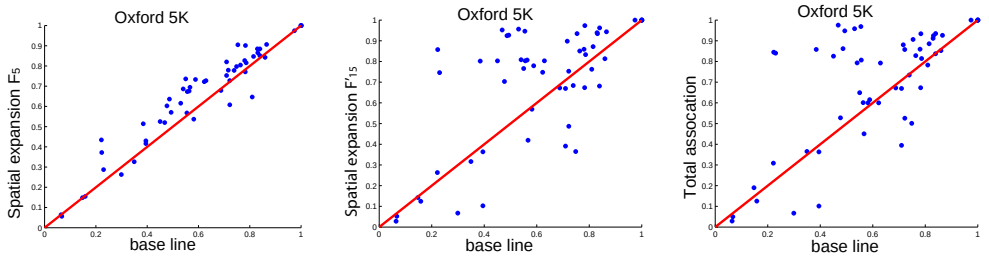
return The retrieval results.

Effects of total association The full scheme combines spatial expansion and visual word re-weighting. Table 4.3 reports the details of the mAP results of the association scheme. As seen in Table 4.3, most queries have significant improvement over baseline. However, it does not obtain further improvement in retrieval accuracy (mAP) compared to the spatial expansion $\mathbf{F}\prime_{15}$. This is because most of the benefit is from expansion of foreground words $\mathbf{W}_S$ rather than adjusting the weights of them. Figure 4.9 compares retrieval results of 55 individual queries on the Oxford 5K dataset. The improvement of retrieval performance is indicated by the points located above the diagonal. As seen in Figure 4.9 (a) and (b), the object-based thesaurus $\mathbf{F}\prime_{15}$ makes the retrieval results of some queries varied dramatically. As discussed in Section 3.5.3, this is because an object-based thesaurus relies on spatial transform, which may associate foreground words to multiple objects. The re-weighting scheme further adjusts the weights of these foreground words. As seen in Figure 4.9 (b) and (c), the difference between total association (after re-weighting) and spatial expansion $\mathbf{F}\prime_{15}$ is small. We illustrate some detailed PR curves in Figure 4.10. As seen from these PR curves, spatial expansion $\mathbf{F}\prime_{15}$ improves the retrieval accuracy compared to baseline (increased area under the PR curves), while the further re-weighting (total association) has less effects than expansion of foreground words $\mathbf{W}_S$. However, total association

Oxford 5K	mAP	
Ground truth	Baseline	Total association
All Souls	0.544	0.750
Ashmolean	0.617	0.721
Balliol	0.563	0.362
Bodleian	0.456	0.938
Chri. Chur.	0.589	0.815
Cornmarket	0.584	0.548
Hertford	0.816	0.915
Keble	0.775	0.773
Magdalen	0.186	0.140
Pitt River	0.995	1.00
Radc. Camb.	0.609	0.723
Total	0.612	0.700

Paris 6K	mAP	
Ground truth	Baseline	Total association
Defense	0.419	0.459
Eiffel	0.463	0.506
Invalides	0.643	0.719
Louvre	0.380	0.378
Moulinrouge	0.607	0.523
Museedorsay	0.546	0.678
Notredame	0.807	0.900
Pantheon	0.920	0.957
Pompidou	0.916	0.898
Sacrecoeur	0.826	0.929
Triomphe	0.507	0.549
Total	0.639	0.682

Table 4.3: The retrieval results of total association on the Oxford 5K and Paris 6K dataset, compared to the baseline method.



(a) (b) (c)

Figure 4.9: Comparison of various spatial expansion to the baseline. Each point represents an individual query of the 55 queries on the Oxford 5K dataset. (a) comparison of spatial expansion $\mathbf{F}_5$ to the baseline. (b) comparison of spatial expansion $\mathbf{F}\prime_{15}$ to the baseline. (c) comparison of total association to the baseline. The points above (below) the diagonal illustrate the increase (decrease) of retrieval accuracy.

achieves further improvement on the large scale dataset (Oxford 105K), compared to spatial expansion $\mathbf{F}\prime_{15}$, as reported in Table 4.7. This is because the object-based thesaurus helps to avoid background information, which is more essential on large scale dataset. Thus total association takes the advantage of both re-weighting and expansion on the Oxford 105K dataset. We adopt total association as our proposed method in the following experiments.

Comparison with post-processing methods Tables 4.4 and 4.5 compare the accuracy and run time with commonly used post-process methods (spatial verification [106] and average query expansion (AQE) [37] methods). The retrieval

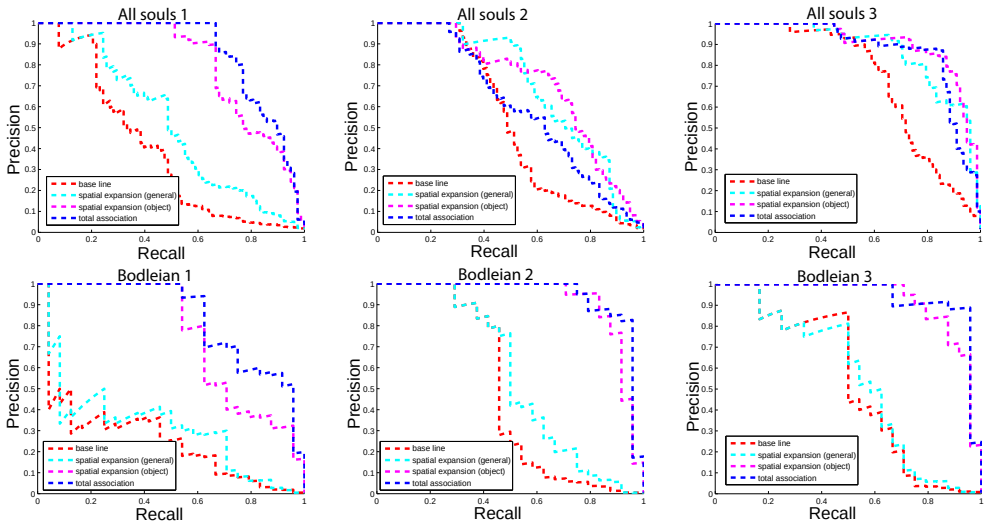


Figure 4.10: Illustration of detailed precision-recall (PR) curves of total association.

results in Table 4.4 are reported in three groups (A-C): Group A compares each step of our method (spatial expansion, visual word re-weighting and total association) with the baseline results, respectively. At each step, our method can outperform the baseline with some extra run time as reported in Table 4.5. The increase of run time is at most double that of the baseline. In particular, using visual word re-weighting alone requires very little extra computation time during query (almost the same as the baseline). Group B reports our method jointly working with spatial verification [106]. As seen in Tables 4.4 and 4.5, our method in Group A can outperform spatial verification without the need for query time processing. We can also jointly use spatial verification with each step of our method as shown in Group B, and obtain further improvement on retrieval accuracy. Group C reports our method jointly working with AQE [37], which requires a spatial verification to refine the query model. This further improves retrieval results compared with the results in Groups A and B. However, the performance gain of these combined methods in this group is small. This is because AQE needs to examine spatial consistency between features, which has already been exploited partly by an object-based thesaurus offline. The accuracy of these top verified results is high for all methods. Moreover, the collection of top verified results is expensive, as reported in Table 4.5 it requires more than 2 seconds using spatial verification. In contrast, our method learns the latent spatially relevant words offline, which is much faster than AQE.

	Methods	V	R	S	TR	Oxford 5K	Paris 6K	105K
	Baseline					0.612	0.639	0.515
A	Spatial expansion ($\mathbf{F}'_{15}$)	√				0.701	0.683	0.667
	Visual word re-weighting		√			0.660	0.674	0.598
	Total association	√	√			0.700	0.682	0.680
B	Spatial verification [106]			√		0.649	0.655	0.571
	Spatial expansion ($\mathbf{F}'_{15}$)	√		√		0.719	0.689	0.704
	Visual word re-weighting		√	√		0.677	0.684	0.611
	Total association	√	√	√		0.710	0.686	0.706
C	QE baseline [37]				√	0.708	0.736	0.679
	AQE [37]			√	√	0.800	0.769	0.767
	Spatial expansion ($\mathbf{F}'_{15}$)	√		√	√	0.806	0.785	0.783
	Visual word re-weighting		√	√	√	0.801	0.777	0.781
	Total association	√	√	√	√	0.804	0.785	0.774

Table 4.4: Comparison with methods requiring spatial consistency examination. S denotes spatial verification used as a post processing, V denotes spatial expansion, R denotes visual word re-weighting and TR denotes re-querying with AQE. Group A: comparison with baseline. Group B: comparison with spatial re-ranking. Group C: comparison with query expansion (AQE).

Method	Oxford 5K	Paris 6K	Oxford 105K
Baseline [106]	0.107	0.140	1.67
Spatial expansion ($\mathbf{F}_5$)	0.144	0.209	1.93
Spatial expansion ($\mathbf{F}'_{15}$)	0.136	0.185	1.85
Visual word re-weighting	0.118	0.147	1.74
Total association	0.147	0.198	1.88
Spatial verification [106]	2.10	4.71	4.34
AQE [37]	2.43	5.47	8.19

Table 4.5: Average run time of retrieval methods, measured by CPU second.

Comparison with pre-processing methods Table 4.6 compares our methods with those requiring training/learning stage as a pre-process. Methods without prior training data collection (a-b) usually need to re-organize query words. For example, method (a) needs to include more visual words than the baseline method to expand the coverage of each individual word, while method (b) organizes visual words into phrases. In contrast, methods with prior training data collection (c-f) have the advantage of selecting a subset of visual words (features). These methods (c-f) can retrieve the given query image almost as fast as the baseline methods. Our method (e) can outperform the other pre-processing methods in terms of retrieval accuracy, except method (f). This method (f) requires investigating the relationship back to the raw feature level, while our method (e) only needs to investigate visual word relationship.

	Method	F	Oxford 5K	Paris 6K
a	Soft-assignment [107]		0.673	0.660
b	Geometry-Preserving [159]		0.696	N/A
c	Descriptor learning (non-linear) [108]	√	0.662	0.678
d	**Visual word re-weighting**	√	0.660	0.674
e	**Total association**	√	0.700	0.682
f	Fine vocabulary [96]	√	0.742	0.749

Table 4.6: Comparison of our methods to those that modify the baseline before the query is executed. F denotes a training data collection needed before hand.

4.2.2 Discussion

Figure 4.11 shows some examples of top retrieved results returned by total association of spatial expansion and visual word re-weighting. As seen in Figure 4.11, the re-weighting of these spatially expanded words helps to remove some false positives, for example *Bodleian*. As discussed in Chapter 3, spatial expansion helps to find correlated visual words, while visual word re-weighting helps to weight heavily the discriminative words. To achieve the improvement of both precision and recall, the total association balances these two methods. Recent work [149] has proposed a similar method to boost the discriminative ability in a fixed spatial region (spatial contextual weighting). However, the method proposed in [149] only focuses on boosting the precision, and applies to the feature space.

We compare our total association method to a number of state-of-the-art methods in Table 4.7. Group A compares methods without post-processing. The results show that total association can outperform many previous methods in this group. Also, our method can outperform some re-ranking methods, *e.g.* spatial verification, without a post-process. Group C illustrates our methods jointly working with spatial verification and various query expansion methods. As seen in Group C, the spatial verification can slightly further improve the retrieval results, while the effects of joint work with AQE and DQE is not evident. For example, total association with DQE achieves mAP scores as 0.816 on the Oxford 5K dataset, while individual DQE achieves mAP score as 0.798 on the same dataset. This is because using an object-based thesaurus has already exploited this spatial consistency information offline. As a result, the performance gain is little when combining object-based thesaurus with AQE or DQE, where the effects of these methods are redundant. Our methods with AQE and DQE also have similar performance to other methods based on query expansion, *e.g.* Contextual synonym

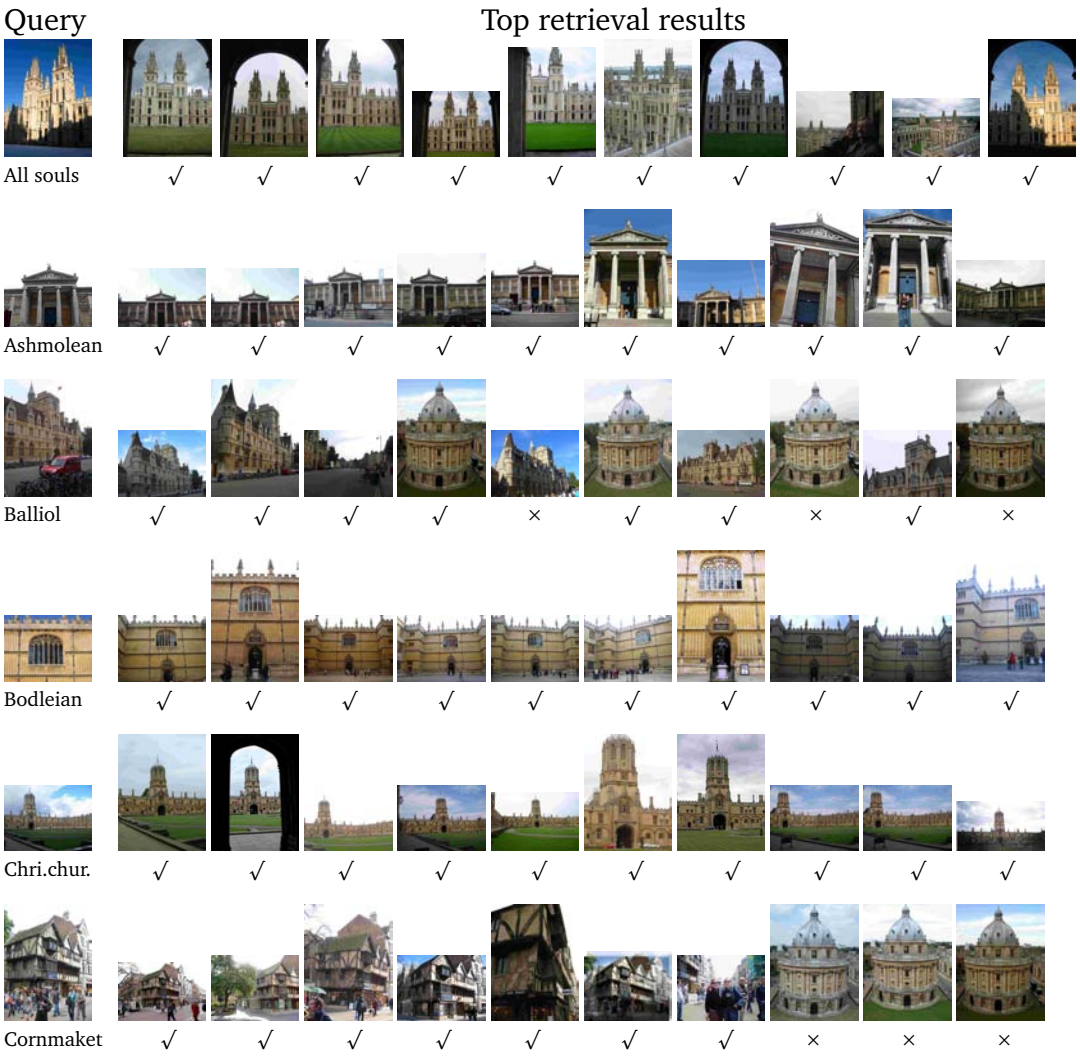


Figure 4.11: Top retrieval results of total association scheme.

dictionary [138], with minimal run time overhead. It is difficult, however, to directly compare the improvement that is due to each of these methods as they are based on different baseline implementations: the baseline BoW method in [138] has approximately 10% higher mAP than the standard used in this thesis and [106, 37, 108].

4.3 A cross-word matching measure via visual thesaurus

We have investigated several methods to improve the standard BoW retrieval system: the spatial expansion refines the BoW representation of the query by

	Methods	Oxford 5K	Paris 6K	Oxford105K
Baseline [106]		0.612	0.639	0.515
A	**Visual word re-weighting**	0.660	0.674	0.598
	Descriptor learning (non-linear) [108]	0.662 [108]	0.678 [108]	0.541 [108]
	Soft-assignment [107]	0.673 [107]	0.660 [107]	N/A
	Spatial expansion ($\mathbf{F}_5$, Chapter 3)	0.685	0.679	0.622
	Geometry-Preserving [159]	0.696 [159]	N/A	0.604 [159]
	Total association	0.700	0.682	0.680
	Spatial expansion ($\mathbf{F}'_{15}$, Chapter 3)	0.701	0.683	0.667
	Fine vocabulary [96]	0.742 [96]	0.749 [96]	N/A
	AUG [142]	0.776 [9]	N/A	0.711 [9]
	SPAUG [9]	0.785 [9]	N/A	0.723 [9]
B	Spatial verification [106]	0.649	0.655	0.571
	QE Baseline [37]	0.708	0.736	0.679
	iSP [34]	0.741 [34]	0.769 [34]	0.649 [34]
	Local geometry [105]	0.788 [105]	0.634 [105]	0.725 [105]
	AQE [37]	0.806	0.769	0.767
	DQE [9]	0.798	0.783	0.809
	Hello neighbors [114]	0.814 [114]	0.803 [114]	0.767 [114]
	Total recall II [34]	0.827 [34]	0.805 [34]	0.767 [34]
C	Spatial expansion ($\mathbf{F}'_{15}$)	0.701	0.683	0.667
	Spatial expansion+ Spatial verification	0.719	0.689	0.704
	Spatial expansion+ AQE	0.806	0.785	0.783
	Spatial expansion+ DQE	0.813	0.789	0.818
	Visual word re-weighting	0.660	0.674	0.598
	Visual word re-weighting + Spatial verification	0.677	0.684	0.611
	Visual word re-weighting + AQE	0.801	0.777	0.781
	Visual word re-weighting + DQE	0.811	0.782	0.787
	Total association	0.700	0.682	0.680
	Total association + Spatial verification	0.710	0.690	0.706
	Total association + AQE	0.804	0.785	0.774
	Total association + DQE	0.816	0.790	0.817
	Contextual synonym dictionary + AQE [138]	0.811 [138]	0.791 [138]	0.797 [138]

Table 4.7: Comparison of the total association to the state-of-the-art methods. Group A: retrieval results of methods that modify the baseline before the query is executed (pre-process). Group B: retrieval results of methods that modify the baseline after the query is executed (post-process). Group C: comparison of methods jointly working with spatial verification and various query expansion methods. Note that we cite the retrieval results of AUG [142] from literature [9].

introducing spatially related words (Chapter 3); the visual word re-weighting improves the similarity measure by balancing the importance of visual words (Section 4.1); and an association scheme to utilize both effects of these two methods (Section 4.2); In these methods, the image similarity is the normalized dot product of BoW vectors (Eq. (4.1)), which is a word-to-word matching. As discussed at the start of this chapter, the standard dot product similarity measure is efficient in matching the image vectors, but intolerant to the quantisation errors.

In this section, we propose a cross-word image matching method such that similar features assigned to different visual words can contribute to the similarity measure. More specifically, a cross-word matching aims to overcome the quantisation errors by introducing nearest neighbors of a visual word, similar

to the Earth Mover's Distance (EMD) [117]. Thus, for a given query/dataset image pair (q, d), we modify similarity measure as follows:

$$\text{sim}(q, d) = \Psi(q, d) + \Gamma(q, d) \tag{4.8}$$

In Eq. (4.8), the dot product similarity $\Psi(q, d)$ is a word-to-word matching as defined in Eq. 4.1, which only contributes to the image similarity when features are mapped in the same word ID. In contrast, the cross-word matching $\Gamma(q, d)$ contributes when features are not mapped in the same word ID but are close in feature or image space. Therefore, the cross-word matching involves nearest neighbors according to various distance measures, which can be (not limited to): i) the Euclidean (L2) distance in the feature space; ii) a visual thesaurus in the image space; iii) in the topic space, the visual word similarity should be based not just on proximity in feature space, but also on their association with the same object.

Our method is related to a couple of state-of-the-art methods in improving the BoW retrieval system. Firstly, our method is an alternative of soft-assignment [107]. Both cross-word matching and soft-assignment aim to improve the BoW retrieval system by exploring inter-word relationships. The soft-assignment method does this by mapping each feature to multiple visual word IDs according to the L2 distance in feature space. In contrast, our method does not need multiple visual word IDs for each feature representation, instead it utilizes a cross-word matching during similarity calculation. Thus, our method is also related to the Earth Mover's Distance (EMD), proposed in [67] in the context of image retrieval, but is computationally simpler, reducing its query time overhead. Moreover, the new image similarity measure can be integrated with many other proposed enhancements to the basic BoW retrieval method, with little cost in terms of implementation or run time performance. For example, our method can combine with the techniques such as spatial verification, and query expansion, and we demonstrate that it can benefit the baseline implementation of each of these techniques. The system framework of our cross-word similarity measure is shown in Figure 4.12.

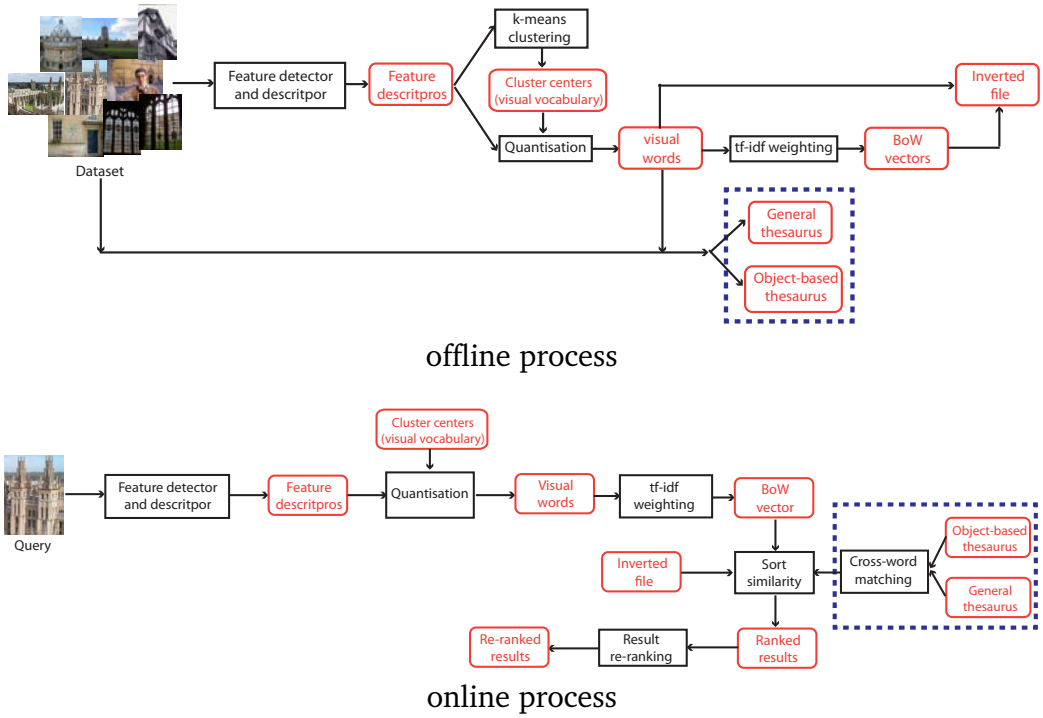


Figure 4.12: System framework of cross-word matching. This is the standard BoW retrieval system with new steps (indicated in dash box) introduced by our method.

4.3.1 A cross-word image similarity measure

The image ranking method in our method uses the cross-word matching to more accurately sort the dataset images. This involves two issues: i) to explore multiple correspondences of visual words, which requires nearest neighbor search of visual words. ii) to optimize the weight for the one-to-multiple correspondences such that the total cost is minimized.

In order to address the first issue, nearest neighbor search of visual words considers various distance measures, which will be discussed in Section 4.3.2. The second issue is an optimization problem, which aims to minimize the cost of matching a query word to its near neighbors (if they exist in a dataset image). Like a BoW retrieval system, we separate the optimization in two stages: offline and online. The offline stage learns weights from static distance, while the online stage only needs to calculate the matching.

Visual word weights from distances (offline)

For each word $w_i \in \mathbf{W}$, we can find its T nearest neighbors $\{w_t^i\}_{t=1}^T$ based on a given visual distance Θ (the L2 word distance, the spatial co-occurrence distance and the semantic distance, which will be discussed in Section 4.3.2). We propose a global weight vector $\mathbf{f}_i$ for each word $w_i \in \mathbf{W}$ such that the cost of moving w_i to its neighbors w_t^i is minimised according to a visual distance $\Theta(w_i, w_t^i)$ computed offline:

$$\begin{aligned} \min_{\mathbf{f}_i} \quad & \sum_{t=1}^T f_{it}\Theta(w_i, w_t^i) \\ s.t. \quad & f_{it} \geq 0,\ \sum_{t=1}^T f_{it} = 1 \end{aligned} \tag{4.9}$$

The optimization problem is minimised when all components $f_{it}\Theta(w_i, w_t^i)$ are equal, leading to a solution:

$$f_{it}^* = \frac{\frac{1}{\Theta(w_i, w_t^i)}}{\sum_{j=1}^T \frac{1}{\Theta(w_i, w_t^i)}} \tag{4.10}$$

The optimized weights $\mathbf{f}_i^* = (f_{i1}, f_{i2}, \cdots, f_{iT})$ of word w_i is therefore inversely proportional to a visual distance between w_i and its T neighbors. Therefore, the weights are a function of a selected inter-word distance Θ_m: f_{mit}^* where m indicates an inter-word distance discussed below. Unlike the weighting method in [67], the weights only need to be calculated once offline.

Cross-word matching similarity (online)

After obtaining the global weights f_{it}^* offline, they are used to estimate the cross-word matching between a query image q and a dataset image d. We only calculate cross-word image distance based on foreground words collected by the object-based thesaurus. This highly reduces the computation of cross-word image distance to a small subset of visual words. For images q and d, one can obtain two visual word subsets $\mathbf{v}_q' = \mathbf{v}_q \cap \mathbf{W}_S$ and $\mathbf{v}_d' = \mathbf{v}_d \cap \mathbf{W}_S$, in which $\mathbf{v}_q$ and $\mathbf{v}_d$ are the visual words contained in the images q and d, and $\mathbf{W}_S$ is found in Section 4.3.2. If a word w_i occurs in both $\mathbf{v}_q'$ and $\mathbf{v}_d'$, it already contributes to the dot product image similarity measure. We therefore remove all such words from both $\mathbf{v}_q'$ and $\mathbf{v}_d'$, so the resulting sets are disjoint. The cross-word matching is then to estimate

Algorithm 8 Image ranking with cross-word matching

Require: Image dataset and queries.

At the training stage (Section 4.3.2)

1. Calculate the tf-idf weight for each word $w_i \in \mathbf{W}$ (Eq. (2.8)).
2. Automatic training data collection: obtain the topics set $\mathbf{K}$ (Eq. (4.14)) and the subset of visual words $\mathbf{W}_S$ which appear as inliers .
3. Compute the visual distance Θ inside the subset $\mathbf{V}'$ (Eq. (4.12), Eq. (4.13) and Eq. (4.15)).
4. Use the visual distance to solve the visual weights $\mathbf{f}_i^*$ for each word w_i (Eq. (4.10)).

At query time

1. Calculate the word-to-word matching (dot product) $\Psi(q,d)$ between two images [130] (Eq. (4.1)).
2. Calculate the cross-word matching $\Gamma(\mathbf{v}'_q, \mathbf{v}'_d)$ between two images (Eq. (4.11)).
3. Rank the images using both word-to-word matching and cross-word matching measures: $\text{sim}(q,d) = \Psi(q,d) + \Gamma(q,d)$ (Eq. (4.16)).

return The retrieval results.

the similarity between these disjoint sets.

For each word $w_i \in \mathbf{v}'_q$, there are up to T nearest neighbours found in $\mathbf{v}'_d$ with visual distance measure Θ. The cross-word matching similarity between images q and d can be computed as follows:

$$\Gamma(q,d) \quad = \frac{1}{|\mathbf{v}'_q|} \sum_{i=1}^{|\mathbf{v}'_d|} \sum_{j=1}^{T} f_{ij}^* s_{ij} \tag{4.11}$$

in which $\mathbf{f}_i^*$ is the visual word weights for w_i and its neighbors, and s_{ij} is the corresponding scaled tf-idf weight difference for pairwise images: $s_{ij} = \exp(-\frac{\|\tau(w_i)-\tau(w_j)\|}{\rho})$ where $w_i \in \mathbf{v}_q$ and $w_j \in \mathbf{v}_d$ and ρ is the scaling factor.

The framework of our method is described in Algorithm 8. Our method is performed in two stages: the offline training stage and the online querying stage. At the training stage, a collection of foreground words are collected by an object-based thesaurus. At the query time, these visual words are used to calculate the cross-word matching if exist in a pair of images. As seen in Eq. (4.10) and Eq. (4.11), the visual distance Θ is essential to our cross-word method Γ. It affects the global weight vector $\mathbf{f}_i$, as well as the cross-word matching similarity. Therefore, it is important to define a visual word distance and use it to measure how similar two visual words are.

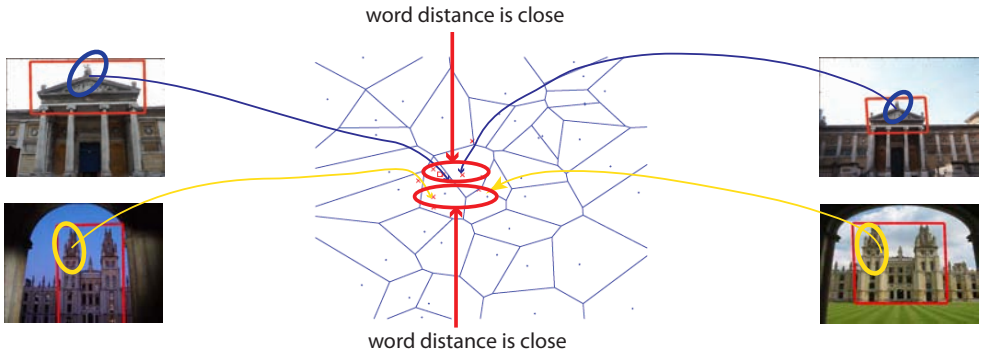


Figure 4.13: Example of the L2 word distance, which help to recover the relevance between similar features assigned to different visual words in the feature space. The illustration of quantisation in feature space is taken from [63].

4.3.2 Inter-word distance measure

We introduce three visual word distance measures to exploit the relevance of different visual words in this section. We formulate the distance between image features in several different spaces:

- Feature space, where the features are quantised as visual words;
- Image space, where each feature has an (x, y) location;
- Topic space, where the features are grouped according to their association to a common object.

These associations are based on different feature attributes, and therefore contribute to a more comprehensive image distance measure in the BoW model.

The L2 word distance

Firstly, we consider the distance measure in feature space, where the visual words (cluster centres) are generated by AKM, as described in Section 2.2.2. A simple distance measure between two words is the $L2$ distance between their locations in feature space (commonly 128 D SIFT space):

$$\Theta_1(w_i, w_j) = \| c(w_i) - c(w_j) \|_2 \tag{4.12}$$

where $c(w_i)$ is the corresponding feature vector of the word w_i (cluster centre). This distance measure has been used in soft assignment [107, 63], among other methods to modify the quantisation error of hard assignment. As shown in Figure 4.13, matched features that are incorrectly assigned into different visual words considered relevant under the definition of the $L2$ word distance (Eq. (4.12)). The $L2$ word distance is simple to compute and requires no pre-process, but does not take into account any semantic information.

The spatial co-occurrence distance

The $L2$ word distance sometimes causes inaccuracy because of quantisation, as noticed in [107]. This leads us to use other distance measures for an inter-word distance. As noted in Chapter 3, word pairs associated with the same objects are likely to co-occur in image regions where the object appears. Therefore, the spatial co-occurrence frequency of words in images can be used as a cue to measure the degree of association with a common object. This can be discovered by an object-based thesaurus structure. Based on this we define a spatial co-occurrence distance measure:

$$\Theta_2(w_i, w_j) = 1 - \frac{2 \times \chi(i,j)}{\sum_m \chi(i,m) + \sum_n \chi(j,n)} \tag{4.13}$$

where $\chi(i,j)$ is the co-occurrence frequency of word pair w_i and w_j in image space within a fixed size neighborhood[1]. The co-occurrence distance Θ_2 ranges from 0 to 1. This distance measure attempts to gauge semantic proximity information between visual words through co-occurrence, at the cost of an additional pre-processing step to generate the co-occurrence figures. However, the spatial co-occurrence information between visual words may not always be caused by association with the same object, so this is only an indirect measure of semantic similarity.

A semantic distance measure

Besides the low level distance measures discussed above, we consider high level topic information. We now introduce a "semantic" distance measure that explicitly

[1]Same as defined in the object -based thesaurus in Chapter 3: up to 15 nearest neighbors occurring within a radius of 50 pixels.

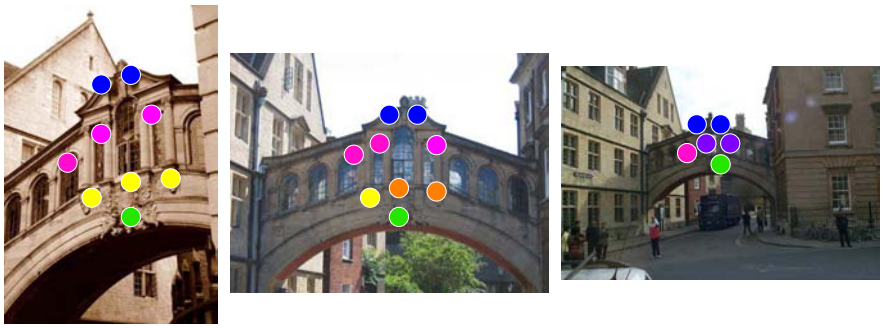
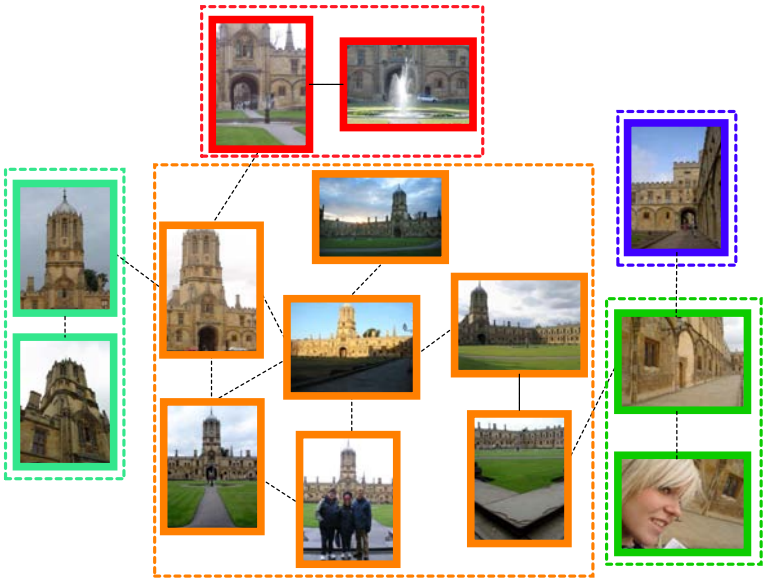


Figure 4.14: Example visual words for the same topic (*bridge of sighs*) with varying viewpoint. The features are mapped into different visual words, which are labeled by the color. However, these visual word should be considered to have small semantic distance, as they frequently happen on the same objects.

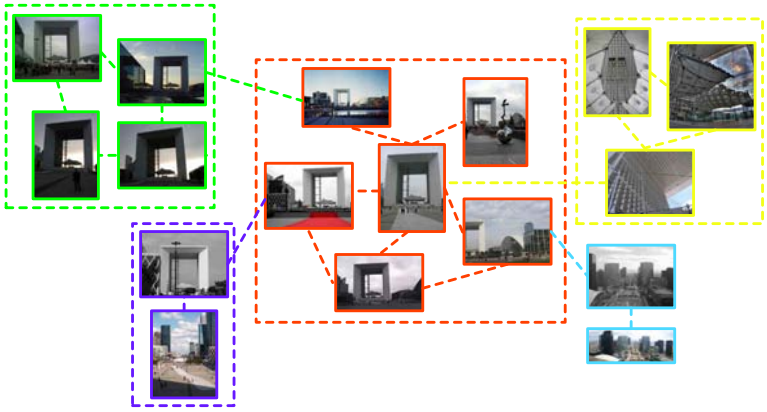
discovers several common objects in order to establish the underlying word-to-object association. As shown in Figure 4.14, the visual vocabulary contains several groups of words that are strongly associated with particular objects in the image collection. These associated visual words can be explored by robustly estimating a geometric transformation between image pairs via RANSAC [54]. Therefore, we can learn the “topic" of the visual words (i.e., the object they belong to) by clustering images according to their geometric consistency. As illustrated in Figure 4.15, all the associated images contain the same building, *Christ church*, but change significantly with viewpoint. In these images, some images are strongly connected, while others have loose connection. By using this matching graph to represent the images, we can convert the problem of topic discovery to that of graph clustering.

More specifically, we extract the topics from the image collection using an object-based thesaurus (as described in Section 3.5) and build a matching graph. In previous works [109, 110, 111, 7], a matching graph is built on the whole dataset. However, clustering the graph becomes expensive when the number of nodes enlarges. In contrast, we select a subset of images by an object-based thesaurus. Our scheme is composed of two modules: i) building a matching graph; ii) partition the matching graph, which are described in detail as follows.

Building a matching graph Based on an object-based thesaurus, a subset of visual words $\mathbf{W}_S \subset \mathbf{W}$ that occur as inliers are obtained from the dataset. Words



(a): built on the Oxford dataset



(b): built on the Paris dataset

Figure 4.15: Examples of the images transitively connected by the geometric information. (a): images transitively connected by the geometric information. The images are from *Christ church* change dramatically from viewpoint, but can be linked together. Note that the number of images in the orange group is larger than what we have shown here. (b): The images from *Defense* are linked together. The numbers of images in sub-graph are larger than what we have shown here.

that only appear as outliers, or do not appear at all, are discarded. The matching graph $\mathbf{G}$ is built on the basis of the spatial layout of visual words. The nodes of $\mathbf{G}$ are the images from the training data. Any two nodes i and j are connected if they are matched via RANSAC with a minimum number of verified inliers. Let $\mathbf{W}_G = (W_G(i,j))$ be the corresponding symmetrical matrix of $\mathbf{G}$. The edge weight $W_G(i,j)$ between nodes i and j is defined as the number of inliers, as proposed

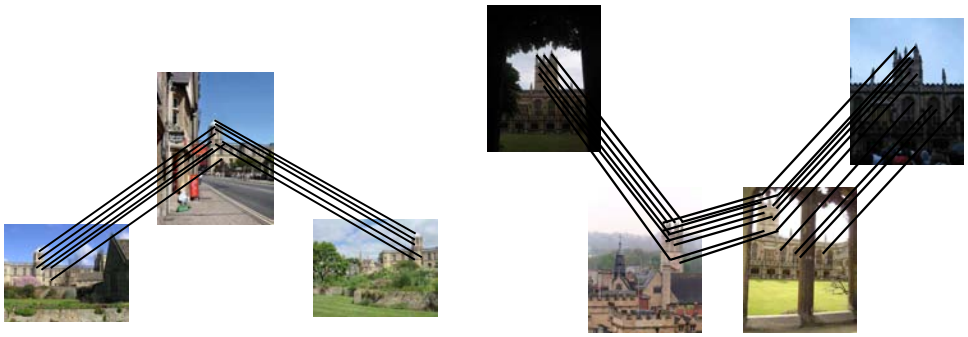


Figure 4.16: Examples of small connected component from the matching graph built on the Oxford 5K dataset. In these cases, each component is treated as a topic.

in [110]. Note that the nodes of **G** need not contain all the dataset images, instead, only the images detected from automatic training data collection are considered. This significantly reduces the size of matching graph.

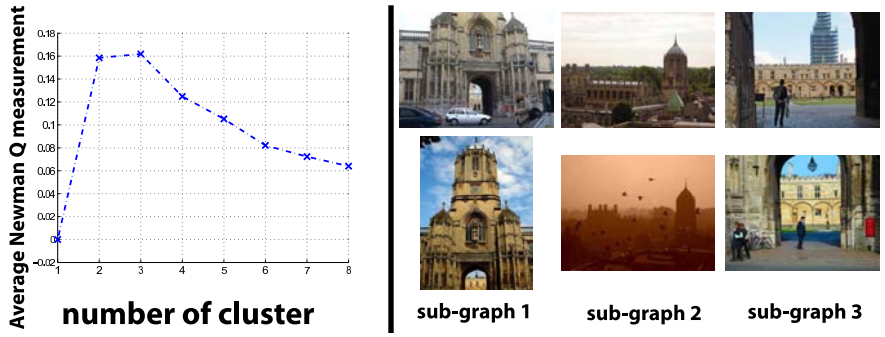
Partition the matching graph After building the graph **G**, the images can be automatically grouped as a set of connected components $\mathbf{C} := \{C_i\}_{i=1}^{M}$ by using a depth first search (DFS) strategy, where each component C_i is associated with a weight matrix $\mathbf{W}_{Gi}$ obtained from $\mathbf{W}_G$. As shown in Figure 4.16, the smaller components found by DFS have a very high probability to contain the same object and this component will be associated with a topic. The larger components often contain multiple objects[2], so they are further partitioned by Normalized-cut algorithm [125] into several sub-components, whose number is determined by maximizing the average Newman Q measure [111, 99]:

$$\max_{K_i \in [1,|C_i|]} \quad Q(C_i) = \frac{1}{K_i} \sum_{l=1}^{K_i} \left(\frac{\mathbf{X}_l^T \mathbf{W}_{Gi} \mathbf{X}_l}{\mathbf{1}^T \mathbf{W}_{Gi} \mathbf{1}} - \left(\frac{\mathbf{X}_l^T \mathbf{W}_{Gi} \mathbf{1}}{\mathbf{1}^T \mathbf{W}_{Gi} \mathbf{1}} \right)^2 \right) \tag{4.14}$$

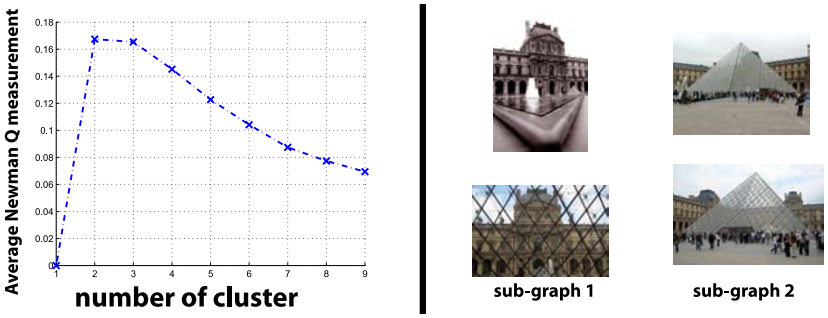
in which $\mathbf{1}$ is $|C_i| \times 1$ vector of all ones, and $\mathbf{X}$ is the indicator matrix with size $|C_i| \times K_i$. Figure 4.17 illustrates different choices of K_i on some large components for the Oxford 5K and Paris 6K datasets. The results show that the buildings (*Christ church* and *Louvre*) are naturally grouped by different canonical views.

The partition of the matching graph repeats on each connected component C_i (if necessary), and the topics are obtained incrementally from each component:

[2]In our experiment, large components contain over 20 images



(a) 39 images, 3 sub-graphs



(b) 44 images, 2 sub-graphs

Figure 4.17: Partition the large components from the Oxford 5K and Paris 6K datasets. Number of images contained in these two large components is shown in the bottom of the pictures.

$\mathbf{K} = \{K_1, K_2, \dots\}$. After obtaining all topics, we investigate the distribution of each visual word in each topic as a vector: $\mathbf{u}_i := \{u_{ij}\}_{j=1}^{|\mathbf{K}|}$, where the entry u_{ij} is the normalized frequency of the visual word w_i in topic j. The semantic distance between word w_i and w_j can be computed as follows:

$$\Theta_3(w_i, w_j) = \| \mathbf{u}_i - \mathbf{u}_j \|_2 \tag{4.15}$$

Based on this distance measure, features, which belong to different images but the same semantic topic, can be considered to be semantically “close". Figure 4.18 shows the distribution of topics in a pair of visual words (w_i, w_j) which have close semantic distance.

As seen in Eq. (4.10), the visual word weights f_{it}^* is defined by the nearest neighbors, and depends on a selected distance measure. We illustrate the effects of various distance measure individually in Eq. (4.10): f_{mit}^*, m = [1,2,3] (the L2 word distance, the spatial co-occurrence distance and the semantic distance).

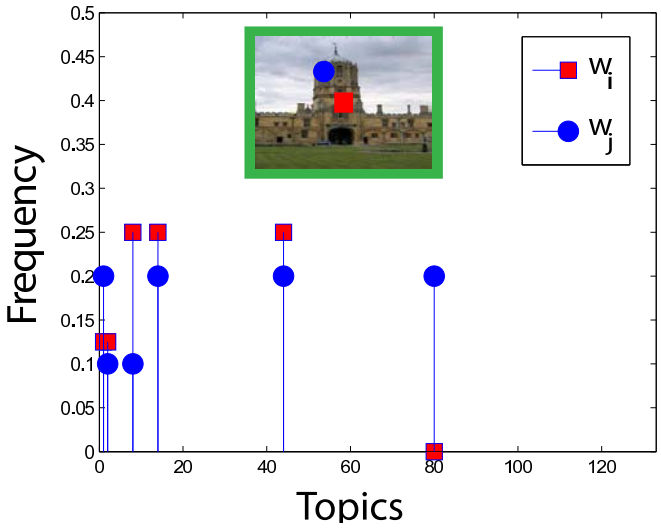


Figure 4.18: Illustration of a pair of visual words close in the semantic distance measure. Note that these two words have similarity topic distribution and thereby refer to the same object.

For each word $w_i \in \mathbf{W}_S$, we can find its T_m nearest neighbors $\{w_t^i\}_{t=1}^{T_m}$ based on the selected visual distance Θ_m. The cross-word similarity $\Gamma_m(q,d)$ (Eq. (4.11)) is computed by the corresponding f_{mit}^* and T_m nearest neighbors. Because the weights are computed offline, and only a subset of nearest neighbours within $\mathbf{V}'$ are used in the distance measure, it is computationally efficient at query time. This reduces the computation of cross-word matching while keeping the accuracy.

4.3.3 Experimental results

The experiments in this section are designed to investigate the effects of different distance measurements on cross-word matching for ranking images. Each visual distance measure proposed in Section 4.3.2 affects computation of the visual weights $\mathbf{f}^*$ (Eq.(4.10)) and therefore affects the cross-word matching Γ (Eq.(4.11)). We test each distance measure in isolation, as well as combinations of distance measures obtained by taking their geometric mean. In particular, we evaluate:

- $\Gamma_1(q,d)$, using Eq.(4.11) with measure Θ_1
- $\Gamma_2(q,d)$, using Eq.(4.11) with measure Θ_2
- $\Gamma_3(q,d)$, using Eq.(4.11) with measure Θ_3

- $\Gamma_4(q,d)$, as $\Gamma_1^{1/2} \cdot \Gamma_2^{1/2}$ (Θ_1 and Θ_2)
- $\Gamma_5(q,d)$, as $\Gamma_1^{1/3} \cdot \Gamma_2^{1/3} \cdot \Gamma_3^{1/3}$ (Θ_1, Θ_2 and Θ_3).

This enables us to incrementally add information into the visual distance model. Each visual distance is added to the standard dot product metric $\Psi(q,d)$ (Eq. (4.1)) to form the final image distance measure $\text{sim}(q,d)$:

$$\text{sim}_m(q,d) = \Psi(q,d) + \Gamma_m(q,d), m = 1...5 \tag{4.16}$$

The experiments are conducted on two groups of image datasets. The first group of datasets are mainly buildings: Oxford [4] and Paris [4], Rome [128], INRIA Holiday [62]. The second group of datasets contains less geometric information than the first one, as examples shown in Figure 2.6.

Comparison of individual visual distance measures The image ranking consists of two independent components: $\text{sim}_i = \Psi + \Gamma_i$, $i = [1...5]$, where Ψ is the baseline dot product similarity and Γ_i is a distance measure. Table 4.8 reports the detailed retrieval performance with these five types of visual distance on the Oxford 5K, Paris 6K and Rome datasets, respectively, while Figure 4.19 shows the performance gain of different visual distance measures (sim_1-sim_5). The individual visual distance (sim_1, sim_2 and sim_3) helps to improve the retrieval performance. For example, the usage of semantic distance (sim_3) alone leads to an increase in the accuracy of 8.3% on the Oxford dataset, 10.8% on the Paris dataset, and 15.6% on the Rome dataset, all compared to the baseline method. However, we obtain some significant increases in mAP scores, *e.g. All Souls*, as well as some decrease, *e.g. Cornmarket* and *Pitt River*. For example, sim_2 almost doubles the mAP score for *Bodleian*, but leads to more than 20% decrease in *Cornmarket* and *Pitt River*. This is because the spatial information contains useful information when the object is large in the image, but can be noisy when the object is small or the images contain background clutter. Therefore, it is unstable to use individual visual distance.

An overall illustration of retrieval performance is shown in Figure 4.20, where the below-diagonal and above-diagonal markers respectively represent performance improvement and degradation. From the results in Figure 4.20 (a)-(c), we obtain slight improvement of the individual distance measure compared to the baseline. Therefore, we investigate combining the visual distance measures to

Oxford 5K	mAP					
Ground truth	Baseline	sim_1	sim_2	sim_3	sim_4	sim_5
All Souls	0.544	0.725	0.665	0.751	0.764	**0.795**
Ashmolean	0.617	0.666	0.777	0.760	0.765	**0.778**
Balliol	0.563	0.559	0.481	0.467	**0.569**	0.550
Bodleian	0.456	0.467	**0.867**	0.580	0.636	0.648
Chris. Chur.	0.589	0.678	0.754	0.689	**0.777**	0.771
Cornmarket	**0.584**	0.576	0.398	0.528	0.552	0.566
Hertford	0.816	0.864	0.906	0.886	**0.918**	0.915
Keble	0.774	0.782	0.854	**0.862**	0.830	0.856
Magdalen	0.186	0.194	0.176	0.164	**0.201**	0.198
Pitt River	**0.995**	**0.995**	0.715	0.961	0.977	0.986
Radc. Camb.	0.609	0.715	0.661	0.643	**0.778**	0.768
Total	0.612	0.662	0.659	0.663	0.706	**0.712**

Paris 6K	mAP					
Ground truth	Baseline	sim_1	sim_2	sim_3	sim_4	sim_5
Defense	0.419	0.436	0.461	0.474	0.475	**0.508**
Eiffel	0.463	0.521	0.587	0.578	0.574	**0.596**
Invalides	0.643	0.719	0.777	**0.791**	0.766	0.786
Louvre	**0.380**	0.369	0.351	0.336	0.373	0.379
Moulinrouge	0.607	0.653	0.659	**0.702**	0.668	0.687
Museedorsay	0.546	0.546	**0.632**	0.646	0.590	0.622
Notredame	0.807	0.847	**0.915**	0.866	0.897	0.909
Pantheon	0.920	0.967	**0.991**	0.972	0.984	0.987
Pompidou	0.916	0.909	**0.924**	0.914	0.919	**0.924**
Sacrecoeur	0.826	0.910	0.946	0.945	0.937	**0.953**
Triomphe	0.507	**0.586**	0.560	0.569	0.584	0.585
Total	0.639	0.679	0.709	0.709	0.706	**0.722**

Rome	mAP					
Ground truth	Baseline	sim_1	sim_2	sim_3	sim_4	sim_5
Arch	0.720	0.714	**0.864**	0.829	0.848	0.851
Castelsantangelo	0.328	0.362	**0.565**	0.463	0.513	0.526
Colosseum	0.533	0.510	0.658	**0.673**	0.612	0.652
Dome	0.915	0.908	0.963	**0.978**	0.944	0.973
Palazzosenatorio	0.932	0.898	**0.967**	0.946	0.924	0.934
Pantheon	0.674	0.871	0.915	0.824	**0.938**	0.933
Pope	0.649	0.686	0.763	0.758	0.757	**0.776**
Spiral	0.914	0.956	0.978	**1.00**	0.956	**1.00**
Trevifountain	0.695	0.730	0.787	0.794	0.804	**0.815**
Vittoriano	0.743	0.878	0.858	0.863	**0.881**	0.880
Total	0.680	0.726	0.807	0.786	0.793	**0.810**

Table 4.8: Retrieval performance evaluation for the Oxford 5K, Paris 6K and Rome building landmarks, with five types of visual distance $sim_1, sim_2, \cdots, sim_5$.

improve robustness.

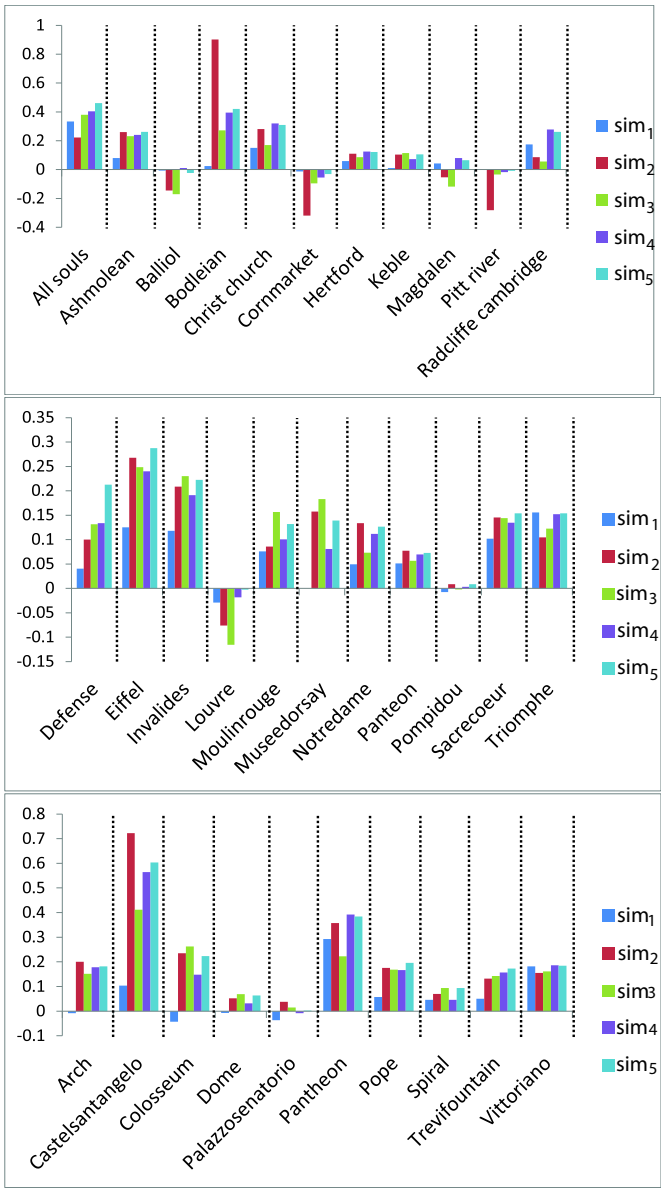


Figure 4.19: Illustration of increase (decrease) amount in mAP scores on the Oxford 5K, Paris 6K and Rome dataset. The vertical axis is the percentage of mAP changes compared between the baseline method and each $\text{sim}_1, \text{sim}_2, \cdots, \text{sim}_5$. The horizontal axis is the results grouped by different landmarks in each dataset.

Comparison of inter-word distance measures We consider multiple attributes of features when calculating image similarity in a cross-word manner. Results in Figures 4.19 and 4.20 show that the fused distances (Γ_4, Γ_5), are more robust than the individuals ones ($\Gamma_1, \Gamma_2, \Gamma_3$). It can bee seen in Figure 4.20 that almost all the markers of sim_4 (the combination of the first two visual components) are above

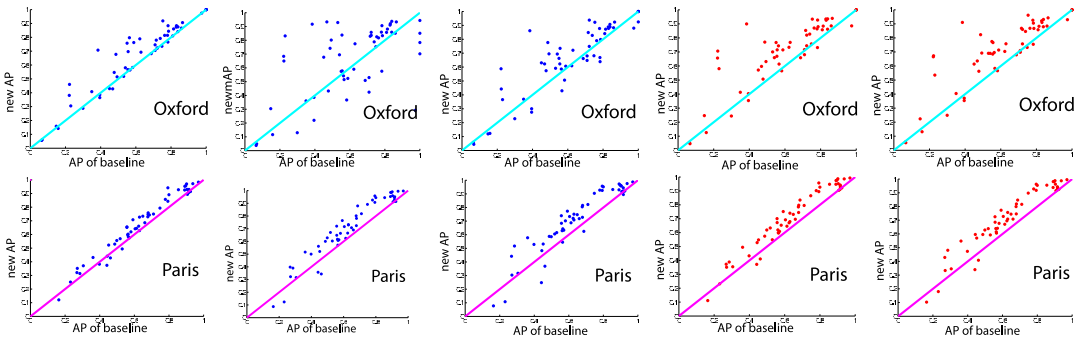


Figure 4.20: Comparison of mAP scores on all the 55 queries on the Oxford 5K and Paris 6K datasets, baseline versus our cross-word matching method. $(a) - (c)$: the individual visual distance components (sim_1, sim_2, sim_3). (d) combination of the first two components (sim_4). (e) combination of all the components (sim_5).

the diagonal; sim_5 (the combination of all visual components) further improve the results and has the highest overall mAP. As the results show in Table 4.8, image ranking using sim_5 attains the best overall retrieval results on all the three datasets. Compared to the baseline method, the mAP score under sim_5 increased 16.3% on the Oxford dataset, 13.0% on the Paris dataset, and 19.1% on the Rome dataset, respectively. As a result, the use of visual distances based on multiple attributes encodes more information, and hence reduces the disadvantage of individual attribute distance to get the best results on all the datasets. Combining all components of visual distance, sim_5, attains the best overall results on all the five datasets.

Specifically, Figures 4.21 and 4.22 show some selected precision-recall curves in two groups: the baseline method and our method sim_5. In these query examples, almost all the precision-recall curves of our method move to the right corner of the graph. Figure 4.20 (e) illustrates the mAP scores of all the 55 queries on the Oxford 5K and Paris 6K datasets, before and after using the cross-word matching (sim_5). Among the 55 query results (mAP), there are 44 query results of our method are better than the baseline method on the Oxford 5K dataset, while 51 query results on the Paris 6K dataset.

Moreover, it is observed that the dot product similarity Ψ degrades quickly. As a result the highly ranked false positives are difficult to distinguish from the true positive in the top ranking results. By including the cross-word matching Γ,

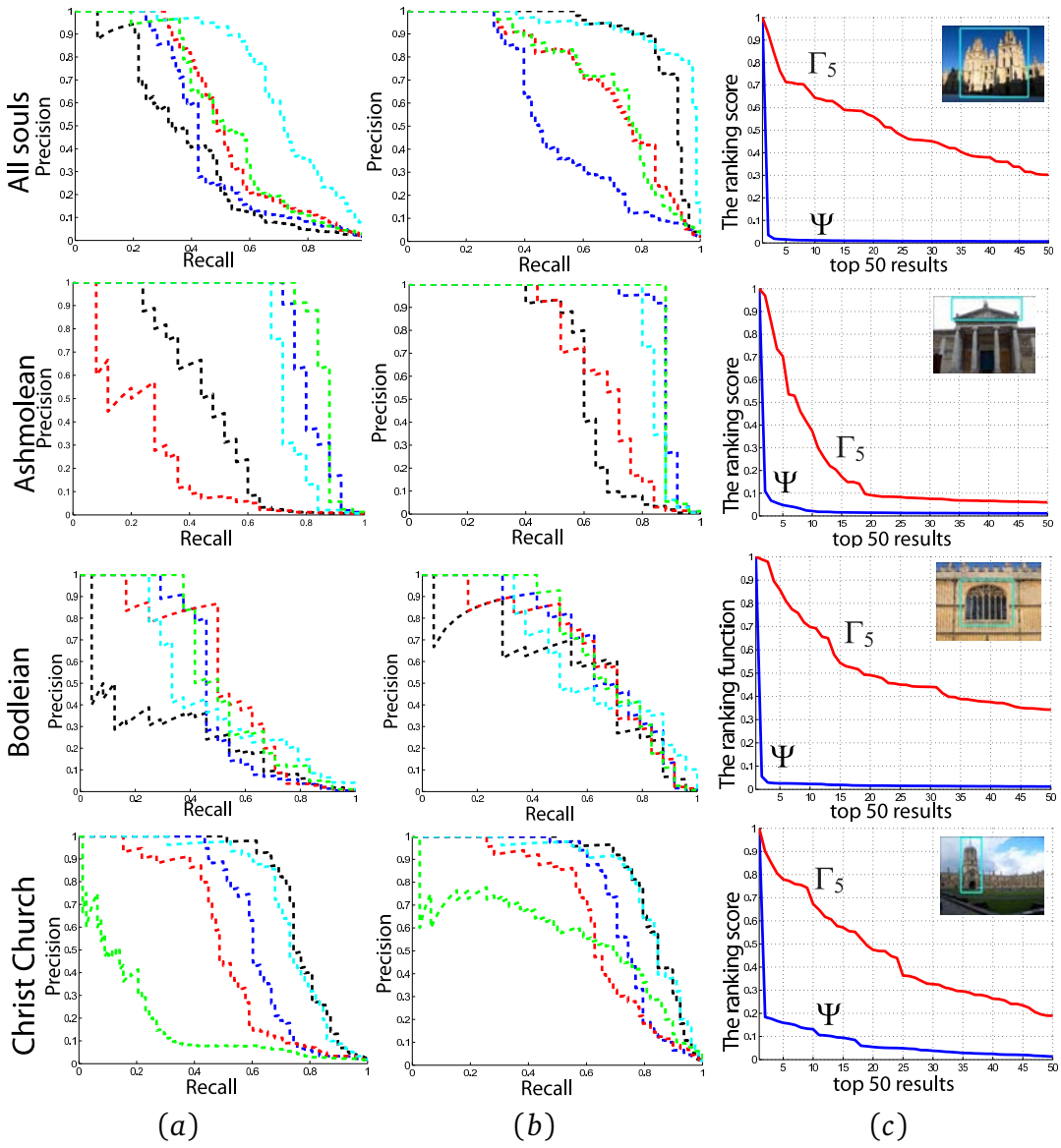


Figure 4.21: Detail of retrieval results on queries of the Oxford 5K dataset. (a): Precision-recall (PR) curves of dot product (baseline) on 5 queries of each landmark. (b): Precision-recall (PR) curves using cross-word matching on 5 queries of each landmark, with combination of visual distance sim_5. The color in (a) and (b) indicates 5 individual queries of each landmark. (c) Scores of dot product image ranking score (blue) and the cross-word image ranking score (red) on one of the query image in (a) and (b). Both of them have been scaled into the same range.

the true positives are more accurately ranked in the ranking lists, as shown in Figures 4.21 and 4.22 (c). Based on these results, we use sim_5 as the cross-word image ranking method in our following experiments. Table 4.9 gives the retrieval performance results of our method on all available datasets. Overall, our method

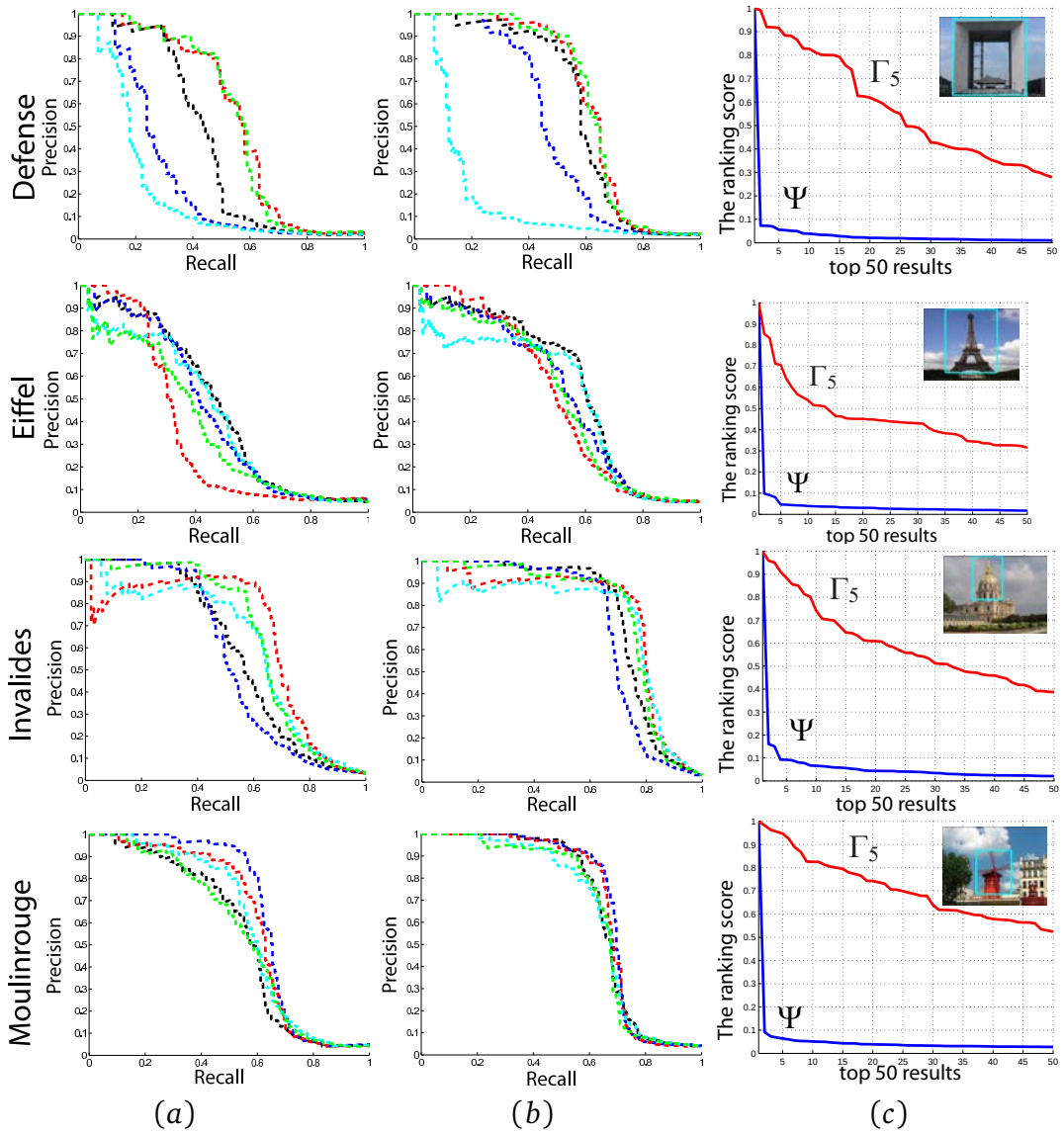


Figure 4.22: Detail of retrieval results on queries of the Paris 6K dataset. Figures in columns (a)-(c) are illustrated as in Figure 4.21.

leads to more than 10% improvement on the mAP scores on all the datasets. This shows that it can work on a number different kinds of datasets. Among them, the Rome dataset had the greatest increase in accuracy. In the second group, the Caltech Categories obtains about 22% improvement in retrieval performance. It improves mAP even under very challenging conditions, such as the ImageNet datasets. The Oxford 5K + 1M dataset illustrates that our method can perform on large scale dataset as well, due to the fact that we explore the visual distance on a subset of images in the dataset. The retrieval accuracy decreases consistently

Dataset	Baseline	Our method
Oxford 5K	0.612	0.712
Paris 6K	0.639	0.722
Rome	0.680	0.810
Holiday	0.548	0.589
Caltech Categories	0.379	0.464
ImageNet (animals)	0.239	0.243
Oxford 5K+ 100K	0.515	0.601
Oxford 5K + 200K	0.499	0.592
Oxford 5K + 500K	0.472	0.566
Oxford 5K + 1M	0.449	0.541

Table 4.9: Results of the retrieval performance on the five datasets. For each dataset, we approximately obtain 10% visual words from $\mathbf{W}$ for the cross-word matching $\mathbf{W}_S$ during training stage.

as the dataset size enlarges, because the Oxford 5K + 1M includes more FLICKR images to search.

Alternate cross-word match measures In the previous experiments, the dot product and cross-word distance measures were combined by addition, after normalising each measure. This reflects our belief that they are measuring largely independent properties of the image. Meanwhile, the multiple cross-word measures in sim_4 and sim_5 are combined using the geometric mean, to compensate for the different ranges they may attain (they are normalised after combination).

We now test these steps empirically, by comparing results obtained previously using sim_5 to those obtained by combining Ψ and Γ by multiplication, and by taking the arithmetic mean of the cross-word measures. Table 4.10 shows that we consistently achieve the highest mAP result across multiple datasets when Γ computed as the geometric mean and the ranking function as the sum of Ψ and Γ, and thus we use this fusion method for the rest of the thesis.

4.3.4 Discussion

The top retrieval results by the cross-word matching are illustrated in Figure 4.23. We compare our cross-word matching method to a number of state-of-the-art methods in Table 4.11.

Oxford 5K	$\Psi + \Gamma$	$\Psi * \Gamma$
$\prod_{m=1}^{L} \Gamma_m^{1/L}$	**0.712**	0.613
$1/L \sum_{m=1}^{L} \Gamma_m$	0.666	0.620

Paris 6K	$\Psi + \Gamma$	$\Psi * \Gamma$
$\prod_{m=1}^{L} \Gamma_m^{1/L}$	**0.722**	0.641
$1/L \sum_{m=1}^{L} \Gamma_m$	0.719	0.647

Caltech Categories	$\Psi + \Gamma$	$\Psi * \Gamma$
$\prod_{m=1}^{L} \Gamma_m^{1/L}$	**0.464**	0.379
$1/L \sum_{m=1}^{L} \Gamma_m$	0.430	0.381

Table 4.10: Results of different distance measure fusion on the Oxford 5K, Paris 5K and Caltech dataset. We compare two kinds of fused distance ($\prod_{m=1}^{L} \Gamma_m^{1/L}(q,d)$ and $\sum_{m=1}^{L} \Gamma_m(q,d)$) together with two ways of using distance functions Ψ and Γ. Note that Ψ and Γ are scaled into the same range.

In Group A, the cross-word matching method can outperform most current state-of-the-art methods, except methods using dataset-side feature augmentation, *e.g.* AUG [142], and SPAUG [9]. Compared to these methods, our cross-word matching is conducted on a small subset of the vocabulary and does not require re-computing the features. We also notice that our method can get similar accuracy to the soft-assignment method, when the cross-word matching relies on the $L2$ visual distance, as expected. Compared with Table 4.8, the retrieval results (mAP) of Θ_1 on the Paris dataset are better than the soft-assignment. However, the result on the Oxford dataset is close to the results reported in [107], but still has a gap in the accuracy. The accuracy depends on the assignment of features to multiple visual words. Our method only uses about 10% of the visual words selected unsupervised from the vocabulary to calculate the cross-word similarity. As a result, the online improvement of similarity measure is efficient.

Group B and Group C compares to methods that use a post-process. Our method can outperform the spatial verification [106] without a query time spatial consistency examination. However, it is lower than various query expansion methods as shown in Group B. In Group C, we jointly use our method with standard post-processing methods, *e.g.* spatial verification and various query expansion methods. For the spatial re-ranking, our method can achieve further improvement on the retrieval performance For the query expansion methods, our

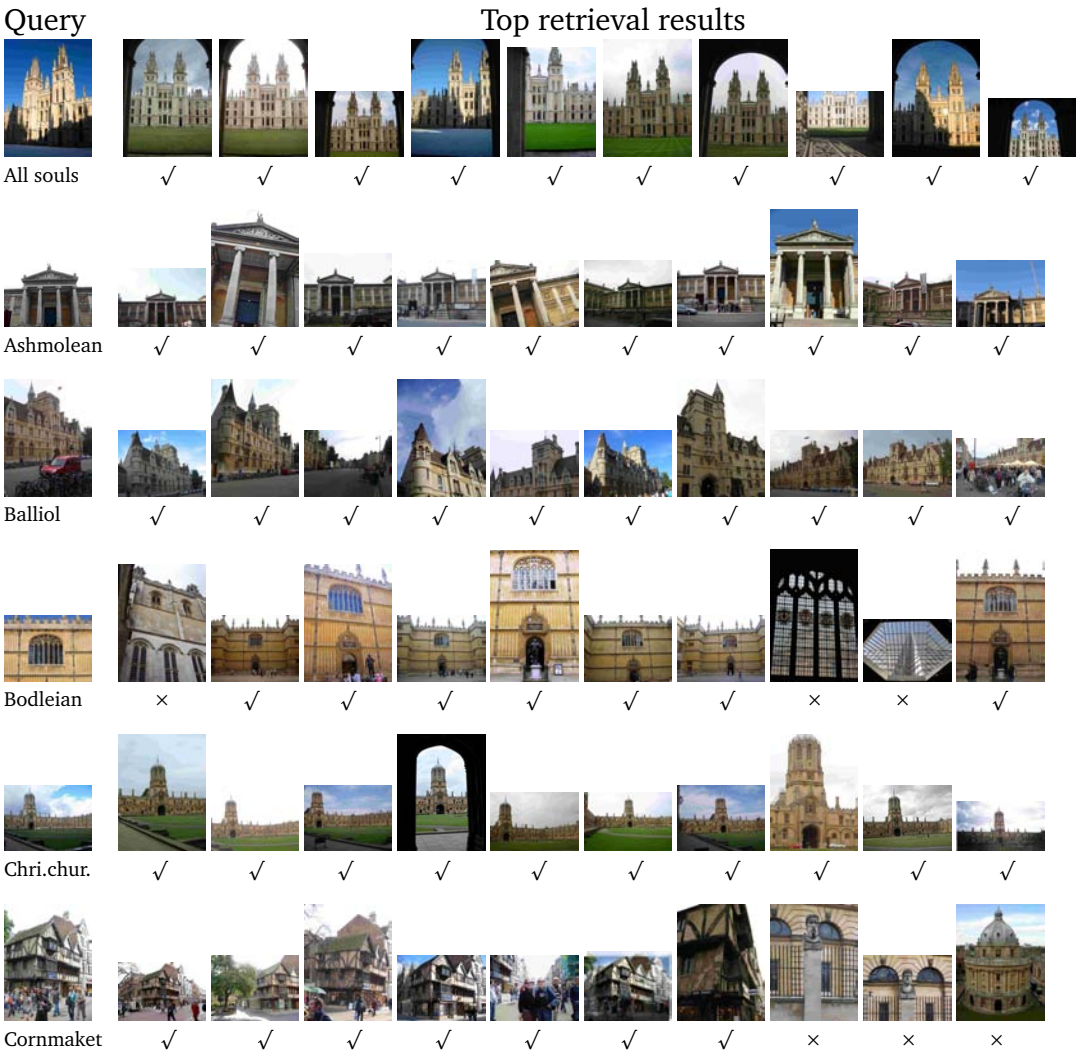


Figure 4.23: Top retrieval results of cross-word matching scheme.

method can be used in the shortlist generation. To improve AQE and DQE, our cross-word matching provides more accurate target images. This leads to further improvement on these post-processing methods. However, this improves the retrieval accuracy slightly, similar to the results reported in Table 4.7. This is due to the spatial information has been repeatedly used in cross-word similarity and the query expansion methods (AQE or DQE). Compared to previous methods, *e.g.* total association, the result when combined with post-processing mostly relies on the correct retrieval images in the top places. Therefore, query expansion methods dominate the performance gain in these combined work.

	Methods	Oxford 5K	Paris 6K	Oxford105K
	Baseline [106]	0.612	0.639	0.515
A	Visual word re-weighting (Section 4.1.1)	0.660	0.674	0.598
	Descriptor learning (non-linear) [108]	0.662 [108]	0.678 [108]	0.541 [108]
	Soft-assignment [107]	0.673 [107]	0.660 [107]	N/A
	Spatial expansion ($\mathbf{F}_5$,Chapter 3)	0.685	0.679	0.622
	Geometry-Preserving [159]	0.696 [159]	N/A	0.604 [159]
	Total association (Section 4.2)	0.700	0.682	0.680
	Spatial expansion ($\mathbf{F}'_{15}$, Chapter 3)	0.701	0.683	0.667
	Cross word	0.712	0.722	0.604
	Fine vocabulary [96]	0.742 [96]	0.749 [96]	N/A
	AUG [142]	0.776 [9]	N/A	0.711 [9]
	SPAUG [9]	0.785 [9]	N/A	0.723 [9]
B	Spatial verification [106]	0.645	0.655	0.571
	QE Baseline [37]	0.708	0.736	0.679
	iSP [34]	0.741 [34]	0.769 [34]	0.649 [34]
	Local geometry [105]	0.788 [105]	0.634 [105]	0.725 [105]
	AQE [37]	0.806	0.769	0.767
	DQE [9]	0.798	0.783	0.809
	Hello neighbors [114]	0.814 [114]	0.803 [114]	0.767 [114]
	Total recall II [34]	0.827 [34]	0.805 [34]	0.767 [34]
C	Spatial expansion ($\mathbf{F}'_{15}$, Chapter 3)	0.701	0.683	0.667
	Spatial expansion+ Spatial verification	0.719	0.689	0.704
	Spatial expansion+ AQE	0.806	0.785	0.783
	Spatial expansion+ DQE	0.813	0.789	0.818
	Visual word re-weighting	0.660	0.674	0.598
	Visual word re-weighting+ Spatial verification	0.677	0.684	0.611
	Visual word re-weighting+ AQE	0.801	0.777	0.781
	Visual word re-weighting+ DQE	0.811	0.782	0.787
	Total association	0.700	0.682	0.680
	Total association+ Spatial verification	0.710	0.690	0.706
	Total association+ AQE	0.804	0.785	0.774
	Total association+ DQE	0.816	0.790	0.817
	Cross word	0.712	0.722	0.604
	Cross word + Spatial verification	0.723	0.726	0.647
	Cross word + AQE	0.821	0.787	0.765
	Cross word + DQE	0.828	0.793	0.797
	Contextual synonym dictionary + AQE [138]	0.811 [138]	0.791 [138]	0.797 [138]

Table 4.11: Retrieval performance comparison with our cross-word distance measure method. Group A: retrieval results of methods that modify the baseline before the query is executed (pre-process). Group B: retrieval results of methods that modify the baseline after the query is executed (post-process). Group C: comparison of methods jointly working with spatial verification and various query expansion methods. Note that we cite the retrieval results of AUG [142] from literature [9].

4.4 Conclusion

In this chapter, we have demonstrated three kinds of enhanced visual similarity measure and show their effects on object retrieval. All of the measures aim to overcome the quantisation errors that can not be addressed by the standard dot product similarity measure. The first method is a re-weighting scheme. It adjust the importance of visual words that are associated with foreground objects. The second method considers both recall and precision boosting, by combing the spatial expansion and visual word re-weighting. The third method is a cross-

word matching scheme. It modifies the distance metric by combing various visual distance measures. All of the methods are built on an object-based thesaurus, which captures the foreground visual words. The experimental results show that our methods outperform most the state-of-the-art methods, and can be combined with other techniques for retrieval performance improvement.

The methods proposed in this chapter, as well as spatial expansion proposed in Chapter 3 rely on exploiting of foreground information embedded in dataset images. Therefore, these methods have limitation when foreground information is insufficient or hard to discover. As a result, these methods are suitable for rigid object retrieval. In order to improve the retrieval results regardless of foreground information, we study result re-ranking methods from next chapters.

Context based re-ranking for object retrieval

Chapter V

In previous Chapters 3 and 4, we have proposed several methods for improving retrieval performance by modification of the BoW representation and similarity measure. We have shown that these methods can jointly work with a number of post-processing methods, *e.g.* spatial verification and various query expansion methods. Note that these methods need to exploit foreground information in a learning stage before online query. As a result, they work well on datasets containing rigid objects, but lack effectiveness when the foreground information can not be extracted from the dataset.

In this chapter, we pay attention to refining the initial ranking scores according to contextual information embedded in the retrieval results. As discussed in Chapter 2, the fundamental problem of a retrieval system is to rank images according to the similarity of their visual content to the query. In the standard BoW retrieval system, dataset images are ranked according to their dot product similarity to the query vector. Thus, each image's rank is independent, *i.e.* only based on a one-to-one comparison with the query, and therefore ignores information from other images in the dataset.

In contrast to the one-to-one comparison used in dot product similarity, the ranking function in a retrieval system should also consider contextual information from other dataset images. As illustrated in Figure 5.1, dataset images containing the same object or scene are grouped together. We define "contexts" as collections of images having common visual properties (objects). In this chapter, we introduce contextual information to the refinement of object retrieval results. For object retrieval, the ranking score of each image is not only determined by their individual similarity to the query, but also influenced by the ranks of other images belonging to the same context. As illustrated in Figure 5.1, images in a context should be encouraged in ranking scores if they support each other, *i.e.* have similar ranks,

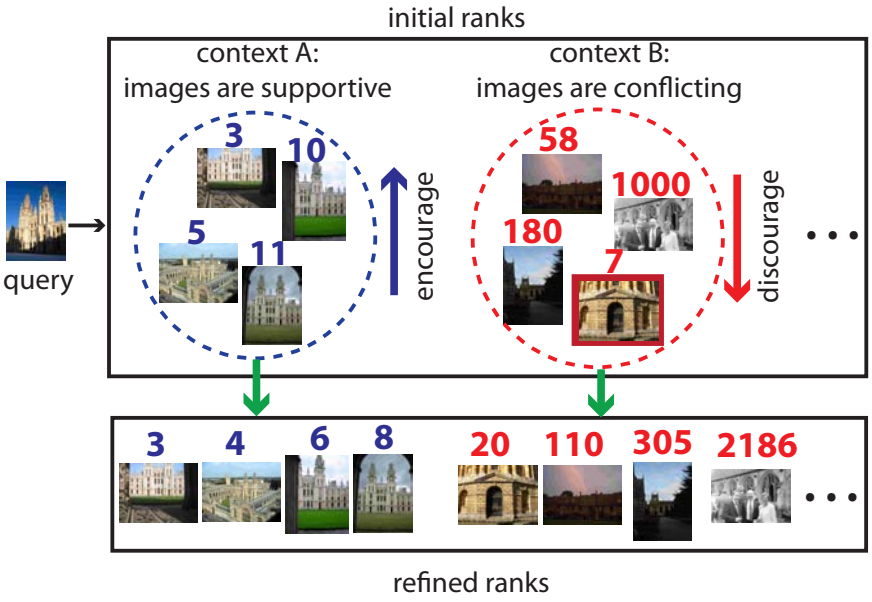


Figure 5.1: Illustration of context based re-ranking. The initial retrieval results (sorted by similarity Ψ) contain both true and false positives. However, visually similar images are more likely to group together (context A) than those are not (context B), and thus should be moved towards the query. The numbers above images are rank orders, before or after context based re-ranking. After context based re-ranking, images in context A (context B) are ranked highly (lowly).

and ranked highly (context A). In contrast, images are discouraged if they have conflicting ranks in a context (context B). Therefore in this chapter, we aim to improve the retrieval results by contextual information obtained online and show its usage in efficiently re-ranking the initial results.

Typically, contexts are obtained by clustering the image dataset. Traditional methods, *e.g.* k-means, are unable to cluster the BoW vectors efficiently and effectively, because the BoW vectors are very sparse and high dimensional. We utilize these properties of the BoW vectors to simplify the clustering: the sparsity results in effectiveness of partition along randomly chosen dimensions; the high dimensionality leads to efficiency because only a small number of dimensions are needed. Based on these observations, we propose a *random space partition* method, by which the dataset is clustered into groups of images after repeated partitions by random dimensions. Once the image groups are available, the initial dot product scores can be refined by analysis of the rankings in each image group. Each image group generates a context score according to the association of image

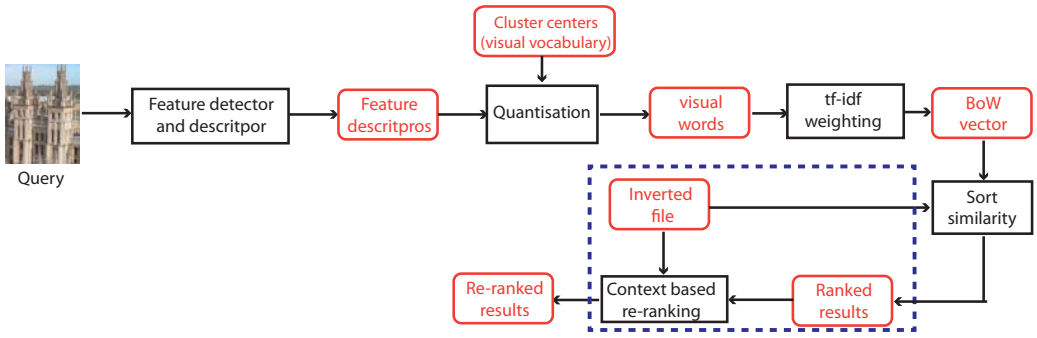


Figure 5.2: System framework of context based re-ranking. This is the standard BoW retrieval system with new steps (indicated in dash box) introduced by our context based re-ranking method. The offline process is unchanged.

ranks in the context, and then each dataset image is re-ranked by the context scores of the groups it belongs to. In our method, contexts are created and analysed at query time, and have minimal computational and storage cost. The place of re-ranking module in the pipeline is illustrated in Figure 5.2.

Query expansion also focuses on refining the similarity measure or re-ranking an initial set of search results, as discussed in Section 2.4. Alternatively, reciprocal similarity [114] can be used to partition search results into near and far sets. The reciprocal similarity is discovered by the k-reciprocal nearest neighbor structure built offline. High level semantic information is helpful in improving the similarity measure, but it requires expensive learning stage to exploit latent relationship between dataset images, as discussed in Section 2.4.5. In contrast, our method softly adjusts the similarity scores at run time by the contextual information. It needs no prior knowledge about the dataset images.

5.1 Context based re-ranking

The standard BoW retrieval system ranks images based on sorting the dot product similarity [130] between the tf-idf vectors $\mathbf{q}$ and $\mathbf{d}$, corresponding to query image q and each dataset image d (Eq. (4.1)). Each dataset image d then obtains a rank order r_d under $\Psi(q,d)$, for which top ranks are probably relevant to query while bottom ranks are effectively random. The ranking is efficient, but neglects contextual information linking the returned results as it only measures similarity

Algorithm 9 Context based re-ranking.

1: **Input:** Query image q, number of random dimensions D.
2: **Output:** Retrieval results.

3: Rank dataset images by sorting dot product similarity Ψ (Eq. (4.1)) and obtain initial ranks of dataset images
4: Select D dimensions (Eq. (5.3)).
5: Generate image groups $\mathscr{C} := \{\mathbf{c}_k\}_{k=1}^{D}$ from inverted file (Eq. (5.2)).
6: Compute the context score $W(q, \mathbf{c}_k)$ for each image group (Eq. (5.5)).
7: Compute context factor $\Omega(q, d)$ for each dataset image d (Eq. (5.6)).
8: Adjust image similarity and re-rank (Eq. (5.1)).
9: **Return:** Re-ranked results.

between the query and each dataset image in isolation.

In order to discover this contextual information, the dataset is clustered into small groups. We propose a **random space partition** method, which is a simplified clustering method for the high dimensional data, to approximately separate the dataset (Section 5.1.1). These contexts are scored based on the ranks of result images belonging to them (Section 5.1.2). The contextual ranking information is used to adjust the dot product similarity Ψ online:

$$\Phi(q, d) = \Psi(q, d) \cdot \exp(\Omega(q, d)) \tag{5.1}$$

where $\Phi(q, d)$ is the refined image similarity. The context factor $\Omega(q, d)$ in Eq. (5.1) indicates positive or negative context score learnt from image groups. Our method is outlined in Algorithm 9 and described further below.

5.1.1 Random space partition

As before, a visual vocabulary is composed of N visual words: $\mathbf{W} := \{w_i\}_{i=1}^{N}$, where $N = 10^6$ in our implementation. An image dataset can be represented as a collection of visual word vectors: $\mathscr{S} = \{\mathbf{d}_j\}_{j=1}^{V}$, in which V is number of images and $\mathbf{d}_j$ is the corresponding tf-idf image vector. The goal of our method is to cluster $\mathscr{S}$ into a set of groups: $\mathscr{C} := \{\mathbf{c}_k\}_{k=1}^{D}$, where each group $\mathbf{c}_k$ is a context that contains a small number (n_k) of dataset images.

The clustering of $\mathscr{S}$ involves two issues: i) scalability: the clustering is conducted on high dimensional vectors $\mathscr{S}$, for which standard k-means methods or graph cut of the image dataset [110] are not feasible. ii) efficiency: as it runs at

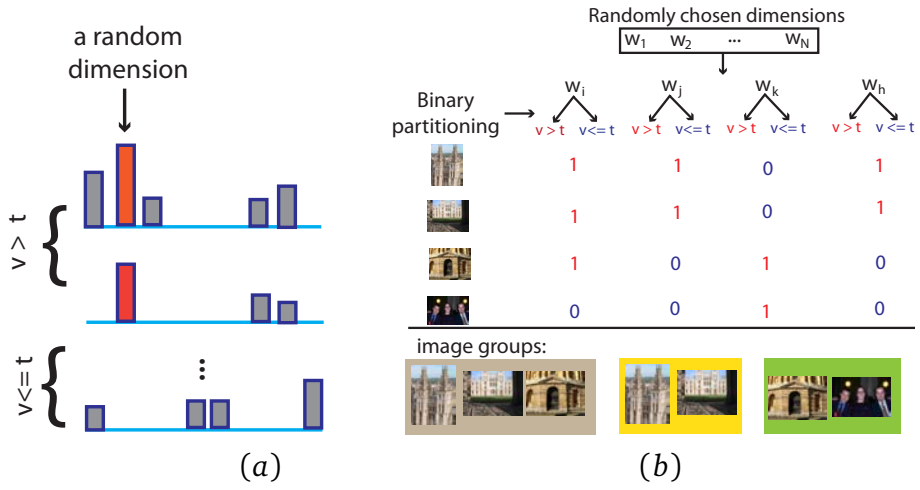


Figure 5.3: (a): Image vector separation by partition of a random dimension. (b): Illustration of random space partition method by using an inverted file.

query time, the partition should have low computation and memory requirements. In order to address these issues, we present a **random space partition** method, described as follows.

Firstly, the image vectors $\mathscr{S}$ are very sparse. For example, there are on average 2500 non-zero entries in the 10^6 dimensional vector of the Oxford 5K dataset. The high sparsity simplifies the partitioning of $\mathscr{S}$. As illustrated in Figure 5.3 (a), $\mathscr{S}$ is separated into two groups by a random dimension of the image vectors, according to whether each vector exceeds a threshold t in this dimension. Note that each dimension i of image vectors corresponds to one visual word w_i, so the images can be quickly accessed by an inverted file, which maps each visual word to images it appears in. Thus, each "column" of the file (as it is shown in Figure 5.3 (b)) corresponds to a visual word w_i and forms an image group $\mathbf{c}_k$:

$$\mathbf{c}_k = \{\mathbf{d}_j\}_{j=1}^{n_k} \quad \text{if } v_j(w_i) > t \tag{5.2}$$

where $v_j(w_i)$ is the number of occurrences of w_i in image j (tf) and t is a threshold. Scalable clustering of $\mathscr{S}$ is achieved by repeated random partitions. As the inverted file is already used for the calculation of tf-idf weights, this involves almost no extra computation or storage beyond the standard BoW pipeline.

Secondly, the efficiency of our method is achieved by performing only D ($D \ll N$) data partitions to generate groups $\mathscr{C}$. According to Eq. (5.2), a dataset

image d might appear in a context or not. Thus, we index each dataset image d by a set of D indicators $\{\mathbb{I}_k^d\}_{k=1}^D$, where $\mathbb{I}_k^d = \{0,1\}$ indicates whether d appears in $\mathbf{c}_k$ or not. Obviously, it is inefficient to use all dimensions (visual words) because only a small number of them are informative, as discussed in Chapter 3. These are usually the query words $\mathbf{Q}$ and their relevant words $\mathbf{S}$ ($\mathbf{Q}, \mathbf{S} \subset \mathbf{W}$). The query-relevant words $\mathbf{S}$ can be discovered offline or online by methods presented in previous chapters, and will be discussed in the next paragraph. We randomly choose a subset of dimensions (visual words) in which $\mathbf{Q}$ and $\mathbf{S}$ are given a higher probability of selection than those that are not relevant (not in $\mathbf{Q}$ and $\mathbf{S}$). This is done by associating visual words to random hash keys under f: $[f(w_1), f(w_2), \cdots, f(w_N)]$:

$$f(w_i) = \begin{cases} a \cdot x & \text{if } w_i \in \mathbf{Q} \\ b \cdot x & \text{if } w_i \in \mathbf{S} \backslash (\mathbf{Q} \cap \mathbf{S}) \\ x & \text{otherwise} \end{cases} \tag{5.3}$$

where $w_i \in \mathbf{W}$, x is a random variable from uniform distribution $U(0,1)$ and the parameters a, b are the weights to give priority to query words $\mathbf{Q}$ and their relevant words $\mathbf{S}$ over others. The D dimensions used for partition are then selected in decreasing order of $f(w_i)$. The scheme of random dimension selection is similar to [38]. Note that both of the parameters a, b are equal or greater than 1.

We define three cases based on values of a, b: **i) Random selection**: $a = 1$, $b = 1$: each visual word has uniform probability of being selected. **ii) Query-dependent selection**: $a > 1$, $b = 1$: words in the given query $\mathbf{Q}$ are more likely to be selected. **iii) Query-expansion selection**: $a > 1$, $b > 1$: words in the query $\mathbf{Q}$ and the query-relevant set $\mathbf{S}$ are more likely to be selected than others.

Query-relevant words $\mathbf{S}$ can be generated as follows: i) *offline*: build a general thesaurus structure offline and obtain $\mathbf{S}$ via spatial expansion online (Chapter 3). In this case, these are spatially related words of the original query used in spatial expansion ($\mathbf{W}_T$). ii) *online*: obtain $\mathbf{S}$ by query expansion [37]. These words are also spatially related to the original query, but examined by a spatial consistency test online, which has been used for average query expansion (AQE). After obtaining D dimensions (visual words), image groups $\mathscr{C} := \{\mathbf{c}_k\}_{k=1}^D$ are used to estimate the context score for re-ranking, as discussed below.

5.1.2 Context factor for re-ranking

Our context based re-ranking method proceeds in two steps. Firstly, our context based re-ranking method aims to learn a query-specific context score $W(q, \mathbf{c}_k)$ for each image group $\mathbf{c}_k$, according to Eq. (5.5). Secondly, a dataset image d will be assigned a context factor $\Omega(q, d)$, which is learnt from n_d image groups it has been mapped to (Eq. (5.6)). In this scheme, the first step aims to measure how and whether images in a context are close to the query, *i.e.* in the top ranked results. Thus, the context score is a signed real value to indicate this property. The second step utilizes these context scores to design a context factor for each dataset image d. As a result, it makes the retrieval system consider not only one-to-one comparison of image similarity, but also contextual influence of groups related to each result images. Thus the dataset image d is re-ranked by the similarity score refined by the context factor (Eq. (5.1)). Details of these two steps are described below.

Compute context score Each image group is assigned a context score $W(q, \mathbf{c}_k)$, which indicates whether and how the image ranks in this group are close to the top/bottom in the ranked result of query q. This is measured by two factors:

i) The association of image ranks in $\mathbf{c}_k$:

$$\mathbf{c}_k = \frac{1}{n_k^2} \sum_{j=1}^{n_k} \sum_{s=1}^{n_k} K(\frac{r_j - r_s}{\rho}) \tag{5.4}$$

where r_j and r_s are the image ranks in $\mathbf{c}_k$, n_k is the group size, K is a Gaussian kernel and ρ is its bandwidth. In this way, the association of a context $\mathbf{c}_k$ is measured by its corresponding association of image ranks in $\mathbf{c}_k$. Note that the parameter ρ can be automatically tuned. Eq. 5.4 can be written as: $\frac{1}{n_k} \sum_{j=1}^{n_k} \frac{1}{n_k} \sum_{s=1}^{n_k} K(\frac{r_j - r_s}{\rho})$, in which $\frac{1}{n_k} \sum_{s=1}^{n_k} K(\frac{r_j - r_s}{\rho})$ is the kernel density estimator of a context image j, measured by its rank samples r_s in the ranking list. Thus, parameter ρ can be tuned based on estimating the standard deviation of the input image ranks of the context [17]. As a result, image groups, which are distributed widely in the ranking list will have less association and will not be weighted strongly.

ii) The number of top (bottom) image ranks: $\frac{t_q(\mathbf{c}_k, H) - b_q(\mathbf{c}_k, H)}{n_k}$, where functions

$t_q(\mathbf{c}_k, H)$ and $b_q(\mathbf{c}_k, H)$ count the number of members $\mathbf{c}_k$ in the top and bottom-H places, respectively. This indicates whether the contexts are close to query q or not.

The context score of group $\mathbf{c}_k$ is estimated by both of these two factors:

$$W(q, \mathbf{c}_k) = \left[\frac{1}{n_k^2} \sum_{j=1}^{n_k} \sum_{s=1}^{n_k} K\left(\frac{r_j - r_s}{\rho}\right) \right] \cdot \frac{t_q(\mathbf{c}_k, H) - b_q(\mathbf{c}_k, H)}{n_k} \tag{5.5}$$

Re-ranking based on context score The re-ranking utilizes these context scores to improve the similarity score of a dataset image d. As each image is assigned to multiple contexts, the context factor is defined as the average score for all contexts it belongs to:

$$\Omega(q, d) = \frac{1}{\sum_{k=1}^{D} \mathbb{I}_k^d} \cdot \sum_{k=1}^{D} \mathbb{I}_k^d W(q, \mathbf{c}_k) \tag{5.6}$$

Images are re-ranked by sorting $\Phi(q, d) = \Psi(q, d) \cdot \exp(\Omega(q, d))$ (Eq. (5.1)). As a result, the initial similarity of images having negative context factors ($\Omega(q, d) < 0$) will be decreased, while those having positive context factors ($\Omega(q, d) > 0$) will be increased.

5.2 Experimental results

The retrieval experiments are conducted on three public object retrieval datasets: two small-scale datasets: Oxford 5K and Paris 6K; and a large scale dataset: Oxford 105K.

5.2.1 Parameter setting

Firstly, we evaluate the effects of various parameter settings on our method.

Randomly chosen dimension number D Figure 5.4 (a) reports the retrieval accuracy for increasing D dimensions selected to generate random partition. As illustrated in Figure 5.4 (a), the accuracy improves as D increases, and then plateaus above a threshold, *e.g.* $D = 7 \times 10^4$ on both Oxford 5K and Paris 6K dataset. This illustrates that the retrieval performance of our method becomes

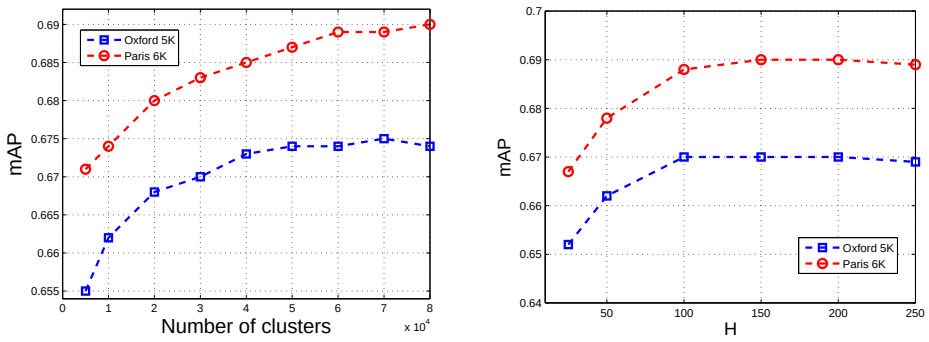


Figure 5.4: (a): Retrieval results comparison with increasing dimensions D. (b): Retrieval results comparison with increasing top/bottom-H.

Methods		Oxford 5K	Paris 6K
Baseline		0.612	0.639
Spatial verification		0.645	0.653
$a = 1, b = 1$	f_1	0.644	0.674
$a > 1, b = 1$	f_2	0.670	0.690
	f_3	0.674	0.690
$a > 1, b > 1$	f_4	0.676	0.691
	f_5	0.684	0.697
	f_6	**0.701**	**0.700**
	f_7	0.692	**0.700**

Table 5.1: Retrieval performance with different weighting functions used in ordering the visual words. Total number of visual words selected: 3×10^4.

stable when there are enough contexts used to calculate the context factor Ω. Moreover, the number of visual words (dimensions) needed to achieve the stability is far less then the vocabulary size ($D \ll N$). Table 5.4 shows the average CPU time as D increases. Intuitively, the re-ranking needs more CPU times as D increases. By considering both accuracy and run time, we set $D = 3 \times 10^4$.

Weighting parameters a, b Table 5.1 illustrates the effects of weighting parameters a, b (Eq. (5.3)) on the retrieval results. Firstly, parameter b is fixed ($b = 1$) such that Table 5.1 investigates the effect of query word weighting (parameter a) in the following manners: i) (f_1): $a = 1$, visual words (dimensions) are randomly selected. ii) (f_2): $a = 10$, query words are 10× more likely to be selected. iii) (f_3): $a = tf$, query words are more likely to be selected, which is similar to f_2 but

Datasets	Baseline	Spatial verification	t_1	t_2	t_3
Oxford 5K	0.612	0.645	**0.701**	0.693	0.645
Paris 6K	0.639	0.653	**0.700**	**0.700**	0.669

Table 5.2: Retrieval performance with different threshold, where the ordering function is f_6. The thresholds are set as: $t_1 = 0$, t_3 is the mean of visual word frequency in image and $t_2 = \frac{t_3}{2}$.

a is proportional to the term frequency of the query word, rather than constant as in f_2. As seen in Table 5.1, the results of f_2 and f_3 are more accurate than f_1. However, the difference between f_2 and f_3 is negligible when a is large. Thus, we use f_2 in the following experiments because it heavily weights query words and its implementation is simpler than f_3. Secondly, we fix $a = 10$ (f_2) and vary the weighting of query-relevant words (parameter b): i) f_4: $b = \frac{10}{8}$; ii) f_5: $b = \frac{10}{4}$; iii) f_6: $b = \frac{10}{2}$; iv) f_7: $b = 10$. The query-relevant words are collected by offline visual thesaurus, as described in Section 5.1.1. In this way, we aim to investigate the effects of increasing weights of query-relevant words in random dimension partition. As reported in Table 5.1, the retrieval performance increases when the weight of b enlarges. It plateaus when b is large enough, *i.e.* $b = \frac{10}{2}$, indicating that there are enough query-relevant words included. Therefore, we set $b = \frac{a}{2}$ (f_6) by default in the following experiments as it achieves the best performance on both datasets.

Partition threshold t Table 5.2 investigates the effects of threshold t in partitioning the dataset (Eq. (5.2)) by comparison of different thresholds. The threshold t, as shown in Figure 5.3, is used to generate context groups, which will affect the re-ranking performance. As seen from Table 5.2, $t = 0$ achieves the best retrieval performance and will be used in the rest of the experiments.

Range of top/bottom H Figure 5.4 (b) reports the retrieval accuracy with increasing top/bottom-H. The parameter H indicates the maximum number of top/bottom images to be considered in the context factor calculation. It can neither be too small (the context factor can not detect any top/bottom ranking information) or be too large (the context factor takes the whole dataset). As seen from Figure 5.4 (b), we obtain stable retrieval accuracy when H exceeds a threshold, $H = 200$, in which H is far less than the dataset size. Thus, we set

Datasets	System	System-baseline	Offline expansion	Online expansion	Off+online expansion
Oxford 5K	S1	0.612	0.701	0.696	0.703
	S2	0.645	0.700	0.703	0.706
	S3	0.806	0.814	0.825	0.830
Paris 6K	S1	0.639	0.700	0.705	0.705
	S2	0.653	0.704	0.709	0.709
	S3	0.769	0.770	0.777	0.773

Table 5.3: Performance for **S** obtained by online and offline expansion. $S1$: Baseline [106], $S2$: Spatial verification [106], $S3$: AQE [37]. The offline expansion is computationally cheaper compared to online expansion, while its performance is close to online expansion.

Methods		Oxford 5K	Paris 6K	Oxford 105K
Baseline		0.107	0.140	1.67
Spatial verification		2.10	4.71	4.34
f_2	$D = 1 \times 10^4$	0.030	0.034	0.44
	$D = 3 \times 10^4$	0.039	0.043	0.48
	$D = 5 \times 10^4$	0.045	0.052	0.51
	$D = 7 \times 10^4$	0.054	0.060	0.54

Table 5.4: Computational cost comparison of spatial verification and context based re-ranking. Note that we only calculate the run time of re-ranking with spatial verification and our method, while do not include the CPU time spent on baseline.

$H = 200$ as default.

5.2.2 Effects of query expansion

Secondly, we evaluate the selection of query expansion methods. The effects of the query-relevant words **S** are evaluated in Table 5.3. In addition to the baseline tf-idf similarity (S1), we also test the re-ranking applied to results obtained by spatial verification (S2) [106] and average query expansion (S3) [37]. Firstly, the retrieval accuracy is 14.5% (9.5%) higher than the baseline system (S1) on the Oxford 5K (Paris 6K) dataset, when **S** is formed by offline expansion. Online expansion is performed by including in **S** only spatially verified words (AQE in [37]). The difference between the retrieval results is minor, *e.g.* 0.701 *v.s.* 0.696 on the Oxford 5K dataset. Moreover, combining offline and online expansion leads to small improvement of mAP scores for S1, S2 and S3. Therefore, we use the computationally cheaper offline expansion in the experiments.

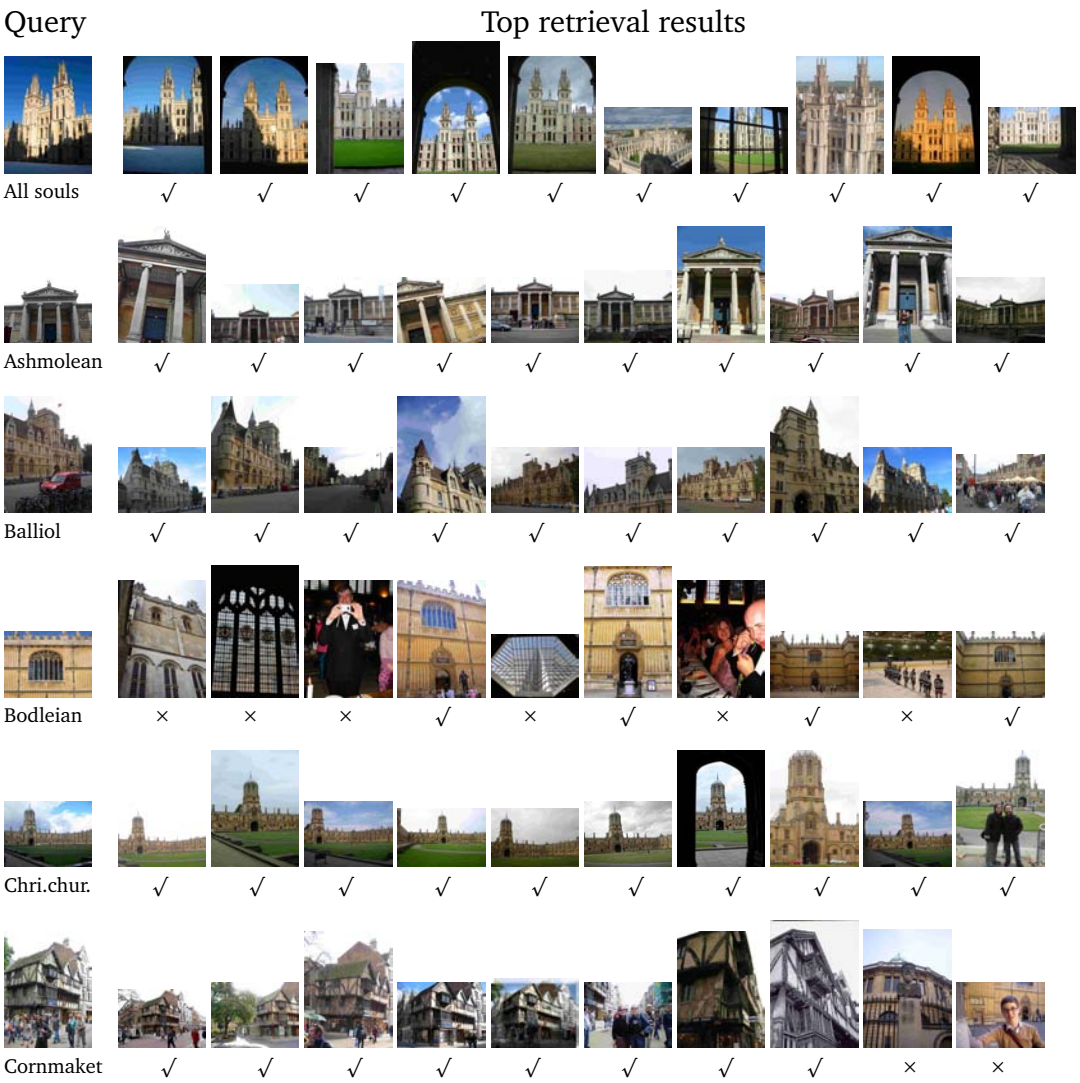


Figure 5.5: Top retrieval results of context based re-ranking method.

5.3 Discussion

We show some top retrieval results in Figure 5.5. The context based re-ranking is useful in filtering false positives, compared to the results shown in Figure 2.7 (baseline). Moreover, we compare the accuracy and computation cost of our method to the state-of-the-art methods that also requires a re-ranking process.

Computational cost As our context based re-ranking method makes use of an inverted file, it requires no more memory usage than baseline tf-idf matching. The

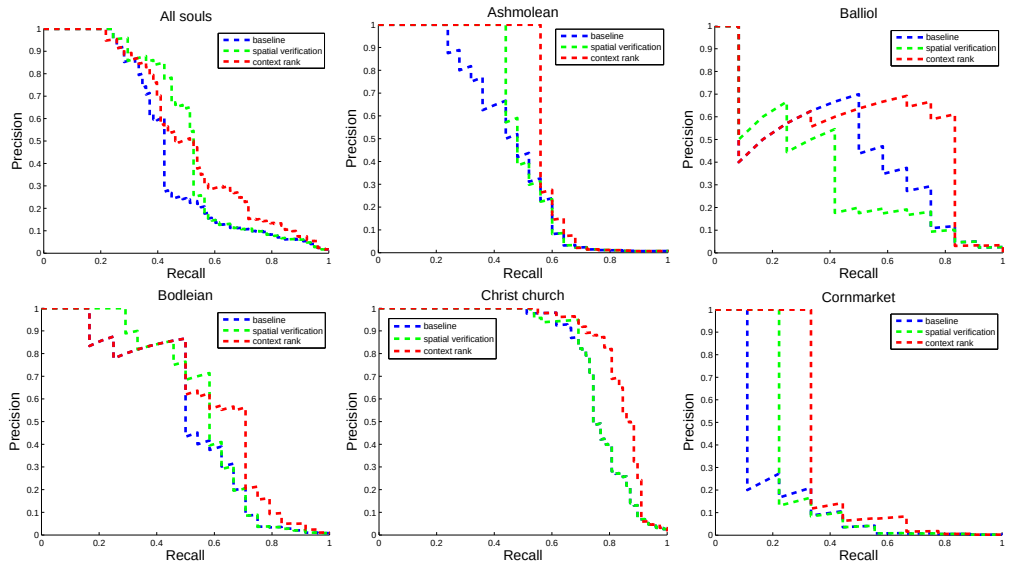
run time of our method is reported in Table 5.4. Firstly, we evaluate the run time by increasing D. This results consistent increase in CPU time, while improving the retrieval results as shown in Figure 5.4. In order to trade-off the effectiveness and efficiency, we set $D = 3 \times 10^5$ as shown in Table 5.4. Secondly, we compare our method ($D = 3 \times 10^5$) with the spatial verification method [106]. As seen in Table 5.4, our method is faster than spatial verification because it does not need spatial consistency examination, which is known to be expensive (costs 2.1 CPU seconds on the Oxford 5K dataset).

Comparison to the state-of-the-art methods We compare our method to other state-of-the-art re-ranking methods (Group B) in Table 5.5, including spatial verification [106], and query expansion methods [37, 34, 9]. Firstly, our re-ranking method outperforms standard spatial verification, both in retrieval accuracy and run time. It also achieves similar retrieval accuracy to QE baseline, but does not outperform other query expansion methods. Secondly, our method can also work with various query expansion methods: the initial retrieval results are re-ranked by our method and then applied with query expansion. As shown in Group C, this leads to further improvement of retrieval performance. The context based re-ranking method can also jointly work with methods proposed in previous chapters: total association and cross-word matching, as reported in Group D. Finally, we show some precision-recall (PR) curves of query examples from the Oxford 5K dataset, which compares the baseline, spatial verification, and our method in Figure 5.6.

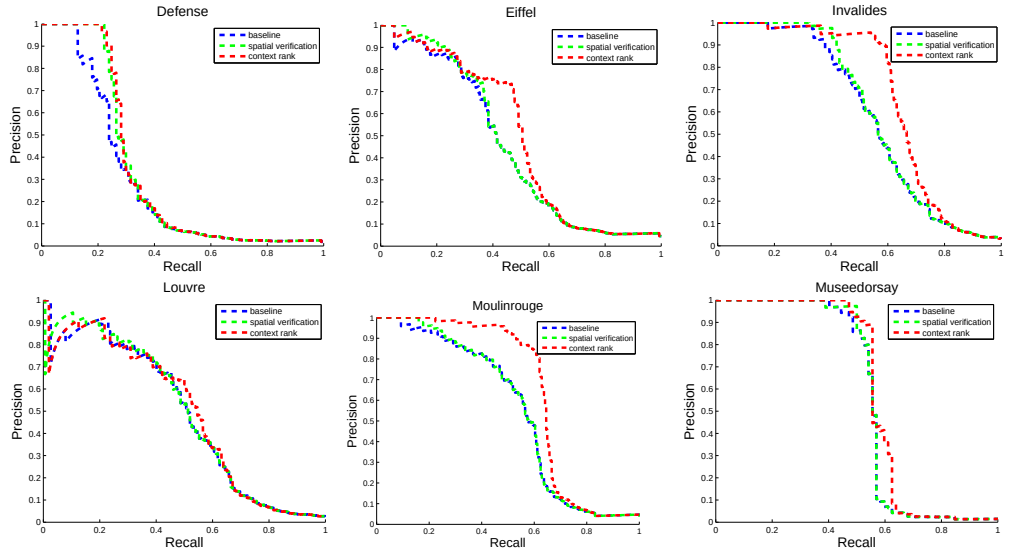
Note that our context based re-ranking method, unlike spatial verification, aims to refine the similarity measure instead of truncating the top ranked results by number of inliers. Therefore, our method is not reliable for verification of true (false) positives. The verification of true (false) positives, however, is essential in many applications of retrieval results. In next chapter, we will discuss the usage of ranking information in verification of the initial retrieval results.

5.4 Conclusion

In this chapter, we has proposed a simple yet effective image re-ranking method, which can be used in any retrieval system built on the BoW model. In contrast



Oxford 5K



Paris 6K

Figure 5.6: Examples of precision-recall (PR) curves of context based re-ranking: baseline, spatial verification and context based re-ranking.

to standard re-ranking methods, our method analyses the image ranks in terms of shared context rather than expensive spatial consistency examination. We explore the contextual information in two steps. Firstly, we use random space partition method to cluster the dataset into a large number of image groups. Secondly, the approximate image groups are used to refine the image similarity score. Images

	Methods	Oxford 5K	Paris 6K	Oxford 105K
	Baseline [106]	0.612	0.639	0.515
A	Visual word re-weighting (Chapter 4)	0.660	0.674	0.598
	Descriptor learning (non-linear) [108]	0.662 [108]	0.678 [108]	0.541 [108]
	Soft-assignment [107]	0.673 [107]	0.660	N/A
	Geometry-Preserving [159]	0.696 [159]	N/A	0.604 [159]
	Total association (Chapter 4)	0.700	0.682	0.680
	Spatial expansion ($\mathbf{F}'_{15}$, Chapter 3)	0.701	0.683	0.667
	Cross word (Chapter 4)	0.712	0.722	0.604
	Fine vocabulary [96]	0.742 [96]	0.749 [96]	N/A
	AUG [142]	0.776 [9]	N/A	0.711 [9]
	SPAUG [9]	0.785 [9]	N/A	0.723 [9]
B	Spatial verification [106]	0.649	0.655	0.571
	Context based re-ranking	0.701	0.700	0.585
	QE Baseline [37]	0.708	0.736	0.679
	iSP [34]	0.741 [34]	0.769 [34]	0.649 [34]
	Local geometry [105]	0.788	0.634	0.725
	DQE [9]	0.798	0.783	0.809
	AQE [37]	0.806	0.769	0.767
	Hello neighbors [114]	0.814 [114]	0.803 [114]	0.767 [114]
	Total recall II [34]	0.827 [34]	0.805 [34]	0.767 [34]
C	**Context based re-ranking**	0.701	0.700	0.585
	Total association+ **Context based re-ranking**	0.704	0.711	0.636
	Cross word+ **Context based re-ranking**	0.735	0.720	0.648
D	**Context based re-ranking** + AQE [37]	0.814	0.770	0.757
	Context based re-ranking + DQE [9]	0.832	0.793	0.790

Table 5.5: Comparison of context based re-ranking to the state-of-the-art methods. Group A: retrieval results of methods that modify the baseline before the query is executed (pre-process). Group B: retrieval results of methods that modify the baseline after the query is executed (post-process). Group C: retrieval results of combining context based re-ranking with our proposed methods. Group D: retrieval results of combining context based re-ranking with query expansion methods. Note that we cite the retrieval results of AUG [142] from literature [9].

having positive (negative) context factors are ranked towards top (bottom).

The experimental results illustrate that the context based re-ranking can improve the baseline method and some improvement methods that modify the baseline before the query is executed (Table 5.5 Group C). However, it does not outperform other re-ranking methods which are more computationally expensive, *e.g.* AQE and DQE. Compared to other methods in Table 5.5 Group B, our context based re-ranking method is more efficient in re-ranking process and is simple to implement. Therefore, it is useful for fast re-ranking of the retrieval results when required.

RANKING CONSISTENCY FOR IMAGE MATCHING AND OBJECT RETRIEVAL

CHAPTER VI

The standard BoW retrieval system applies fast computation of similarity measure between query/dataset image pairs, *i.e.* dot product similarity, but lacks robustness to varied image conditions: scale, viewpoint, lighting and partial occlusion of objects, as shown in Figure 1.2. In Chapter 5, we addressed this issue by applying contextual information to re-ranking retrieval results. In this chapter, we investigate the ranking consistency between image ranks. Similar to context based re-ranking presented in Chapter 5, we begin with the initial ranking results and re-rank them without any prior knowledge learnt from the dataset. In contrast to context based re-ranking, the re-ranking method proposed in this chapter is based on a fast verification method to remove false positives from the initial retrieval results.

We propose a ranking consistency examination, which is an alternative to previous methods relying on spatial consistency. It is observed that consistency in ranked results indicates that the corresponding query images are likely to contain similar content. An example is shown in Figure 6.1, in which similar images have common results in the top ranked results when they are used as queries, while results from dissimilar images have no intersection. Based on this observation, we propose to refine the image retrieval results with ranking consistency information, while retaining efficiency and not relying on low-level information, *e.g.* spatial or geometric feature information. The system framework is illustrated in Figure 6.2. We also show a number of applications using object retrieval results.

Firstly, we propose a simple yet effective image similarity criterion, named **ranking consistency**, in which the similarity between two images is measured by the similarity of the ranked lists that result from using them as queries. The usage of ranking consistency in the image domain is motivated by the ranking result comparison used in information retrieval. Figure 6.1 illustrates our key idea with

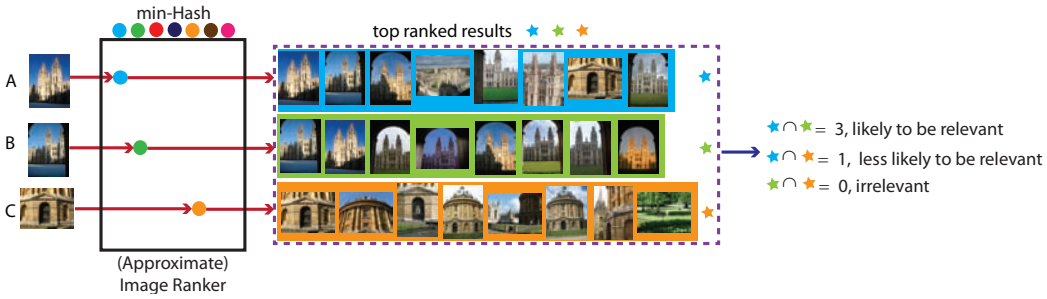


Figure 6.1: Ranking consistency overview. The examples are top ranked results of *All souls 1*, where the input images *A* and *B* are relevant, but both of them are irrelevant to image *C*. Our method generates some top ranked results for each image by list-wise min-Hash. The similarity between images is measured by the similarity between their top ranked results.

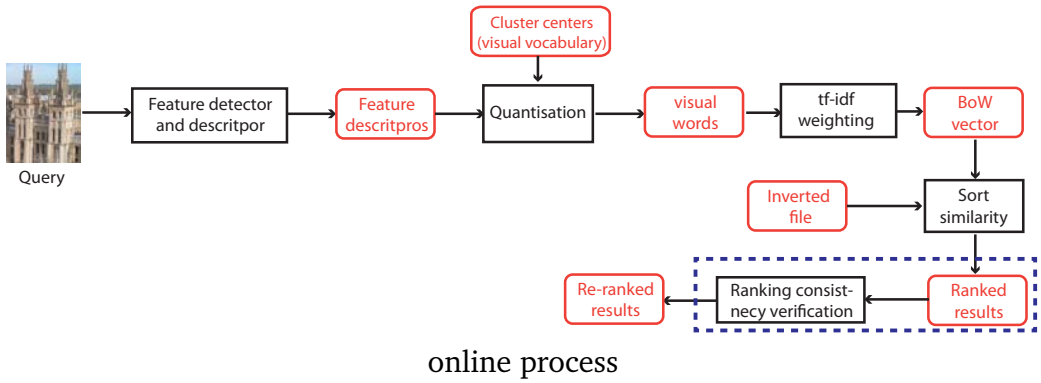


Figure 6.2: System framework of ranking consistency. Modules need to be changed are indicated in blue dash boxes. The offline process is unchanged.

some top retrieved results by similar query images A and B, and dissimilar images A and C. Note that image C is a highly ranked false positive result of query A. The retrieved results using image C as a query are completely different from the results of images A and B, supporting the fact that images with similar contents are consistent in their top retrieval results. The observation motivates us to use ranking consistency as a verification method: *images whose content matches a query can be inferred on the basis of their ranking consistency.* The ranking consistency criterion can work with any retrieval system, as it only requires ranks of images. In addition, our ranking consistency criterion does not require the comparison of any geometric information, unlike [106].

Secondly, we propose an efficient image re-ranking method, **ranking verification**, to re-rank an initial set of retrieved results by the embedded ranking

consistency information. The ranking verification requires online computation of query results for a fixed number of top ranked images (typically $K = 200$ in our implementation). Therefore, the re-ranking process is either inefficient using the standard normalized dot product vector comparison [130, 106] or less accurate if we use the approximate similarity comparison [38]. Instead, we use list-wise min-Hash to generate rapid approximate ranking lists of each image vector, whose consistency is then measured to evaluate the final image result ranking. The effectiveness of our method is due to two factors. First, our final ranking is the result of multiple min-Hash queries, so errors in individual queries can be tolerated. Second, we take into account multiple words from each hash function, which increases the average recall of the approximate method.

The ranking verification requires little extra computational cost per image. We only need to store the hash keys for each image instead of information about each feature which is required for spatial verification. The ranking consistency similarity can be intrinsically used in many image retrieval related problems, *e.g.* expansion of the query model, as an alternative to spatial verification. We also illustrate a graph structure of dataset images built on ranking consistency, which is useful for object mining in large image sets.

There has been a great deal of research to increase the accuracy of image retrieval by measuring the spatial consistency between the query and result images [106, 159, 35], as discussed in Chapter 2. However, these methods highly rely on the spatial or geometric information between pairwise images, and thus are less effective when the dataset images lack geometric information. In contrast, information retrieval has used the ranking information to enhance retrieval systems for many years. These methods consider ranking information in the following ways: i) learning to rank with relevance feedback; ii) re-ranking with results consistency.

Relevance feedback: Relevance feedback aims to refine the ranking model by some labeled data which is relevant or irrelevant to the query. This is known as the relevance feedback problem in information retrieval [120]. Relevance feedback is also used in some image retrieval systems to refine the ranking functions. These methods can be categorised into two groups. The first group of methods focus on formulation of a new query to take into account the relevant features to the original query. For example, query expansion [37] adds more relevant features from an

automatic scheme of sampling selection; Bayesian relevance feedback [47] needs users to identify retrieved images as being relevant or not, and then adjusts the query by Bayesian decision theory; Trademark retrieval [39] dynamically improves both the query formation and similarity measure with relevance feedback. The second group of methods usually use a pairwise ranking method, *e.g.* Ranking SVM [68] to sort the documents (or images) for given query [56, 73]. In order to train a ranking classifier, these methods need to know either some ranking preference in advance [56], or user provided information [73].

Results consistency: Results consistency uses the relevance of ranked results to improve retrieval performance as a post-processing step. There are a large number of ranking similarity measures, *e.g.* Spearman's ρ [134] and Kendall's τ [71]. The similarity measurement scores documents sharing many common results highly, which indicates the ranking consistency in these documents. The consistency of ranking has been considered in a number of image retrieval methods, such as [81, 102, 52]. In these works, the initial retrieved results are processed with some high level information, *i.e.* a relevance model to evaluate the linked text search results for similarity measurement [81]; a distance matrix defined by the similarity of ranking lists to take into account contextual information [102, 52]. However, these methods require expensive post-process.

Our ranking verification method is partly inspired by the neighborhood connection method [114], which also uses nearest neighbor results of dataset images to construct a network and thus dataset images are separated into close and far set for re-ranking. Conversely, our method is built on a collection of truncated ranking lists, as described in Section 6.1. In addition, our method uses hash-based search method to find similar images. To find the similar images efficiently, a similar hash-based method is proposed in [75], with inter and intra query expansion. The method needs to index each feature to the hash table, and compare the query features with all the features (millions to billions) in the dataset, which is very expensive in memory usage as noticed in [63]. However, our method hashes each image (as a BoW vector) which is more efficient than [75]. Furthermore, our method needs not rely on the geometric information.

6.1 Ranking consistency similarity

In this chapter, we introduce a **ranking verification** scheme which can efficiently and effectively re-rank the initial retrieved results returned by any retrieval system. The scheme is illustrated in Figure 6.2. Firstly, we describe how pairwise image similarity is measured by calculating the consistency of the image ranks that result from using each image as a query over a fixed dataset (Section 6.1.1). Secondly, this similarity measure leads to a method for re-ranking retrieval results which places the query-relevant images higher than the irrelevant images (Section 6.1.2). In the interest of efficiency, we produce approximate ordered results with list-wise min-Hash method (Section 6.1.3).

6.1.1 Ranking consistency measures

We use ranking consistency information to measure how similar two images are. As observed in Figure 6.1, images sharing common top ranked results also often share similar visual content, such as images A and B in Figure 6.1. This is known as the *ranking consistency* in information retrieval. Compared to spatial consistency [106], ranking consistency is more tolerant to image variations due to viewpoint or occlusion, and does not rely on the existence of a dominant rigid transform between matching objects in the images.

The ranking consistency criterion measures the similarity between a pair of images using the similarity of their ranked results, a.k.a. *ranking lists*, when each image is used as a query. Typically, the most salient members of ranking lists are those at the top, while lower ranked results are irrelevant. This means that we need only compare top ranked results. For any image i used as a query, its ranking list is a permutation of the dataset images sorted by descending similarity score, and truncated to a list of length h:

$$\mathbf{r}^{(i)}_{1:h} := [r^{(i)}_1, r^{(i)}_2, \cdots, r^{(i)}_h] \tag{6.1}$$

where h is the window size $h \ll V$, and V is the dataset size [1]. As a result, the image similarity is measured by the truncated ranking list similarity.

[1]Note that our method can be applied in any retrieval system as a post-process. In this thesis, the retrieval system is built on the BoW model. Therefore, the default ranked results are sorted by the descend normalized dot product similarity between tf-idf vectors.

Ranking list similarity has several standard solutions, which can be either top-weighted or not. The retrieval results contribute equally in the non top-weighted similarity, while the high ranked results are weighted more heavily in the top-weighted similarity. The widely used non top-weighted similarity in information retrieval includes Kendall's τ [71] and Spearman's ρ [134]. However, the top of the ranking list is more significant than the bottom in many real ranking cases. Therefore, top-weighted methods, *e.g.* , Yilmaz's τ_{AP} [157] and Melucci's τ_* [91], are better suited. We use two similarity measurements from the information retrieval literature to compare the truncated results.

The Jacarrd similarity The Jacarrd similarity between two ranking lists of images i and j is defined as the size of intersection divided by union:

$$J(i,j,h) = \frac{|\mathbf{r}^{(i)}_{1:h} \cap \mathbf{r}^{(j)}_{1:h}|}{|\mathbf{r}^{(i)}_{1:h} \cup \mathbf{r}^{(j)}_{1:h}|} \tag{6.2}$$

Note that $J(i,j,h)$ ranges from 0 (disjoint) to 1 (identical). The Jacarrd similarity does not include order information, and thus is non top-weighted similarity.

The rank biased overlap similarity Let X_d be the size of intersection of lists $\mathbf{r}^{(i)}$ and $\mathbf{r}^{(j)}$ to depth d (top-d ranked results):

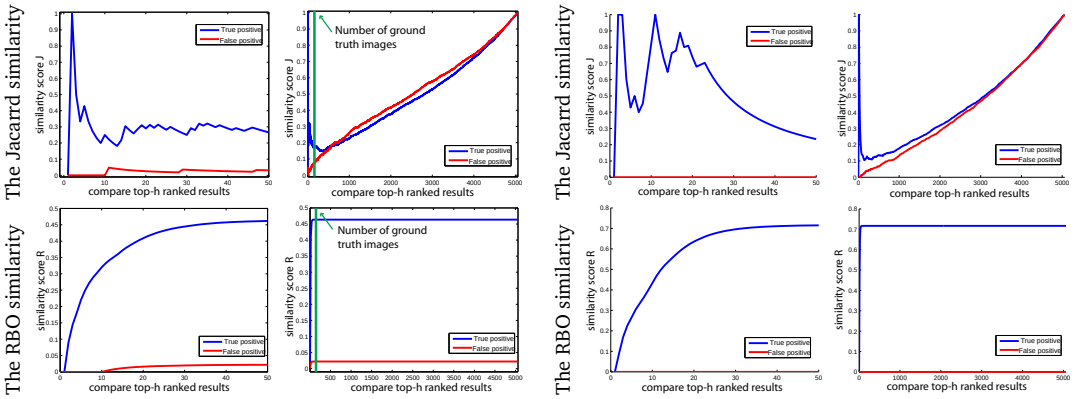
$$X_d(\mathbf{r}^{(i)}, \mathbf{r}^{(j)}) = |\mathbf{r}^{(i)}_{1:d} \cap \mathbf{r}^{(j)}_{1:d}| \tag{6.3}$$

The rank biased overlap (RBO) similarity [150] for a truncated ranking list is defined as:

$$R(i,j,p,h) = (1-p)\sum_{d=1}^{h} p^{d-1} \cdot \frac{X_d}{d} \tag{6.4}$$

where parameter p determines how steeply the weight declines. The smaller p is, the more highly top results are weighted ($p \in [0,1]$).

The effects of these two similarity measurements are illustrated in Figure 6.3 on two queries *All souls* and *Ashmolean*. The Jacarrd similarity relies heavily on the window size h, where it changes dramatically within range [0, 50]; when h is larger than the number of true positive results (*i.e.* when $h = 3000$), the similarity scores of positive results are close to or even higher than the ones of negative results. This is because most of the intersecting images between the ranking lists



(*a*) All souls. # of ground truth images : 78.

(*b*) Ashmolean. # of ground truth images : 25.

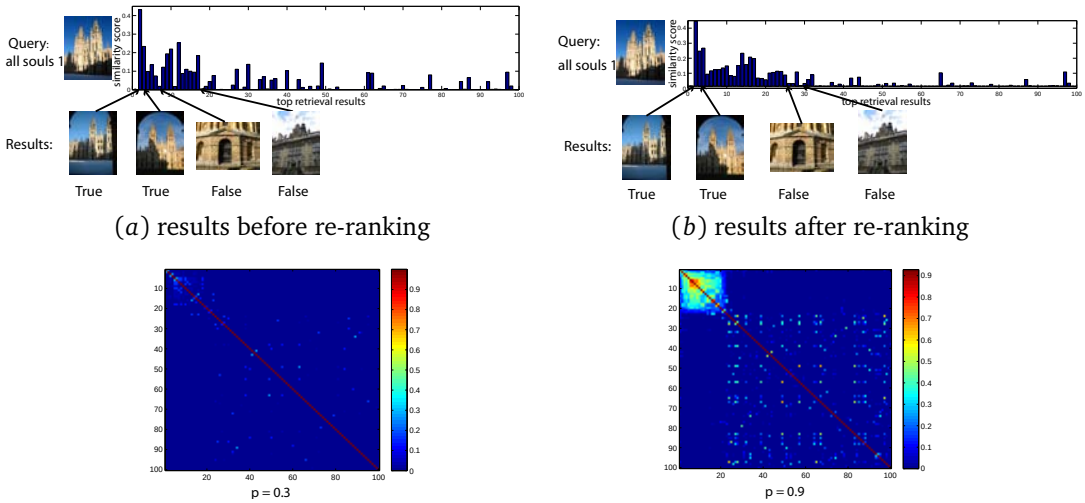
Figure 6.3: Examples of similarity scores computed by Jacarrd similarity and RBO similarity on various queries. The scores reflect the similarity between results from a query and a true (blue) / false (red) positives, where h ranges from 0 to 50 or h ranges from 0 to the full size of dataset. The true positive images present the object clearly, while the false positive images do not contain the target object. Note that the true / false positives are chosen from top-50 ranked results returned by the baseline.

are query-unrelated images. On the other hand, the RBO similarity score increases monotonically: $h_2 > h_1 \Rightarrow R(i,j,p,h_2) \geq R(i,j,p,h_1)$. As h increases, the similarity scores of true positives are always higher than the ones of false positives. In the following experiments, we set $h = [.005 \times V]$ by default for both the Jacarrd and RBO similarity measures unless mentioned. As seen in Figure 6.3, the RBO similarity is more suitable as a ranking list similarity measure than the Jacarrd similarity, because it weights the top ranked results more strongly than lower results. The weight is set as $p = 0.9$ for RBO similarity, as shown in Figure 6.4.

6.1.2 Result re-ranking using ranking consistency

Given a query image, the top-K retrieval results returned by a retrieval system might include some false positives. Our *ranking verification* method post-processes the top-K results such that highly ranked false positives are moved lower in the list.

The process is conducted by iteratively choosing the most similar images from the top results to the given query. First, we collect the top-K ranked results for the query image. We then use each result as a query to generate K ranking lists. After collecting K ranking lists, we conditionally select images into list $S := \{s_k\}_{k=1}^{K}$,



(a) results before re-ranking

(b) results after re-ranking

(c) 2D view after re-ranking, $p = 0.3$

(d) 2D view after re-ranking, $p = 0.9$

Figure 6.4: Examples of ranking consistency similarity for a particular query *All souls 1*. The ranking consistency similarity is measured by RBO similarity. (a) Retrieval results before re-ranking, where images are ordered by dot product similarity. (b) Retrieval results after re-ranking, where images are re-ordered by ranking verification. The similarity score in (a) and (b) is illustrated as ranking consistency similarity. The similarity matrix of ranking consistency similarity after re-ranking: (c) The similarity matrix of size 100×100 as the window size $h = [.005 \times 5K]$ and $p = 0.3$. (d) The corresponding similarity matrix of (c) when $p = 0.9$.

where k is the order after re-ranking of each element s_k. Given the first selection $S := \{s_1\}$ (usually the query itself), the second image (s_2) is chosen as that whose ranking list is maximally similar to the ranking list of s_1. Once chosen, the image is eliminated from the following selection. The remaining images are selected conditionally in the same way:

$$s_k = \underset{x \in \{x_i\}_{i=1}^{K} \setminus \{s_j\}_{j=1}^{k-1}}{\arg\max} \prod_{j=1}^{k-1} \text{sim}(x, s_j) \tag{6.5}$$

where s_k is decided by the previous selected elements $\{s_j\}_{j=1}^{k-1}$. The similarity score $\text{sim}(x, s_j)$ can be the Jacarrd similarity or RBO similarity, but is not limited to these two measurements. This process continues until K images have been selected. The top results are re-ranked by the order in S.

As shown in Figure 6.4 (a), there is highly ranked false positives in the initial ranked results. This is because the dot product similarity is inaccurate in measuring these query/dataset image pairs. The ranking consistency similarity is

able to distinguish the false positive by comparing their ranking lists. As seen in Figure 6.4, the ranking consistency similarity of the false positives is close to zero, while the ranking consistency similarity of true positives is relatively high. After re-ranking, highly ranked false positives are ranked down while the true positives remain at the top. The similarity of the corresponding re-ordered ranking lists are shown in Figure 6.4 (c) and (d), for RBO with different parameter p. As seen in Figure 6.4, p should be set large as to consider lower ranked results.

6.1.3 Fast approximate ranking list computation with list-wise min-Hash

The ranking verification involves computation of K ranking lists. By default, each ranking list is usually created by the normalized dot product similarity between tf-idf weighted BoW vectors [130]. This measure is widely used by many retrieval systems and is generally efficient enough for real time operation when it is only calculated once per query. However we now need to calculate it K times (typically $K = 200$). An accelerated method for ranking similar images is therefore necessary, such that the computation time for ranking lists is significantly reduced while the ranking list similarity is not materially affected.

In order to meet these requirements, we propose a **list-wise min-Hash** method to generate approximate ordered results of a given query. Our method is based on min-Hash, which is a popular hashing method for approximate near neighbor search. The min-Hash method is originally proposed to detect duplicate web pages [19]. In the image domain, it is applied for near duplicate image detection [36, 38] with particular design for the high dimensional BoW vector. It produces a random permutation of each visual word in the vocabulary as a hash function f. The hash key is extracted from each image i as the smallest element of the visual word set $\mathbf{v}_i$ under the permutation generated by a hash function f:

$$m(f, \mathbf{v}_i) = \arg\min_{w_k \in \mathbf{v}_i} f(w_k) \tag{6.6}$$

By repeating the hash function M times, each dataset image obtains a set of M hash keys. The similarity between two images (i, j) can be approximated by the number of collisions divided by M:

$$\text{sim}(\mathbf{v}_i, \mathbf{v}_j) = \frac{1}{M} \sum_{k=1}^{M} |m(f_k, \mathbf{v}_i) = m(f_k, \mathbf{v}_j)| \tag{6.7}$$

where M is the total number of min-Hash functions. This method successfully reduces the run time of ranking list generation, when $M \ll N$. However, it has limited effectiveness in near neighbor search because min-Hash considers subset of words in each image and thus is not robust to image condition changes. In order to improve recall in the near neighbor search, list-wise min-Hash uses multiple hash keys for the same hash function. The hash keys in our method are extracted from each image i as the v smallest elements of the visual word set $\mathbf{v}_i$ under the permutation generated by a hash function f:

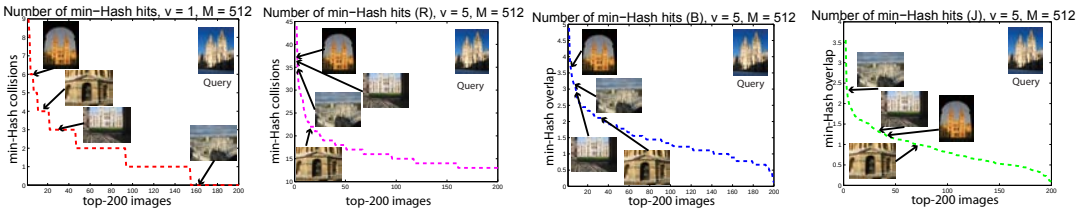
$$m^v(f, \mathbf{d}_i) = \{f(w_k)\}_{k=1}^{v}, w_k \in \mathbf{d}_i \tag{6.8}$$

where $f(w_1) < f(w_2) < \cdots f(w_k) < \cdots < f(w_v)$, v is the hash induced ordering. Therefore, each hash function maps to v (dependent) hash keys, so the collision between matching images is more likely in the face of varying image conditions. With M independent hash functions $\mathscr{F}$, the collisions embedded in the similarity measure can be calculated in the following ways:

$$\mathrm{sim}^v(\mathbf{v}_i, \mathbf{v}_j) = \begin{cases} \frac{1}{M}\sum_{k=1}^{M} |m^v(f_k, \mathbf{v}_i) \cap m^v(f_k, \mathbf{v}_j) \neq \emptyset| & \text{Binary} \\ \frac{1}{M}\sum_{k=1}^{M} J(m^v(f_k, \mathbf{v}_i), m^v(f_k, \mathbf{v}_j), v) & \text{Jacarrd} \\ \frac{1}{M}\sum_{k=1}^{M} R(m^v(f_k, \mathbf{v}_i), m^v(f_k, \mathbf{v}_j), p, v) & \text{RBO} \end{cases} \tag{6.9}$$

where $f \in \mathscr{F}$. In Eq. (6.9), a simple *Binary* collision measures two list-wise min-Hashes by the times that the intersection is not empty for M hash functions. Two ranking list comparison methods, the Jacarrd similarity (Eq. (6.2)) and the RBO similarity (Eq. (6.4)), are also suitable to measure pairwise list-wise min-Hashes.

Our method captures more collisions among visually similar images than the standard min-Hash method. In the standard min-Hash method ($v = 1$), the higher number of common visual word in two images, the higher probability of them to have same min-Hash values. This is a hard match, where each hash function, *i.e.* a hash function f only contributes to the similarity measure between pairwise images when the smallest elements of both ordered word sets are same. It is alleviated by cross-matching in list-wise min-Hash ($v > 1$). During the list-wise min-Hash generation, the hash keys extracted from each hash function are different visual word ordered by the same random permutation. A hash



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

Figure 6.5: Comparison of random collisions inf standard min-Hash ($v = 1$) and list-wise min-Hash ($v = 5$) in generating the ranking lists. The number of hash functions is $M = 512$.

function f contributes to the similarity measure between pairwise images when any intersection happens in the top v smallest elements, as shown in Eq. (6.8). Therefore, the cross-word matching of the similarity measure for $v > 1$ increases recall in varied image conditions compared to the standard ($v = 1$). Although this also has the potential to reduce precision, in practice this is not observed as the hash results are aggregated over several lists.

Figure 6.5 demonstrates the effects of list-wise min-Hash by measuring the similarity between top-200 results and the original query *All souls*. The standard min-Hash ($v = 1$) leads to unstable hits, *e.g.* a false positive obtains 4 collisions, which is higher than two true positives (Figure 6.5 (a)). The list-wise min-Hash ($v = 5$) is more accurate, where the false positive images accumulates less min-Hash overlap than the other true positives (Figure 6.5 (b)-(d)). Moreover, the overlap measure (B, J, R) has little effects on similarity scores as illustrated in the curves in Figure 6.5, because v is a small number. We set Eq. (6.9) with Jacarrd similarity by default unless mentioned otherwise. The retrieval accuracy of other overlap measures in Eq. (6.9) is further explored in Section 6.2.1.

Algorithm 10 briefly describes the full retrieval system of ranking verification with list-wise min-Hash. In summary, our ranking verification method can be separated into two processes: offline and online. The offline process extracts hash keys from each dataset images, while the online process re-orders the dataset images by selecting whose result rankings are consistent with the query image.

Algorithm 10 Ranking verification with list-wise min-Hash

Input: Dataset images and query image q.

Offline process:
Feature extraction and tf-idf weighting for all images (Section 2.2.3).
Represent dataset images as the BoW vectors: $\{\mathbf{d}_i\}_{i=1}^{V}$.
For $\forall f \in \mathscr{F}$, generate ν min-Hash values for each $\mathbf{d}_i$ (Section 6.1.3, Eq. (6.8)).

Online process:
Return top-K ranked result of query q: $\{x_i\}_{i=1}^{K}$ (Section 2.2.3).
Using each x_i as a query, obtain K result lists by list-wise min-Hash similarity (Section 6.1.3, Eq. (6.9)).
Re-ranking the top-K results (Section 6.1.2):
$S := \{s_1\}$, s_1 is the query image q.
for $k = 2$ to K **do**
    Select s_k according to the chosen images (Section 6.1.2, Eq. (6.5)).
    $S = S \cup s_k$.
end for
Return: Re-ranked results S.

6.2 Experiments

The experiments in this section show that our ranking verification method can be used to improve the initial retrieval results as a post-process stage. Furthermore, the ranking consistency similarity is used as an image similarity measure for both query expansion and unsupervised dataset mining. The experiments are organized as follows: i) re-ranking the top-K retrieval results, such that false positive results are ranked lower (Section 6.2.1); ii) shortlist generation for query expansion, such that quality query-relevant images can be added to the original query model (Section 6.3); iii) weights between images in building a dataset graph, which aims to cluster the images into different topics (Section 6.4). The experiments of issue i) are reported and discussed in this section; while the experiments of issues ii) and iii) will be described in Sections 6.3 and 6.4, respectively.

6.2.1 Experimental results of ranking verification

We show that ranking verification, as described in Section 6.1.2, can act as a substitute for spatial verification, but with greater efficiency and flexibility, as it does not rely on the presence of a strong geometric relation between the images.

Table 6.1 lists the methods that will be compared. $R0$ is the baseline retrieval system in which images are ranked by dot product similarity; $R1$ is a standard

$R0$	Baseline	Scoring: Dot product similarity of tf-idf weight vectors [130]. No post-process.
$R1$	Spatial verification	Scoring: $R0$ is re-ranked by the number of inliers detected by spatial verification [106].
$R2$	Ranking consistency	Scoring: $R0$ is re-ranked by ranking consistency similarity. Ranking list similarity: The Jacarrd similarity (Eq. (6.2)). Ranking List: normalized dot product.
$R3$	Ranking consistency	Scoring: $R0$ is re-ranked by ranking consistency similarity. Ranking list similarity: The RBO similarity (Eq. (6.4)). Ranking List: normalized dot product.
$R4$	Ranking consistency	Scoring: $R0$ is re-ranked by ranking consistency similarity. Ranking list similarity: The RBO similarity (Eq. (6.4)). Ranking lists: list-wise min-Hash similarity $v = 5$.

Table 6.1: Summary of various ranking list generation.

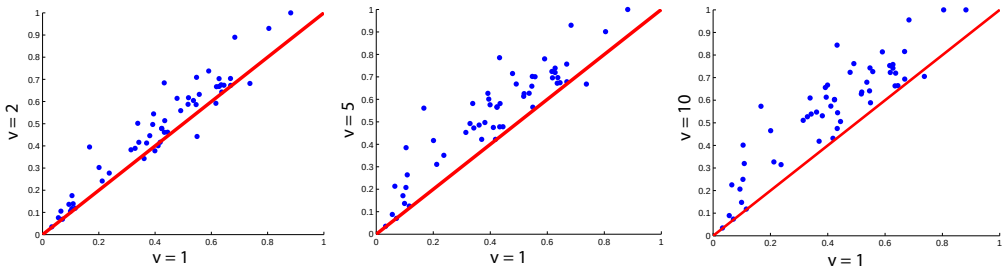
Dataset		R0	$R1$	$R2$	$R3$	$R4$
A	Oxford 5K	0.612	0.645	0.668	0.674	0.654
	Paris 6K	0.639	0.653	0.655	0.653	0.652
B	Caltech categories	0.379	0.381	0.397	0.397	0.403
	ImageNet (animals)	0.239	0.239	0.241	0.240	0.240
C	Oxford 105K	0.515	0.571	0.609	0.591	0.595
	Oxford 1M	0.455	0.524	0.553	0.543	0.527

Table 6.2: Retrieval performance comparison on five datasets (top-200). The window size for ranking consistency is set as $h = [.005 \times H]$, except on the Caltech categories ($h = [.03 \times H]$ for a large number of recalls.).

implementation of spatial verification to re-rank the initial top-K results from $R0$; $R2$, $R3$ and $R4$ are novel, each using a different implementation of our ranking consistency measure to re-rank the initial top-K results obtained by $R0$. Among them, $R2$ and $R3$ compute the ranking list similarity between ranked results returned by dot product similarity. In the case of $R4$, list-wise min-Hash is used to retrieve approximate ranked results for ranking list comparison. In this case, the min-Hash scheme obtains the hash keys by one of the distance measure: set similarity, weighted set similarity or approximate histogram intersection as described in [36, 38]. In our experiments, the distance measure is defined as set similarity [38] by default in both Eq. 6.6 and Eq. 6.8. The comparison is organized as follows:

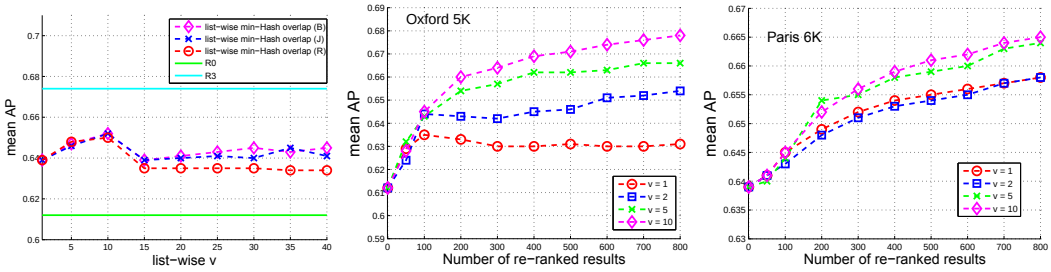
	Oxford 5K			Paris 6K		
Depth K	$R1$	$R2$	$R3$	$R1$	$R2$	$R3$
$K = 0$	0.612	0.612	0.612	0.639	0.639	0.639
$K = 100$	0.641	0.652	0.660	0.645	0.647	0.645
$K = 200$	0.645	0.668	0.674	0.653	0.655	0.653
$K = 400$	0.645	0.687	0.691	0.655	0.665	0.657
$K =$ FULL	0.646	0.711	0.717	0.655	0.702	0.690

Table 6.3: Retrieval performance with different number of re-ranked images (depth K) on the Oxford 5K and Paris 6K datasets. The window size ($R2$ and $R3$) is set as $h = [.005 \times H]$.



(a) (b) (c)

Figure 6.6: Retrieval accuracy with approximate near neighbor search. We compare the accuracy of different list-wise min-Hash: $v = 2, 5, 10$ to the standard min-Hash method $v = 1$.



(a) (b) (c)

Figure 6.7: Accuracy after re-ranking with various list-wise (v) min-Hash compared to baseline. (a) compares the re-ranking accuracy by the Binary similarity (B), the Jacarrd similarity (J) and the RBO similarity (R) used to measure the list-wise min-Hash overlap. The number of re-ranking images is fixed to $K = 100$. (b) and (c) compare the re-ranking accuracy with increasing number of re-ranked results, where the min-Hash overlap is measured by the Jacarrd similarity (J).

Performance with different ranking list similarity measure ($R2$ *v.s.* $R3$) : Tables 6.2 and 6.3 compare two kinds of ranking list similarity measurements, the Jacarrd similarity ($R2$, Eq. (6.2)) and the RBO similarity ($R3$, Eq. (6.4)) in

selection of the top-K results. The comparison is conducted in two steps. Firstly, we fix the number of images to re-rank (top-200 images) and evaluate the retrieval accuracy of $R2$ and $R3$ on different datasets (Table 6.2, $R2$ *v.s.* $R3$). As seen in Table 6.2, there is less than 7% variation in the mAP scores for all datasets. On the large scale datasets (Group C in Table 6.2), $R3$ performs slightly worse than $R2$. This is because the initial ranked results contain many errors at top ranks, while the RBO similarity (Eq. (6.4)) is top weighted. As a result, it causes the re-ranking performance slightly worse than using Jacarrd similarity, which is non top-weighted.

Secondly, the retrieval results of $R2$ and $R3$ both improve as the number of re-ranked images increases, as shown in Table 6.3. Therefore, both of the similarity measurements are effective in ranking verification. However, as described in Section 6.1.1, the Jacarrd similarity ($R2$) needs to choose the parameter h carefully, and thus we prefer the RBO similarity ($R3$) in the following experiments.

Performance with different number of re-ranking images (K) : Table 6.3 shows the retrieval accuracy after applying ranking verification with various verification list generation methods ($R1$, $R2$ and $R3$) on the top-K results. As K increases, the improvement of spatial verification ($R1$) is small, *i.e.* the difference of mAP scores is invisible when K increases from 200, 400 and all images. This is because spatial verification cannot determine true positives when there are few spatially consistent feature matches between the low ranked images and query. In contrast, our ranking consistency methods ($R2$ and $R3$) can improve the retrieval results when K enlarges and achieve better performance than the spatial verification. For example, $R2$ obtains 9.2% increase in mAP score while the spatial verification obtains 5.4% on the Oxford 5K dataset when both re-ranking the top-200 images. To be consistent with the number of re-ranking results in spatial verification [106], we apply re-ranking to the top-200 results in our ranking verification method.

Performance of list-wise min-Hash (v) : We illustrate the accuracy of list-wise min-Hash applied in approximate near neighbor search, which is examined in the following aspects: i) **Retrieval accuracy**. Figure 6.6 examines the accuracy of list-wise min-Hash search on the 55 queries of Oxford 5K dataset. Note that the dataset images are sorted by the sum of list-wise overlap (Eq. (6.9)), instead of normalized

$<v = 1, M = 256>$ 0.621	$<v = 1, M = 512>$ 0.633	$<v = 1, M = 1024>$ 0.657
	$<v = 2, M = 256>$ 0.645	$<v = 2, M = 512>$ 0.648
		$<v = 4, M = 256>$ 0.658

Table 6.4: Re-ranking accuracy with varying number of hash functions M and list-wise min-Hash v.

dot product similarity. Compared to the standard min-Hash ($v = 1$), list-wise min-Hash is more accurate when v enlarges. As seen in Figure 6.6 (c), almost all the query results ($v = 10$) move above the diagonal, which indicates that list-wise min-Hash ($v = 10$) produces more accurate ranking lists than the standard ($v = 1$). ii) **Re-ranking accuracy**. Figure 6.7 illustrates the accuracy of various list-wise min-Hash measures used to re-rank top-K retrieved results. Figure 6.7 (a) illustrates the effectiveness of the three overlap measures in Eq. (6.9) with increasing v, compared to the $R0$ and $R3$. Each measure shows the same overall pattern: a slight improvement for $1 < v < 15$ which is then negated for larger values of v as the list-wise min-Hash becomes less distinctive. Note that the re-ranking accuracy by approximate ranking list is higher than the baseline $R0$, but lower than the results of $R3$. Figure 6.7 (b) and (c) reports the re-ranking accuracy with the increasing number of re-ranking results K. The retrieval accuracy increases continuously with K, but at higher computational cost. Therefore, we set $v = 5$ and $K = 200$ in our experiments by default.

Table 6.4 illustrates the accuracy with various numbers of hash function (M) and list-wise min-Hash overlap sizes (v) used in re-ranking the top-200 results. It tests the trade off between numbers of hash function (M) and list-wise min-Hash overlap sizes (v). It is observed in Table 6.4 that we can obtain similar retrieval accuracy with higher value of v but fewer hash functions M. Table 6.2 summarises the re-ranking performance of $R0$-$R4$, which shows that ranking verification with approximate ranking lists ($R4$) can keep the retrieval accuracy of the results of $R2$-$R3$, and is close to spatial verification $R1$.

Performance of different distance measures used in min-Hash: Table 6.5 illustrates the effects of various distance measures used in min-Hash scheme as proposed in [36, 38]: i) set similarity (sim_s), ii) weighted set similarity (sim_w) and iii) approximate histogram intersection (sim_h), for both standard min-Hash

	Oxford 5K			Paris 6K		
# of phrase v	sim_s	sim_w	sim_h	sim_s	sim_w	sim_h
$v = 1$	0.633	0.636	0.649	0.649	0.648	0.645
$v = 5$	0.654 (R4)	0.650	0.661	0.652 (R4)	0.652	0.651

Table 6.5: Comparison of various similarity measures used in min-Hash scheme. Number of hash functions: $M = 512$ and depth $K = 200$. The similarity measures are: set similarity (sim_s), weighted set similarity (sim_w) and approximate histogram intersection (sim_h), as proposed in [38].

		bits	R0	Re-ranking			
				R1	R3	R4	[105]
f	Word IDs	20	√	√	√	√	√
f	Geometry	160	–	√	–	–	–
		24	–	–	–	–	√
f	tf-idf weight	32	√	√	√	√	√
h	Hash key	13	–	–	–	√	–
	Total (KB) :		16.25	66.25	16.25	20.41	23.75

Table 6.6: Average memory usage per image on the Oxford 5K dataset. f : bits per feature. h: bits per hash function. Our experiment generates (average) 2.5K features and $M = 512$ hash functions for each image.

Ranking list generation	Oxford 5K		Paris 6K		Oxford 105K	
	Run time	mAP	Run time	mAP	Run time	mAP
List-wise min-Hash (v = 1)	0.28	0.633	0.30		1.16	0.513
List-wise min-Hash (v = 5)	0.38	0.654	0.52	0.652	2.39	0.595
Dot product	27.21	0.674	31.66	0.653	≃ 10min	0.591
Spatial verification	2.10	0.668	4.71	0.655	4.34	0.571

Table 6.7: Average run time of re-ranking the top-200 results of a query on three datasets, measured by CPU second. Note that the results are computed sequentially.

($v = 1$) and list-wise min-Hash ($v = 5$). As reported in Table 6.5, the re-ranking accuracy is close in each group. This is also similar to retrieval results reported in Table 6.2, where ranking lists are generated by dot product similarity ($R2$ and $R3$). As seen in these experiments, our ranking verification method can make use of several min-Hash variations.

Computational cost : We illustrate the computational cost of our ranking verification method in re-ranking the top-200 results, with comparison to the widely used spatial verification method. **Memory usage**: Table 6.6 summarises the average memory cost for each image. Note that Table 6.6 compares various

re-ranking methods outlined in Table 6.1, except $R2$. This is because $R2$ is too expensive to run online. As seen in Table 6.6, spatial verification ($R1$) is the most expensive method in terms of memory requirement, because it needs to keep geometric information for each feature. Perd'och *at el* [105] shows that the geometry per feature can be minimized to 24 bits without dropping the spatial verification accuracy. Our ranking verification methods, $R3$ and $R4$, have less memory cost than the minimized geometry [105]. Compared to the baseline $R0$, the usage of hash keys to index images ($R4$) requires little additional memory usage per image. **Run time**: We compare the run time of re-ranking top-200 ranked results with ranking verification and spatial verification. In ranking verification, the majority of run time is spent on generating 200 ranking lists, which are obtained sequentially by the exact near neighbor search (dot product) or the approximate nearest neighbor search (list-wise min-Hash). In spatial verification, the majority of run time is spent on homography estimation between pairwise images. Therefore, the total run time depends the number of images to be examined and number of query relevant image can be found, instead of dataset size. As demonstrated in Table 6.7, the list-wise min-Hash method takes less than a second to compute 200 ranking lists, while the dot product requires about half a minute. The gap becomes larger on the Oxford 105K dataset, in which it is no longer feasible to calculate the ranking lists by dot product similarity. As a result, we use $R4$ by default in our ranking verification method.

6.2.2 Discussion

This section discusses ranking verification compared to other improvement methods. Firstly, Table 6.8 Group B compares the state-of-the-art re-ranking methods. Compared to them, our method performs better than spatial verification [106]. It does not outperform methods equipped with a query expansion step, *e.g.* AQE and DQE, which achieve high accuracy by a re-querying. However, our method requires less computation cost as well as no need of any prior knowledge about the dataset. Secondly, Table 6.8 Group C reports experimental results of ranking verification jointly working with methods presented in previous Chapters. The re-ranking helps to further improve the retrieval results. Finally, our ranking verification method can be embedded in AQE and DQE to obtain an effective shortlist as shown in Group D. In this Group, we show that ranking verification is also able to truncate

	Methods	Oxford 5K	Paris 6K	Oxford 105K
Baseline [106]		0.612	0.639	0.515
A	Visual word re-weighting (Chapter 4)	0.660	0.674	0.598
	Descriptor learning (non-linear) [108]	0.662 [108]	0.678 [108]	0.541 [108]
	Soft-assignment [107]	0.673 [107]	0.660	N/A
	Geometry-Preserving [159]	0.696 [159]	N/A	0.604 [159]
	Total association (Chapter 4)	0.700	0.682	0.680
	Spatial expansion ($\mathbf{F}/_{15}$, Chapter 3)	0.701	0.683	0.667
	Cross word (Chapter 4)	0.712	0.722	0.604
	Fine vocabulary [96]	0.742 [96]	0.749 [96]	N/A
	AUG [142]	0.776 [9]	N/A	0.711 [9]
	SPAUG [9]	0.785 [9]	N/A	0.723 [9]
B	Spatial verification [106]	0.649	0.655	0.571
	Ranking verification	0.654	0.652	0.595
	Context based re-ranking (Chapter 5)	0.701	0.700	0.585
	QE Baseline [37]	0.708	0.736	0.679
	iSP [34]	0.741 [34]	0.769 [34]	0.649 [34]
	Local geometry [105]	0.788 [105]	0.634 [105]	0.725 [105]
	AQE [37]	0.806	0.769	0.767
	DQE [9]	0.798	0.783	0.809
	Hello neighbors [114]	0.814 [114]	0.803 [114]	0.767 [114]
	Total recall II [34]	0.827 [34]	0.805 [34]	0.767 [34]
C	Context based re-ranking (Chapter 5)	0.701	0.700	0.585
	Total association+ Context based re-ranking	0.704	0.711	0.636
	Cross word+ Context based re-ranking	0.735	0.720	0.648
	Ranking verification	0.654	0.652	0.595
	Total association+ **Ranking verification**	0.708	0.687	0.682
	Cross word+ **Ranking verification**	0.738	0.720	0.658
D	Context based re-ranking+ AQE [37]	0.814	0.770	0.757
	Context based re-ranking+ DQE [9]	0.832	0.793	0.790
	Ranking verification + AQE [37]	0.764	0.755	0.742
	Ranking verification + DQE [9]	0.727	0.776	0.745

Table 6.8: Comparison of ranking verification to the state-of-the-art methods. Group A: retrieval results of methods that modify the baseline before the query is executed (pre-process). Group B: retrieval results of methods that modify the baseline after the query is executed (post-process). Group C: retrieval results of combining ranking verification with our proposed methods. Note that we cite the retrieval results of AUG [142] from literature [9].

Group D: retrieval results of combining context-based re-ranking or ranking verification with query expansion methods.

the ranking lists, similar to a spatial verification. Note that in these methods, ranking verification + AQE (DQE), the verification step is replaced by our ranking verification methods. As a result, the retrieval accuracy decreases simultaneously, compared to AQE or DQE using a spatial verification to truncate the ranking lists. More details will be discussed in next section.

6.3 Application I: Query expansion with ranking verification

Given a single query sample, query expansion (QE) methods refine the query model by adding more relevant features from its shortlist images. For example, QE baseline [37] adds visual words from the top-5 retrieved results (without spatial

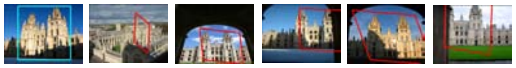
Query	Method	Shortlist images	# verified images
All soul	Spatial verification		24
	Ranking verification		16
Magdalen	Spatial verification		4
	Ranking verification		7

Table 6.9: Example of shortlist images generated by spatial verification (R1) and our ranking verification (R4). Note that ranking verification does not require a user-specified bounding box, and thus the verified results are not able to specify the location of the object.

verification); AQE [37] adds visual words from (up to) top-50 retrieved results, which are located in the corresponding regions matched to the query with spatial verification; DQE [9] treats the BoW vectors collected in AQE [37] as positive samples, and trains a linear classifier such that query-relevant images will be weighted more highly than others. Therefore, these QE methods rely heavily on the shortlist images because unrelated images will corrupt the expanded query model.

6.3.1 Shortlist generation and query expansion models

Usually, the shortlist is generated by spatial verification as proposed in AQE and DQE[2]. It sets a minimum number of inlier correspondences between each dataset image x_i and the query q. As discussed above, spatial verification is difficult to perform on widely separated images. Furthermore, it requires a bounding box to specify the query object, which is unavailable in some cases. We show that ranking verification provides a quality shortlist, in which query-relevant images can be selected by the ranking consistency instead of spatial consistency, and without the need for a bounding box. Let the shortlist initialized as empty $U := \emptyset$. Similarly to spatial verification, we set a threshold t for the ranking list similarity between each dataset image x_i and the original query q. A potential dataset image is added

[2]The shortlist used in QE baseline are the top retrieved results. We treat this method as baseline in comparison of QE methods.

$U := U \cup \{u_k\}, u_k = x_i$, if the similarity score $sim(q, x_i) > t$, and we stop when the length of shortlist is greater than 50.

6.3.2 Experimental results

We illustrate the effects of shortlist generated by our ranking verification on various query expansion (QE) methods. Firstly, the shortlist should contain query-relevant images. Table 6.9 shows some examples of shortlist images generated by these two kinds of verification methods. Compared to spatial verification, our ranking verification method is less sensitive to visual changes of target images. For instance, there are only 4 spatially verified results found for query Magdalen because the object contains few inliers for geometric matching. In contrast, the number of verified results enlarges when using our ranking verification method for query Magdalen . Figure 6.8 investigates the number of images verified by increasing thresholds: $t \in (0, 0.5]$ for ranking verification and up to 26 inliers for spatial verification. As the threshold enlarges, the proportion of true positives is close to 1 in the shortlist images found by both verification methods. Therefore, both of the methods can guarantee that the images collected are all true positives if the thresholds are high enough, but ranking verification collects, on average, more true positive results. The threshold for ranking verification is $t = 0.1$, *i.e.* any dataset images having lower ranking consistency score than 0.1 will be rejected in the verification process. Similarly, the threshold for spatial verification is the number of inliers returned by the RANSAC estimation. We reject dataset images having less than 10 inliers consistent to the original query in spatial verification.

Secondly, we show that ranking verification can act as a substitute of spatial verification in many techniques needing a verification process to obtain shortlist. Table 6.10 combines AQE [37], DQE [9] with our ranking consistency method. We replace the shortlist generated by spatial verification with that generated by ranking verification, as shown in Table 6.9. As we do not use a bounding box to specify object in the target images, it is difficult to surpass the performance of AQE and DQE with $R1$ (spatial verification). The retrieval performance is slightly lower on Oxford and Paris datasets, where the queries are rigid objects. However, our ranking verification method has advantage in dataset containing less geometric information, for example, the Caltech Categories dataset. As seen from Table 6.2 ranking verification can improve the retrieval performance in cases

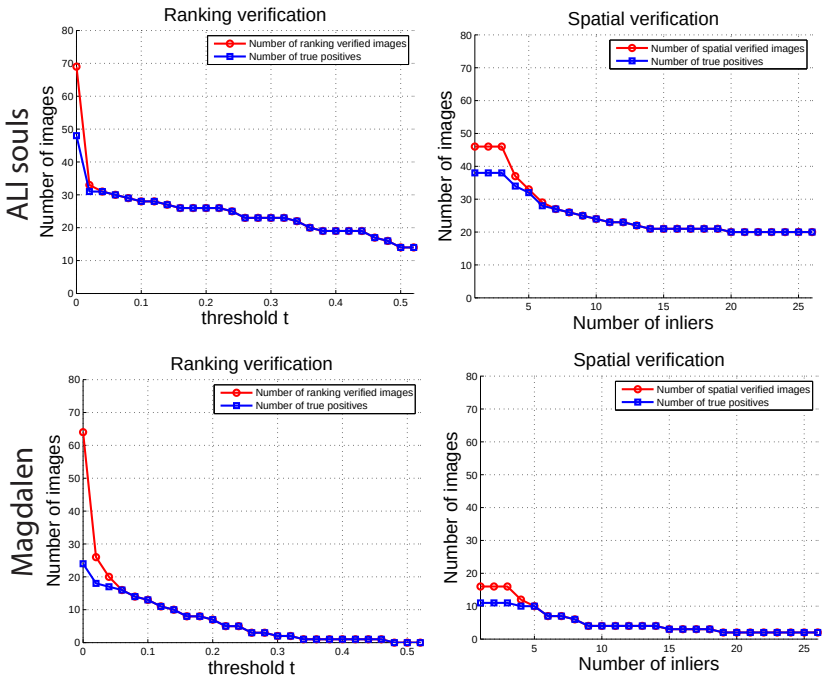


Figure 6.8: Illustration of the number of verified images *v.s.* the number of true positives obtained by ranking verification and spatial verification. The ranking verification helps to find more true positives for a fixed threshold.

dataset	QE baseline [37]	AQE [37]			DQE [9]	
	R0	R1	R4	iSP [34]	R1	R4
Oxford 5K	0.708	0.806	0.764	0.825 [34]	0.798	0.727
Paris 6K	0.736	0.769	0.755	0.722 [34]	0.783	0.776
Caltech Categories	0.497	0.447	0.519	N/A	0.313	0.520
Oxford 105K	0.679	0.767	0.742	0.761 [34]	0.809	0.745

Table 6.10: Retrieval performance of query expansion (QE) combined with ranking consistency method. We report the results of QE baseline [37], AQE [37] and DQE [9] from our implementation, which have slight difference in baseline results compared to these papers. The results of incremental spatial verification (iSP) is cited from Total Recall II [34]. Note that our method does not need to use a bounding box to specify the object, and thereby collect all the visual words appearing in the shortlist in AQE and DQE.

where spatial verification can not identify objects in the target images, on the Caltech Categories dataset. Consistently, both the performance of AQE and DQE with $R4$ can outperform $R1$ in those datasets depicting objects with less rigid spatial structure. Therefore, ranking verification is more adaptive to various image datasets, and we show its further usage in image dataset mining below.

6.4 Application II: Discovery of dataset images

Unsupervised topic discovery in image datasets is an active area of research. This is usually conducted on a matching graph, which is built on the dataset images, such that one can apply graph-based methods to segment the graph, and thus obtain the visual content of the whole dataset. A relational graph, $\mathbf{G} = (\mathbf{R}, \mathbf{E}, \mathbf{W})$ with nodes $\mathbf{R}$, edges $\mathbf{E}$ and weights $\mathbf{W}$, is constructed on the basic of similarity between pairwise dataset images. Each dataset image is treated as a node, two nodes i and j are linked by a weighted edge according to the similarity between i and j. As a result, the graph $\mathbf{G}$ relies heavily on the similarity measurement of images.

In previous work [111, 109, 142], the weight between a pair of nodes in the graph is proportional to the number of inliers detected by spatial verification between these two images. In this section, we show that nodes can be connected with ranking consistency similarity. The edge weight $W(i,j) \in \mathbf{W}$ between node i and j is defined as: $W(i,j) = sim(x_i, x_j)$, where $sim(x_i, x_j)$ is the RBO similarity between the ranking lists of x_i and x_j, $sim(x_i, x_j) \in [0,1]$. After building the graph $\mathbf{G}$, we apply Hierarchical Authority Shift (HAS) [32] to automatically cluster the nodes $\mathbf{R}$. The clustering does not require pre-defined number of clusters.

6.4.1 Experimental results

We adopt the Oxford dataset where each image has been manually labeled according to its visual quality: "good", "ok", "junk", and "unseen". We build up the graph $\mathbf{G}$ on a collection of images which include all "good", "ok', "junk" images containing the buildings and the rest of them are random "Unseen" image, as illustrated in Table 6.11. Every image in the corpus is treated as a query, which results in an edge between the query and its verified retrieval results if exist. After building the graph, we apply Hierarchical Authority Shift (HAS) [32] to cluster the dataset, such that each image will be assigned to one of the clusters. Figure 6.9 shows some clustering results of the Oxford dataset, each of them describes different major views of the buildings.

To quantify the clustering performance, we calculate the average coverage of cluster results on different graphs, which are built on spatial verification and ranking verification, respectively. That is, for each cluster, we count the number of images belonging to different labels: "good", "ok", "junk", or "unseen". The

Figure 6.9: Clustering results of the Oxford dataset. The graph data includes "good", "ok" and "junk" images. Note that we show parts of the results in some clusters that are too large to display.

	# image	Ranking verification		Spatial verification	
		# clusters	avg. rate	# clusters	avg. rate
good + ok	567	51	0.975	28	0.977
good + ok + junk	845	103	0.950	46	0.973
good + ok + junk + unseen (500)	1345	165	0.900	59	0.915
good + ok + junk + unseen (1000)	1845	232	0.886	76	0.826

Table 6.11: The average coverage of dataset images via different graphs built on spatial verification and ranking verification.

coverage of one cluster is the fraction of the number of matching images to the total number of images contained in this cluster. Note that we ignore the clusters if they are mainly "unseen" images. The overall score for the dataset clustering is the average of these coverage scores which can be used to judge the overall connection of the graph. Detailed results are reported in Table 6.11, where the graph built on spatial verification performs well on segmentation of relatively clear images. However, its accuracy drops dramatically when there is increasingly noisy data, as shown in Figure 6.10. The graph built on ranking verification is less sensitive to the image condition changes, as it is not easily disturbed by noisy data.

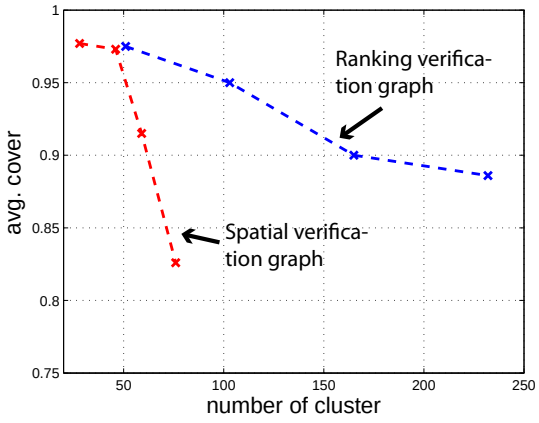


Figure 6.10: The average coverage of dataset images via different graphs built on spatial verification and ranking verification, corresponding to Table 6.11.

6.5 Conclusion

In this chapter, we have proposed an image matching framework by using ranking consistency, which aims to explore the underlying ranking relationships among images. The proposed image matching framework introduces ranking consistency into the process of ranked results verification, leading to a more robust similarity measure as well as retrieval results. To ensure the efficiency of ranking consistency, a list-wise min-Hash scheme is developed to accomplish the task of an approximate similarity ranking for large scale image datasets. Because of its efficiency and effectiveness as a retrieval post-process, our ranking consistency method is easily integrated into various retrieval-related applications (*e.g.* query expansion and scene summarization). Experimental results have shown the flexibility and efficacy of the proposed image matching framework while retaining the computational efficiency.

OBJECT RETRIEVAL WITH GROUP-QUERY

CHAPTER VII

In previous chapters, we have discussed a number of methods built on the BoW model to improve the retrieval accuracy. These methods make use of several auxiliary steps to improve the image representation or similarity measure. Chapter 3 has illustrated the improvement of the BoW representation by a spatial expansion, in which a visual thesaurus is proposed to collect spatially related visual words offline. Chapter 4 has presented enhancements for the standard image similarity measure. These methods use an offline learning stage to collect visual word correlation and importance, and improve the query model online. Chapters 5 and 6 discuss the improvement of image similarity used for ranking dataset with embedded rank information obtained online. These two methods do not need to change the offline stage.

Despite encouraging results, these methods have difficulty in capturing the diverse distribution of possible appearances of the query object, leading to a strong dependence on query image quality. As shown in Figure 7.1, the performance when retrieving the same building object varies dramatically: the retrieval case with front viewpoint (I) in Figure 7.1 has significantly higher accuracy than the other cases (the right side viewpoint (II) and left side viewpoint (III) in Figure 7.1). In the cases of (II) and (III), query expansion [37] usually fails because there are not enough true positives detected by spatial verification, as investigated in [34].

Standard object retrieval is a query-by-example problem, in which image similarity is examined between a simple query and dataset images. Less common is the use of multiple query images to specify a single object. For example, image sharing websites such as FLICKR or Facebook group images into communities containing the same or similar subjects. Typically, each community contains images of the same object from varying viewpoints. We define a **group-query** as a small collection of images, such that the target object (used for query) is depicted as a

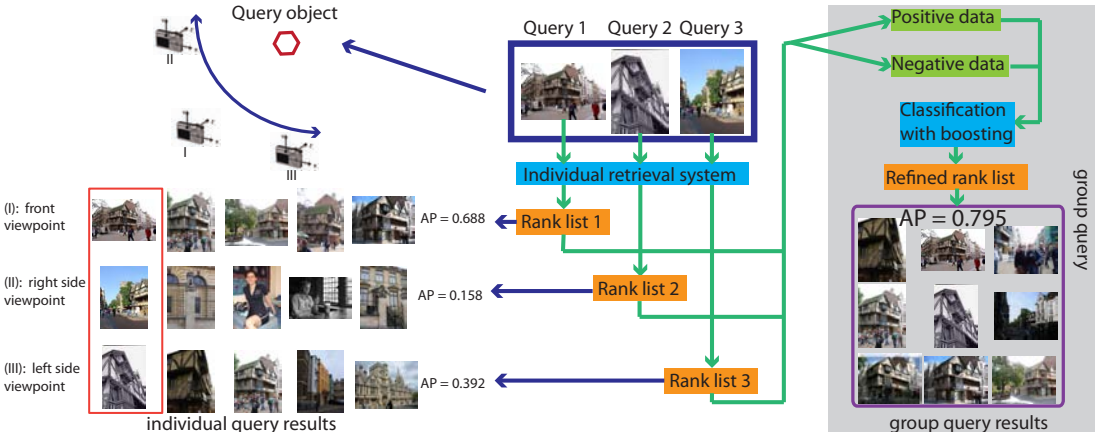


Figure 7.1: Illustration of group object retrieval method, using 3 query images of the same object but taken from different viewpoints. Individual query results are mined for positive and negative examples, which are then used to train a boosted classifier. The classification function is then used to rank image results. Typically, it obtains higher accuracy than is possible from any individual query.

variety of images containing it.

Based on this, we introduce a group-query based object retrieval method (illustrated in Figure 7.1), which uses a small group of images of a query object to reflect its appearance variation (*e.g.* different viewpoints). The aim of group-query is to find all images in the dataset that contain the same object. Intuitively, group-query more accurately describes the target object than retrieval with a single query. However, it involves two issues: i) the description of the input group-query; ii) the ranking function to sort the dataset images. We address the first issue by an automatic collection of relevance feedback, as discussed in Section 7.1. The second issue is more complicated, because the query includes more than one instance image. This can be done by two types of ranking function: the standard ranking function with an average query vector or a discriminative function with a set of query vectors. Section 7.2 discusses the group-query retrieval by averaging the image vectors, which has been detailed in [37, 8] [1]. Section 7.3 discusses our discriminative ranking function used for ranking the dataset images, which is built on previous methods [115, 9].

In this chapter, we prefer the discriminative function because it is more flexible to various query instances. The outline of our discriminative method is

[1]We did the similar work [30] of group-query with [8] at the same period.

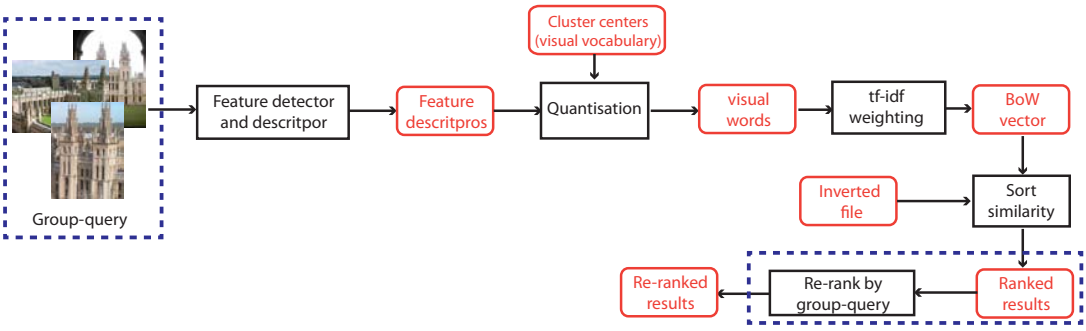


Figure 7.2: System framework of BoW based retrieval system, focusing on group-query and ranking function (offline process). Modules need to be changed are labeled in dash box. The offline process is unchanged.

described as follows (also see Figure 7.1):

1. A group-query is defined as the set $\mathscr{Q} = \{\mathbf{q}_i\}_{i=1}^{M}$, where each instance $\mathbf{q}_i$ is represented as a tf-idf vector and M is the number of query instances.

2. Let $\mathscr{R} = \{\mathbf{r}_i\}_{i=1}^{M}$ denote the corresponding set of ranking lists, where each ranking list $\mathbf{r}_i$ is obtained by performing pairwise dot product based image matching between the query $\mathbf{q}_i$ and the dataset images.

3. Organize training samples $\mathscr{T}$, which contain positive and negative image samples selected from $\mathscr{R}$.

4. Train a discriminative ranking function $F(\mathbf{d})$ with the training samples $\mathscr{T}$.

5. Re-rank the dataset images according to the weights given by the discriminative ranking function $F(\mathbf{d})$.

Among these steps, we pay attention to designing an effective and efficient ranking function for capturing the underlying affinity relationships between $\mathscr{Q}$ and the database images, such that relevant images are highly ranked while irrelevant images are lowly ranked. Figure 7.2 shows the modules proposed in this chapter.

Related work: One of the few previous works to exploit the group-query information is [115], where users input multiple query images as positive samples of a class, along with negative images that do not contain the object.

Using these positive and negative samples, a discriminative classification model is learnt to rank all images in the dataset. Consequently, the learnt ranking model [115] is independent of the retrieval database, and requires a large number of predetermined positive images (provided by users) and negative images (collected offline) for high retrieval accuracy. Alternatively, a target object in the dataset can be matched by a discriminative relevance evaluation, where positive and negative queries are used to obtain the mutual information score [92]. A discriminative ranking criterion is well suited to the use of multiple query images as it models the set of positive samples non-parametrically, and can therefore accommodate a diverse set of image views. It also naturally benefits from the addition of extra positive and negative samples.

7.1 Forming a group-query and training samples

We form a group-query $\mathcal{Q} = \{\mathbf{q}_i\}_{i=1}^{M}$ as a set of related images containing the same object, as examples shown in Figure 7.1. The standard ranking function compares image similarity by one-to-one comparison in the query-by-example methods. However, the group-query needs to consider the similarity between a dataset image and a small collection of images. We process the group-query by two kinds of ranking functions: i) average the group-query, such that the ranking function is the same as the standard retrieval system (dot product similarity), which is discussed in Section 7.2. ii) treat the group-query as a whole, and train a classifier to separate the dataset images, which is discussed in Section 7.3.

Both of these methods require training samples to be collected in advance. Initially, we obtain training samples $\mathcal{T} = \{(\mathbf{d}_i, y_i)\}_{i=1}^{K}$ where $\mathbf{d}_i$ is the tf-idf vector and $y_i \in \{1, -1\}$ indicates whether the corresponding image contains the object. Images (tf-idf vectors) with at least n matches to any query image are provided as positive examples and images with the lowest ranked non-zero similarity scores are taken as negative examples. There are many ways to collect positive samples from query instances, for example spatial verification [106] or ranking verification presented in Chapter 6. These methods are able to collect reliable query relevant images from the retrieval results. In this chapter, we firstly adopt spatial verification in training data collection, same as [9], in Section 7.2. Section 7.4.3 will discuss the retrieval results with ranking verification.

For each query instance $\mathbf{q}_i$, we use spatial verification [106, 37] to collect positive training samples that have a minimum number ($\geq$ 10) of spatially consistent matches and select the lowly ranked samples in $\mathbf{r}_i$ as negative training samples, as done in [9].

7.2 Average group-query retrieval

The averaging function directly uses the image vectors contained in $\mathscr{Q}$ to form a new query vector. The dataset images are sorted by the standard dot product similarity between query/dataset images. It can be conducted in two ways as described below:

- Averaging group-query: Using the group-query $\mathscr{Q} = \{\mathbf{q}_i\}_{i=1}^{M}$, we are able to represent the object by averaging tf vectors of $\mathscr{Q}$, similar to the average query expansion (AQE) [37] (Eq. (2.13)). The averaged vector is used to query the dataset, and the retrieval process is the same as the standard method.

- Averaging positive training data: The query instances in $\mathscr{Q}$ are sometimes not sufficient to describe an object, thus we expand them with positive training data samples $\mathscr{T} = \{(\mathbf{d}_i, y_i = 1)\}_{i=1}^{K}$ obtained by spatial verification. This is useful when the input images can not well describe the target object. The retrieval process is the same as the standard method.

Either method described above is effective when the query images are clear and related. However, the input query might contain noisy instances from different buildings, *e.g.* there is one instance *Ashmolean* (highlighted in red) that does not belong to *All souls* as shown in Figure 7.3. Note that the averaging function assumes all query images are similar and treats each of them as the same. The retrieval results therefore highly depend on the number of quality query instances can be collected in the training samples. As shown in Figure 7.3, the top retrieval results contain a couple of images of *Ashmolean*. These images are removed from the top ranked results after averaging with the positive training data, which are mostly of *All souls*, as shown in Figure 7.4.

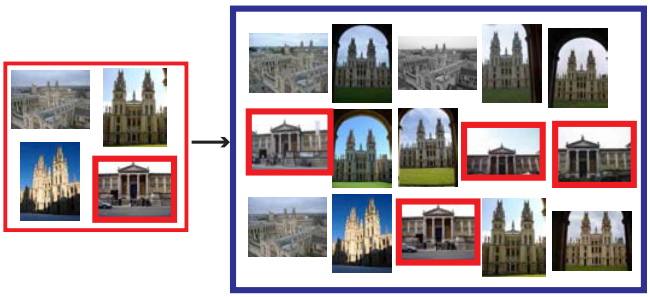


Figure 7.3: Retrieval results of noisy group query. The group-query contains one instance that is different from others (highlighted in red). The retrieval results are calculated by averaging group-query.

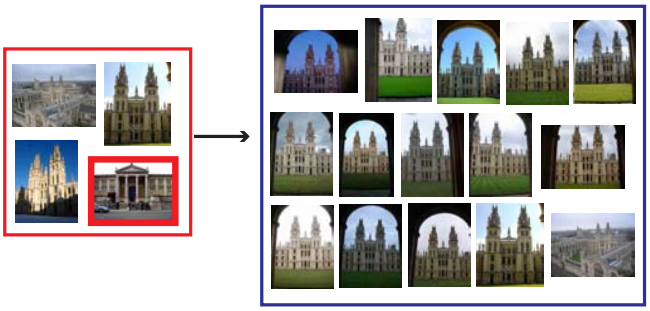


Figure 7.4: Retrieval results of noisy group query. The group-query contains one instance that is different from others (indicated in red). The retrieval results are calculated by averaging positive training data.

7.3 Discriminative group-query retrieval

In this section, we discuss the discriminative ranking function for group-query retrieval. In contrast to the averaging ranking function, dataset images are sorted by weights learnt from positive and negative training data. This involves a training stage to obtain a decision boundary between positive and negative samples, and thus the discriminative ranking function should be efficient during run time. In this section, we apply a linear SVM and a non-linear boosting classifier.

7.3.1 Discriminative ranking function with linear SVM

Using the training data samples $\mathcal{T} = \{(\mathbf{d}_i, y_i)\}_{i=1}^{K}$, we are able to train a *discriminative* model to separate the positive and negative data, as illustrated in

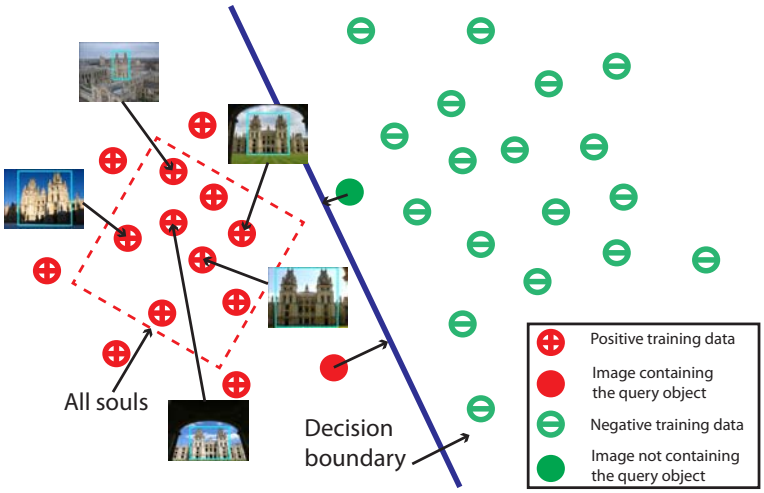


Figure 7.5: Example of multiple samples used as query *All souls*. A discriminative classifier is trained by linear SVM, in which the positive data is collected from the shortlist of these samples.

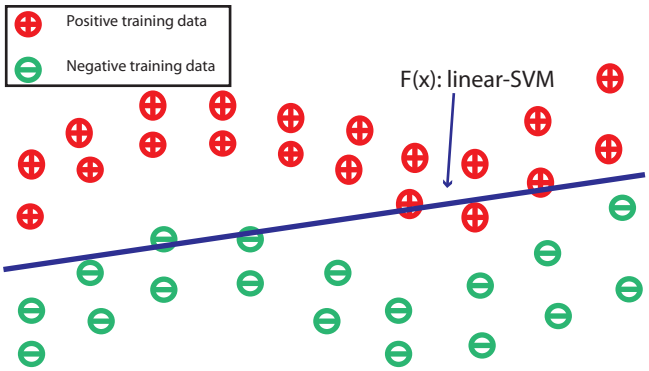


Figure 7.6: Illustration of training data separation by a linear SVM classifier.

Figure 7.5. The group-query, *All souls*, is now described by 5 images with different views of the building. A linear SVM classifier is efficient and effective in solving this problem, as used in DQE [9]. Our method, following DQE, trains a linear SVM using these positive and negative BoW vectors to obtain a weight vector $\mathbf{w}$. The mathematical formulation of learning a linear SVM classifier is given as follows:

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \quad & \tfrac{1}{2}\mathbf{w}^T\mathbf{w} + C\textstyle\sum_i \xi_i \\ s.t. \quad & y_i(\mathbf{w}^T\mathbf{d}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \end{aligned} \tag{7.1}$$

where C is a trade-off control factor.

After learning the weight vector $\mathbf{w}$, the ranking score between a dataset image and the group-query is the signed distance to the decision boundary, as illustrated in Figure 7.5. Mathematically, the ranking score can be measured by value $\mathbf{w}^T\mathbf{d}$, where $\mathbf{d}$ is the BoW vector of a dataset image. However, a linear separation is unable to capture non-linear discriminative information, as illustrated in Figure 7.6. As a result, we propose a non-linear classifier to rank the dataset images but still keep the efficiency of a linear SVM.

7.3.2 Discriminative ranking function with boosting

In this section, we learn a ranking function that aims to capture non-linear discriminative information from the training data samples. Linear SVM classifiers are adopted as weak learners in a boosting framework [45, 147] due to their simplicity and efficiency. Using ensemble learning, these linear SVM classifiers can be adaptively combined to generate a strong classifier. In each boosting iteration, a linear SVM classifier is learned over a subset of $\mathscr{T}$, which is obtained by weighted random sampling over $\mathscr{T}$. Instead of simple linear SVM ranking functions used in [9], our boosting-like ranking function effectively captures nonlinear discriminative information by constructing a nonlinear decision boundary using an additive linear approximation, as illustrated in Figure 7.7. To cope with the nonlinear classification problem, non-linear kernel SVM is an alternative, but prediction is usually computationally expensive, making it impractical for scalable object retrieval. Therefore, our ranking function is more suitable because it can not only improve the retrieval effectiveness but does so with low computational complexity as [9]. Algorithm 11 provides the details of constructing the boosting framework.

Moreover, as most computation of Algorithm 11 is spent on training linear SVMs, its run time can be reduced by parallelising SVM training before boosting iteration, similar to [76]. We do this by collecting a weak classifier pool of linear SVM before boosting from random positive and negative samples from $\mathscr{T}$. In effect, steps 5 and 6 in Algorithm 11 are executed in parallel before the boosting iteration begins. After training the discriminative ranking function, an approximate ranking function $F(\mathbf{d})$ is formed to fit the data by selecting from these weak classifiers during each boosting iteration: $F(\mathbf{d}) = \sum_{t=1}^{T} \alpha_t \cdot f_t(\mathbf{d})$. By applying the ranking function F, each dataset image $\mathbf{d}$ will be assigned a signed ranking score. The new

Algorithm 11 Ranking function learning using Adaboost-LinearSVM

Input: Training samples $\mathcal{T} = \{(\mathbf{d}_i, y_i)\}_{i=1}^{K}$, maximum number of boosting iteration T.
Output: Ranking function $F(\mathbf{d})$.

Initialize: Data distribution: $D_1(i) = \frac{1}{K}$, $\forall i$ and $t \leftarrow 1$.
while $t < T$ **do**
    Weighted random sampling positive and negative data from $\mathcal{T}$ with distribution D_t.
    Train a linear SVM $f_t(\mathbf{d})$ on the training set.
    Calculate the training error: $\epsilon_t = \sum_{i=1}^{K} D_t(i) I(y_i \neq f_t(\mathbf{d}_i))$, where I is the indicator function.
    Calculate the weight: $\alpha_t = \frac{1}{2} \ln \frac{(1-\epsilon_t)}{\epsilon_t}$.
    Update the distribution $D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha y_i f_t(\mathbf{d}_i))}{Z_t}$, where Z_t is a normalization factor.
    $t \leftarrow t + 1$.
end while
Return: Ranking function $F(\mathbf{d}) = \sum_{t=1}^{T} \alpha_t \cdot f_t(\mathbf{d})$.

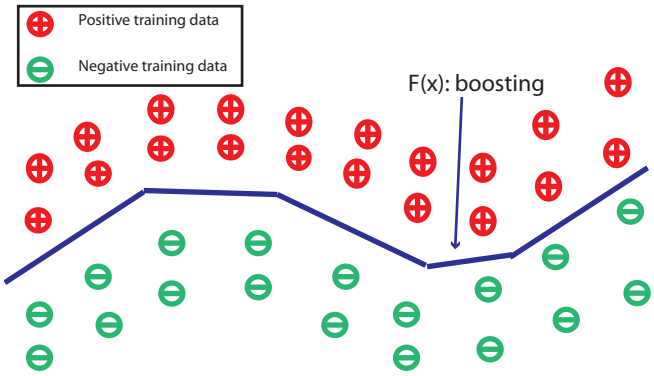


Figure 7.7: Illustration of training data separation by a boosting classifier.

ranking list is sorted by $F(\mathbf{d})$.

In contrast to the averaging function, the discriminative function trains a weighted vector to rank the dataset images, instead of treating each query instance as the same. This helps to overcome the sensitiveness of averaging ranking function, as examples illustrated in Figures 7.3 and 7.4. More specifically, when $M = 1$, our group-query method degenerates to a single query method similar to the DQE method in [9], but with an additional boosting step as described in Section 7.3.2. We use the same query instances in Figures 7.8 and 7.9, but rank dataset images as their distance to a weighted vector trained by a linear-SVM. As shown in Figures 7.8 and 7.9, the top ranked results contain both images of *All souls* and *Ashmolean*.

As shown in Figure 7.1, a group-query helps to overcome the limitation of

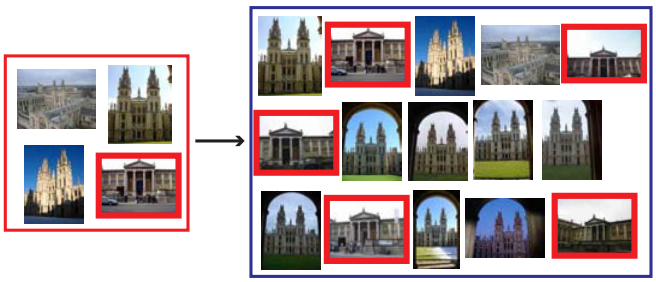


Figure 7.8: Retrieval results of noisy group query. The group-query retains one instance that is different from others (indicated in red). The retrieval results are calculated by discriminative group-query. The positive training samples are these four query instance, while the negative training samples are their bottom ranked results.

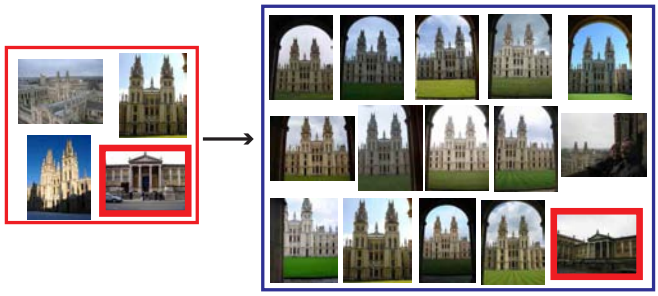


Figure 7.9: Retrieval results of noisy group query. The group-query is the same as used in Figure 7.8. The retrieval results are calculated by discriminative positive training data. The positive training samples are top verified results returned by these four query instance, while the negative training samples are their bottom ranked results.

individual query. For example, *All souls* and *Radcliffe camera* from the Oxford dataset have average precision (AP) scores 0.96 and 0.97 by using the linear-SVM, respectively. The boosting-like ranking function attains slightly higher results, with AP scores being 0.98 and 0.97 on the same clear landmarks. The boosting-like framework is useful in dealing with some lower quality data, *i.e.* objects are unclear in the query instance. For example, the retrieval performance with the linear SVM ranking function degrades greatly in the cases of lower quality object landmarks, *e.g., Magdalen* and *Keble* with the AP score being 0.288 and 0.692, respectively. In contrast, using boosting enables the retrieval accuracy (mAP) to reach 0.407 and 0.870 on *Magdalen* and *Keble*. For an intuitive understanding, Figure 7.10 shows the top ranked results of using two different ranking functions with respect to *Magdalen*. As is seen in Figure 7.10, the top ranked results of our boosting-like ranking function is better than those of the linear SVM ranking

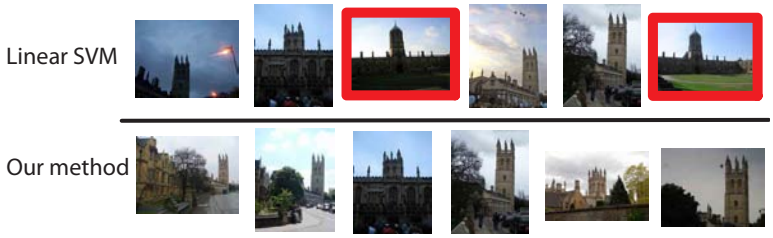


Figure 7.10: Top-6 retrieved results of using the linear SVM ranking function and our boosting-like ranking function with respect to the object landmark *Magdalen* (illustrated in Figure 2.7). The highlighted images correspond to false positive samples.

function (containing a few false positive samples) because query is poor quality.

7.4 Experiments

The retrieval experiments are conducted on three public object retrieval datasets: two small scale datasets (Oxford 5K and Paris 6K) and a large scale dataset Oxford 105K.

7.4.1 Experimental setup

The training sample collection builds sets of positive examples (spatial verified results) and negative examples (from the bottom of $\mathscr{R}$ with lowest nonzero similarity scores). In each boosting iteration, we randomly select (up to) 50 and 200 tf-idf vectors as positive and negative examples, respectively. The linear SVM classifier is trained using the LIBSVM tool [24] with $C = 1$ as in [9]. The maximum iteration number T in boosting learning is set to 20. The run time for a single linear SVM is 0.28s.

Group query instances are provided online. These can be any images of a particular object. In our experiments, we utilize annotated image collections to organize query instances $\mathscr{Q}$ which are varied but contain the same object landmark. The annotation of the Oxford and Paris datasets [4] has been provided. Both of the datasets contain 11 building landmarks for evaluation, with four kinds of visual condition (as described in Section 2.3): "good", "ok", "junk" or "unseen". The query instances are organized as follows: i) we randomly select the same number of images from "good" and "ok" collection as **high quality query** to

Landmark	$M=4$	$M=1$	Landmark	$M=4$	$M=1$
All souls	0.980	0.746	Hertford	0.895	0.801
Ashmolean	0.805	0.418	Keble	0.870	0.844
Balliol	0.867	0.200	Magdalen	0.407	0.439
Bodleian	0.799	0.774	Pitt river	0.852	0.974
Chri. chur.	0.864	0.762	Radc. came.	0.965	0.326
Cornmarket	0.711	0.657			

Table 7.1: Comparison of (maximum) individual and group-query retrieval performance on the Oxford 5K dataset.

evaluate the effect of group-query (Table 7.1-7.4 and Figure 7.12). ii) similarly, images from "ok" and "junk" are treated as **low quality query** to evaluate the discriminative ranking function (Table 7.2 and Figure 7.13). iii) To compare with the state-of-the-art methods, we use the 55 queries with bounding boxes defined in [4] (Table 7.5).

7.4.2 Experimental results

We illustrate the effects of our group-query method as follows:

Effects of using group-query Table 7.1 compares the mAP score obtained for each landmark on the Oxford 5K dataset, using individual query images ($M=1$) and groups of 4 query images ($M=4$). The individual query results are obtained by running each of the 4 queries in the group separately, and storing the maximum result. The query groups are sampled using the "high quality" strategy. It is clear from Table 7.1 that the group-query improves retrieval performance for 9 out of 11 queries, and by an average of 29.9%. Note that our method fails in the cases of (*Magdalen, Pitt river*) due to a lack of quality positive samples. Figure 7.11 illustrates that the group-query can result in significantly higher precision-recall performance than any individual query, using the object landmark.

Evaluation of query instance number M We investigate the effects of varying the numbers of query instances. Firstly, Figure 7.12 investigates the effects of increasing number of instance in high quality query. As seen in Figure 7.12, the retrieval accuracy increases as M enlarges and plateaus when $M=10$. This indicates that our method only needs a small number of high quality query

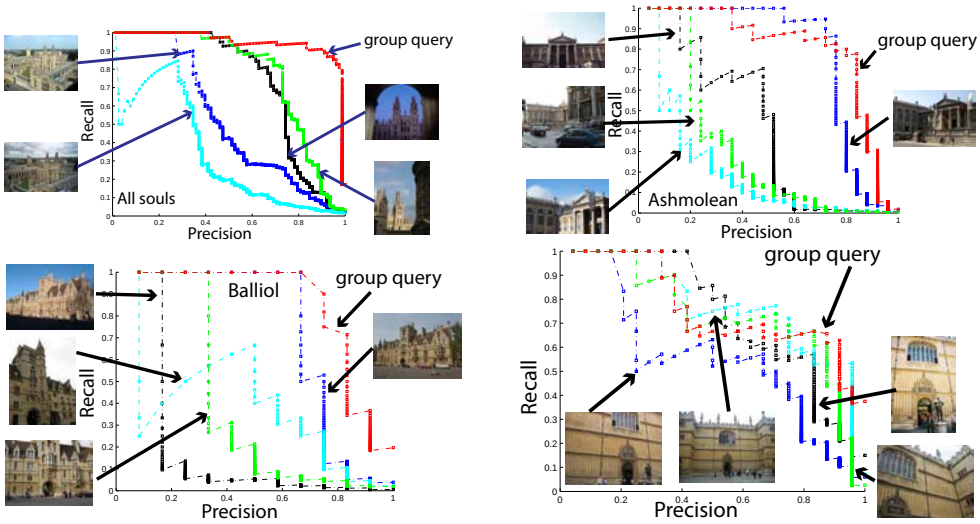


Figure 7.11: Precision-recall (PR) curves of individual query *v.s.* group- query. Group query obtains higher retrieval accuracy than any individual query.

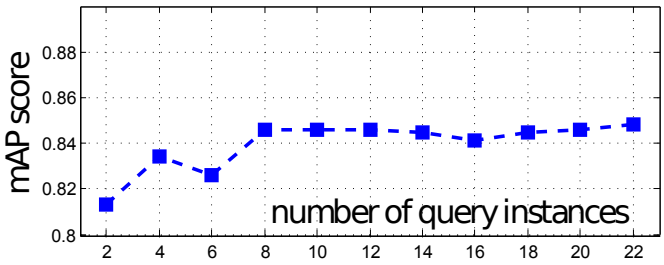


Figure 7.12: Retrieval performance with different numbers of high query quality instances.

instances. As M increases, the computational cost increases linearly, due to the repetition of the dot product ranking and spatial verification for each query instance. In order to balance effectiveness and efficiency, M is set to 4 high quality query in the experiments below. Similarly, Figure 7.13 investigates the retrieval accuracy with increasing number of low quality query images. The retrieval accuracy fluctuates as M enlarges but plateaus 0.8 when M is greater than 6.

Investigation of using linear SVM *v.s.* Adaboost-LinearSVM Table 7.2 compares the retrieval performance using two types of classifiers used in the discriminative ranking function: the linear SVM (referred to as R1) and the *AdaBoost-LinearSVM* (referred to as R2). The comparison experiment is conducted on two types of group query, that is, retrieval with high quality query instances

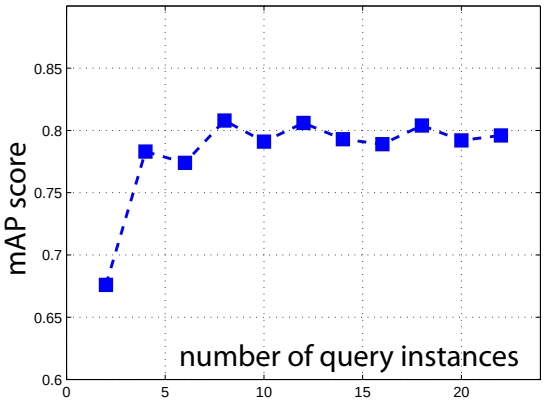


Figure 7.13: Retrieval performance with different numbers of low quality query instances.

(“good" and “ok" images in Groups A, B and C) and retrieval with low quality query instances (“ok" and “junk" images in Groups C, D and E). As is seen in Table 7.2, the discriminative ranking function with *AdaBoost-LinearSVM* always performs better than linear SVM. The superiority of the boosting-like ranking function is more evident in the low quality query. For example, in Group E the retrieval performance using boosting-like ranking function (R2) is 3.7% (5.9%) higher on the Oxford 5K (Paris 6K) datasets than the results using linear SVM ranking function (R1). Moreover, Table 7.3 illustrates the re-ranking CPU time of different discriminative ranking functions. From Table 7.3, we see that our method achieves the best retrieval accuracy with high efficiency. By parallelising SVM training (Section 7.3.2), we reduce the re-ranking time at a slight cost to mAP because the linear-SVM classifiers are pre-trained in a pool.

Comparison with query expansion Similar to our method, average query expansion (AQE) [37] also applies spatial verification to top ranked retrieval results and collect a number of true positives. However, it uses query averaging of the visual words collected in the positive images, while our method trains a discriminative ranking function with the same positive images and additional negative images, as discussed in Section 7.2. Table 7.4 compares our group-query method with average query expansion (AQE) [37] in the following aspects: i) the best (maximum mAP socre) of AQE for each individual query. ii) AQE on all positive examples collected by the group-query. As is seen in Table 7.4, our group-query method can outperform AQE in different experimental configurations, even

	Method	M	Oxford 5K	Paris 6K
A	R1	2	0.789	0.772
high	**R2**	2	0.813	0.780
B	R1	4	0.809	0.857
high	**R2**	4	0.834	0.871
C	R1	10	0.833	0.864
high	**R2**	10	0.846	0.875
D	R1	2	0.668	0.701
low	**R2**	2	0.676	0.714
E	R1	4	0.755	0.682
low	**R2**	4	0.783	0.722
F	R1	10	0.756	0.791
low	**R2**	10	0.791	0.824

Table 7.2: Retrieval performance with different discriminative ranking functions: R1: group-query with linear SVM [9], R2: group-query with boosting. M denotes the maximum number of queries in $\mathcal{Q}$. Both **high** and **low** quality queries are tested, as indicated in the table. The number of query instances is up to M.

Method	mAP	re-ranking CPU time (s)
linear SVM	0.809	0.37
Adaboost-linearSVM	0.834	1.12
RBF-kernel SVM	0.815	12.26
Adaboost-linearSVM (parallelised)	0.830	0.49

Table 7.3: Average re-ranking CPU time for different classifiers used in the discriminative ranking function: linear SVM, Adaboost-linearSVM, and non-linear SVM (RBF-kernel). The group-query ($M = 4$) is conducted on the Oxford 5K dataset.

Method	A	B	Oxford 5K	Paris 6K
tf-idf + dot product (maximum)		4	0.631	0.605
AQE (maximum)	√	4	0.742	0.657
AQE (positive examples)	√	4	0.743	0.809
Group-query	√	4	0.834	0.871

Table 7.4: Retrieval results comparison between group-query and query expansion methods, applied to both individual and group queries. A denotes use of spatial verification. B denotes the maximum number of queries in $\mathcal{Q}$.

when given the same images.

Comparison with the state-of-the-art methods We compare our method with several state-of-the-art methods, as listed in Table 7.5 Group B focuses on query model refinement with post-processes. The instances $\mathcal{Q}$ used for group-query

	Methods	Oxford 5K	Paris 6K	Oxford 105K
	Baseline [106]	0.612	0.639	0.515
A	Visual word re-weighting (Chapter 4)	0.660	0.674	0.598
	Descriptor learning (non-linear) [108]	0.662 [108]	0.678 [108]	0.541 [108]
	Soft-assignment [107]	0.673 [107]	0.660	N/A
	Geometry-Preserving [159]	0.696 [159]	N/A	0.604 [159]
	Total association (Chapter 4)	0.700	0.682	0.680
	Spatial expansion ($\mathbf{F}'_{15}$, Chapter 3)	0.701	0.683	0.667
	Cross word (Chapter 4)	0.712	0.722	0.604
	Fine vocabulary [96]	0.742 [96]	0.749 [96]	N/A
	AUG [142]	0.776 [9]	N/A	0.711 [9]
	SPAUG [9]	0.785 [9]	N/A	0.723 [9]
B	Spatial verification [106]	0.649	0.655	0.571
	Ranking verification (Chapter 6)	0.654	0.652	0.595
	Context based re-ranking (Chapter 5)	0.701	0.700	0.585
	QE Baseline [37]	0.708	0.736	0.679
	iSP [34]	0.741 [34]	0.769 [34]	0.649 [34]
	Local geometry [105]	0.788 [105]	0.634 [105]	0.725 [105]
	DQE [9]	0.798	0.783	0.809
	AQE [37]	0.806	0.769	0.767
	Hello neighbors [114]	0.814 [114]	0.803 [114]	0.767 [114]
	DQE + Boosting (single query)	0.823	0.782	0.818
	Total recall II [34]	0.827 [34]	0.805 [34]	0.767 [34]
	DQE + Boosting (Group-query)	0.896	0.856	0.890
	DQE + Boosting (Group-query with ground truth)	0.901	0.852	0.890
C	Context based re-ranking (Chapter 5)	0.701	0.700	0.585
	Total association+ Context based re-ranking	0.704	0.711	0.636
	Cross word+ Context based re-ranking	0.735	0.720	0.648
	Ranking verification (Chapter 6)	0.654	0.652	0.595
	Total association+ Ranking verification	0.708	0.687	0.682
	Cross word+ Ranking verification	0.738	0.720	0.658
	Group-query	0.901	0.852	0.890
	Total association+ **Group-query**	0.904	0.852	0.893
	Cross word+ **Group-query**	0.909	0.856	0.901

Table 7.5: Comparison of group-query to the state-of-the-art methods. Group A: retrieval results of methods that modify the baseline before the query is executed (pre-process). Group B: retrieval results of methods that modify the baseline after the query is executed (post-process). Group C: combination of pre-process and post-process. In this group, Group-query refers to DQE + Boosting (Group-query with ground truth) in Group B. Note that we cite the retrieval results of AUG [142] from literature [9].

in Table 7.5 Group B are high quality query ($M = 1$) for "DQE + Boosting (single query)" and high quality query ($M = 4$) for "DQE + Boosting (Group-query)", respectively. The instances $\mathcal{Q}$ are the 5 ground truth files defined in [4] of each landmark for " DQE + Boosting (Group-query with ground truth)". As seen in Group B, all group query methods can outperform the state-of-the-art. In particular, when only a single query instance is available, our method "DQE+Boosting (single query)" can outperform DQE [9] (using linear SVM ranking function instead). In cases where multiple query instances are available, boosting group retrieval with a small number of query instances can significantly improve the retrieval results. In Table 7.5 Group C, "Group-query" refers to DQE+Boosting with 5 ground truth files of each landmark in Oxford 5K and

Figure 7.14: Illustration of group-query retrieval results, with $M = 4$ high quality query instance, corresponding to the precision-recall curves shown in Figure 7.11.

Paris 6K datasets. Combining with total association and cross-word matching can further improve the retrieval results. However, the performance gain is small because the retrieval results are trained weights according to positive (negative) samples. The positive samples are collected by spatial verification, whose information has already partly been exploited by an object-based thesaurus used in total association and cross-word matching. Finally, we show retrieval examples by high and low quality query in Figure 7.14 and Figure 7.15, respectively. Retrieval

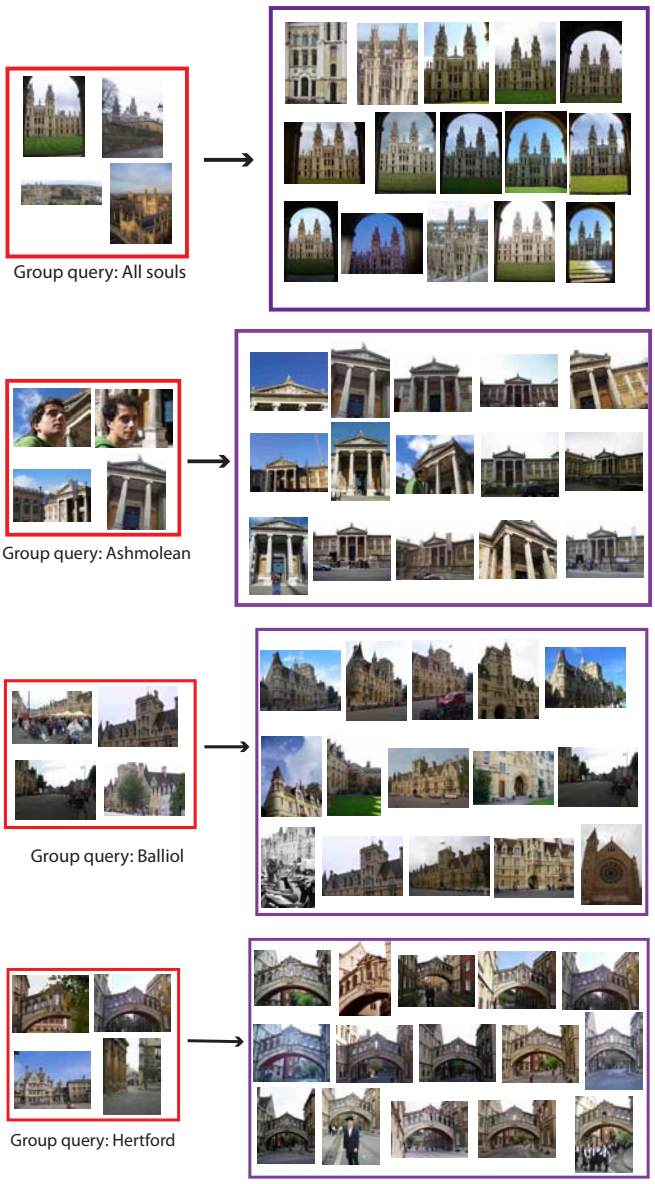


Figure 7.15: Illustration of query retrieval results, with $M = 4$ low quality query instances used as group-query.

with high quality query images can lead to accurate retrieval performance. The performance is slightly affected if the quality of query images become low, but not as badly as a single low quality query.

Dataset	QE Baseline [37]	AQE [37]			DQE [9]		**Group-query**	
	R0	R1	R4	iSP [34]	R1	R4	R1	R4
Oxford 5K	0.708	0.806	0.764	0.825 [34]	0.798	0.727	0.901	0.760
Paris 6K	0.736	0.769	0.755	0.722 [34]	0.783	0.776	0.852	0.871
Caltech Categories	0.497	0.447	0.519	N/A	0.313	0.520	0.333	0.720
Oxford 105K	0.679	0.767	0.742	0.761 [34]	0.809	0.745	0.890	0.840

Table 7.6: Retrieval performance of group-query with ranking verification, and other query expansion methods. The ranking verification methods $R1$ and $R4$ are consistent to Table 6.1.

7.4.3 Group-query with ranking verification

In this section, we show that group-query can also work with verification results returned by ranking verification proposed in Chapter 6. Similar to spatial verification, ranking verification is able to determine a set of reliable true positives. However, it does not require geometric information from raw features and thus is not restricted to rigid objects, *e.g.* Oxford buildings.

Table 7.6 reports the results of group-query combined with ranking verification. We use the 5 queries provided by [4] for each landmark as an object. Similar to the results of DQE (R1 and R4), ranking verification (R4) works effectively when the objects in target images can not be detected by spatial verification (R1). This is more evident in the Caltech Categories, where the various methods, including AQE, DQE and group-query with spatial verification (R1), perform worse than the QE baseline. These results illustrate that our ranking verification is an alternative verification method, that is especially useful when the target objects are not rigid among dataset images.

7.5 Conclusion

In this chapter, we have introduced the notion of a group-query, and shown its effectiveness for object retrieval. The group-query can be input manually, or automatically gathered from a single query image. We proposed a boosted discriminative ranking function to refine the group-query model. The proposed ranking function captures nonlinear discriminative information on the retrieved data samples effectively and efficiently. Experimental results show that our method can achieve higher performance than competing methods.

Moreover, our ranking verification method can operate as an alternative to

spatial verification to generate reliable positive and negative training data. The experimental results show that our method is more flexible than spatial verification on various datasets at little extra cost.

CONCLUSION

CHAPTER VIII

This section summarizes the contributions of this thesis, and discusses several directions of future research for object retrieval.

8.1 Contributions

Throughout this thesis we have developed a number of methods to improve the retrieval performance of a BoW retrieval system. It has been shown that there are three main issues affecting the retrieval performance: i) image representation; ii) image similarity measure and iii) re-ranking retrieval results. The methods in this thesis have addressed all these issues. Their performance, together with the state-of-the-art methods, are summarized in Table 8.1.

In Chapter 3, we propose a visual thesaurus structure to store spatial relatedness of visual words and a spatial expansion method to improve the image representation of query. There are two ways to organize these visual words: based on a general thesaurus (**F**) or an object-based thesaurus (**F′**). We prefer object-based thesaurus (**F′**) in object retrieval because it only considers foreground words $\mathbf{W}_S$, thus avoids background information.

In Chapter 4, we focus on issue ii). We improve the standard tf-idf weight scheme such that foreground words will be weighted heavily. An association of spatial expansion and re-weighting is then proposed to consider both relatedness and importance of visual words. We also present a cross-word matching similarity between a pair of query/dataset images, such that matching features can contribute to the similarity even if they are mapped to different visual words. Because a learning stage is required, we have noticed that there is some limitation of the retrieval performance improvement in the methods proposed in Chapters 3 and 4. The object-based thesaurus mostly works effectively on retrieving rigid objects, as it relies on spatial transform to detected foreground.

From Chapter 5, we pay attention to issue *iii*) in the retrieval system. Our goal is to refine the initial retrieval results only with the ranking information. Chapter 5 presents a re-ranking method that considers contextual ranking information, such that relevant images are ranked based on other images in a context. Chapter 6 proposes a ranking consistency examination, leading to a more robust retrieval performance. Both methods are also easily integrated into various retrieval-related applications. Compared to other post-processing methods (Table 8.1, Group B), our methods presented in Chapters 5 and 6 rely neither on geometric information of raw features nor on a re-querying process as used in query expansion. Instead, these re-ranking methods only take account of ranking information returned by the initial retrieval results. Therefore, our re-ranking methods are more flexible on various image datasets.

In Chapter 7, we define the image retrieval based on a group-query, and rank the dataset images by trained weights. We construct a nonlinear ranking model using an ensemble of linear SVM that are adaptively weighted by boosting. This outperforms other state-of-the-art methods on the standard retrieval datasets.

8.2 Summary

The full comparison of the state-of-the-art methods to our proposed methods is listed in Table 8.1.

As seen in Table 8.1 Group A, the retrieval accuracy (without re-ranking) on Oxford 5K dataset has increased incrementally in recent years, *i.e.* from 0.612 (Philbin *et al.* [106], 2007) to 0.785 (Arandjelović and Zisserman [9], 2012). These methods take account of increasing information about the dataset images. The retrieval accuracy benefits from the enriched query model. Visual word re-weighting (Chapter 4) and descriptor learning [108] are two methods that do not need to expand the query words. Thus, their retrieval accuracy is relatively low, compared to other methods in Group A. The expansion of query words, *e.g.* soft-assignment [107], spatial expansion ($\mathbf{F}_5$ and $\mathbf{F}\prime_{15}$, Chapter 3) and total association (Chapter 4), achieves about 10% increase in accuracy compared to the baseline. However, the expansion costs extra computational time during query. Other methods exploit the feature relatedness and use them to improve the similarity measures, *e.g.* geometry-preserving [159], fine vocabulary [96], cross-

	Methods	Oxford 5K	Paris 6K	Oxford 105K
	Baseline [106]	0.612	0.639	0.515
A	**Visual word re-weighting** (Chapter 4)	0.660	0.674	0.598
	Descriptor learning (non-linear) [108]	0.662 [108]	0.678 [108]	0.541 [108]
	Soft-assignment [107]	0.673 [107]	0.660	N/A
	Spatial expansion ($\mathbf{F}_5$, Chapter 3)	0.685	0.679	0.622
	Geometry-Preserving [159]	0.696 [159]	N/A	0.604 [159]
	Total association (Chapter 4)	0.700	0.682	0.680
	Spatial expansion ($\mathbf{F}\prime_{15}$, Chapter 3)	0.701	0.683	0.667
	Cross word (Chapter 4)	0.712	0.722	0.604
	Fine vocabulary [96]	0.742 [96]	0.749 [96]	N/A
	AUG [142]	0.776 [9]	N/A	0.711 [9]
	SPAUG [9]	0.785 [9]	N/A	0.723 [9]
B	Spatial verification [106]	0.649	0.655	0.571
	Ranking verification (Chapter 6)	0.654	0.652	0.595
	Context based re-ranking (Chapter 5)	0.701	0.700	0.585
	QE Baseline [37]	0.708	0.736	0.679
	iSP [34]	0.741 [34]	0.769 [34]	0.649 [34]
	Local geometry [105]	0.788 [105]	0.634 [105]	0.725 [105]
	AQE [37]	0.806	0.769	0.767
	DQE [9]	0.798	0.783	0.809
	Hello neighbors [114]	0.814 [114]	0.803 [114]	0.767 [114]
	DQE + Boosting (Chapter 7)	0.823	0.782	0.818
	Total recall II [34]	0.827 [34]	0.805 [34]	0.767 [34]
	DQE + Boosting (group) (Chapter 7)	0.896	0.856	0.890

Table 8.1: Retrieval results summary. Group A: retrieval results of methods that modify the baseline before the query is executed (pre-process). Group B: retrieval results of methods that modify the baseline after the query is executed (post-process). Note that we cite the retrieval results of AUG [142] from literature [9]. Our methods presented in this thesis are highlighted.

word (Chapter 4), AUG [142] and SPAUG [9]. These methods are more efficient than expansion of query words, especially AUG [142] and SPAUG [9], and can outperform other methods in Group A. However, their effects highly rely on the feature information collected offline and are limited to rigid objects. In summary, the effects of methods in Group A depend on the amount of embedded information recovered from dataset.

Table 8.1 Group B shows that a result re-ranking process is useful in improvement of retrieval accuracy. Most methods need a spatial consistency examination, *e.g.* spatial verification [106, 105, 34], query expansion [37, 9, 114, 34] and group-query (Chapter 7). Therefore, the retrieval performance mostly benefits from information collected during spatial consistency examination rather than the original query model. Again, these methods have limited effects, as

Methods		Oxford 5K	Paris 6K	Oxford 105K
Baseline [106]		0.612	0.639	0.515
C	Spatial expansion ($\mathbf{F}_5$, Chapter 3)	0.685	0.679	0.622
	Spatial expansion+ Spatial verification	0.716	0.689	0.676
	Spatial expansion+ AQE	0.815	0.778	0.773
	Spatial expansion+ DQE	0.810	0.785	0.798
	Spatial expansion ($\mathbf{F}'_{15}$, Chapter 3)	0.701	0.683	0.667
	Spatial expansion+ Spatial verification	0.719	0.689	0.704
	Spatial expansion+ AQE	0.806	0.785	0.783
	Spatial expansion+ DQE	0.813	0.789	0.818
	Visual word re-weighting (Chapter 4)	0.660	0.674	0.598
	Visual word re-weighting+ Spatial verification	0.677	0.684	0.611
	Visual word re-weighting+ AQE	0.801	0.777	0.781
	Visual word re-weighting+ DQE	0.811	0.782	0.787
	Total association (Chapter 4)	0.700	0.682	0.680
	Total association+ Spatial verification	0.710	0.690	0.706
	Total association+ AQE	0.804	0.785	0.774
	Total association+ DQE	0.816	0.790	0.817
	Cross word (Chapter 4)	0.712	0.722	0.604
	Cross word+ Spatial verification	0.723	0.726	0.647
	Cross word+ AQE	0.821	0.787	0.765
	Cross word+ DQE	0.828	0.793	0.797
D	Context based re-ranking (Chapter 5)	0.701	0.700	0.585
	Total association+ Context based re-ranking	0.704	0.711	0.636
	Cross word+ Context based re-ranking	0.735	0.720	0.648
	Ranking verification (Chapter 6)	0.654	0.652	0.595
	Total association+ Ranking verification	0.708	0.687	0.682
	Cross word+ Ranking verification	0.738	0.720	0.658
	Group-query (Chapter 7)	0.901	0.852	0.890
	Total association+ Group-query	0.904	0.852	0.893
	Cross word+ Group-query	0.909	0.856	0.901

Table 8.2: Summary of retrieval results combination.

they mostly work on rigid object retrieval. Our methods, ranking verification (Chapter 6) and context based re-ranking (Chapter 5), only need to know the ranking information of initial retrieval results, thus are more flexible. Both of them can achieve close re-ranking accuracy with spatial verification. They can work with a number of pre-processing methods as listed in Table 8.2 Group D. Furthermore, ranking verification can work with various query expansion methods, as illustrated in Chapter 6, without a spatial consistency examination.

Table 8.2 Group C illustrates the effects of combination of our proposed methods and query expansion (AQE and DQE). As seen from Group C, the performance gain is small compared to results using query expansion individually. This is because an object-based thesaurus (required by total association and cross-word matching) already exploits spatial consistency between features offline. The online query expansion, which also needs a spatial consistency examination, therefore has limited effects.

Table 8.2 Group D reports the retrieval results of various improvement

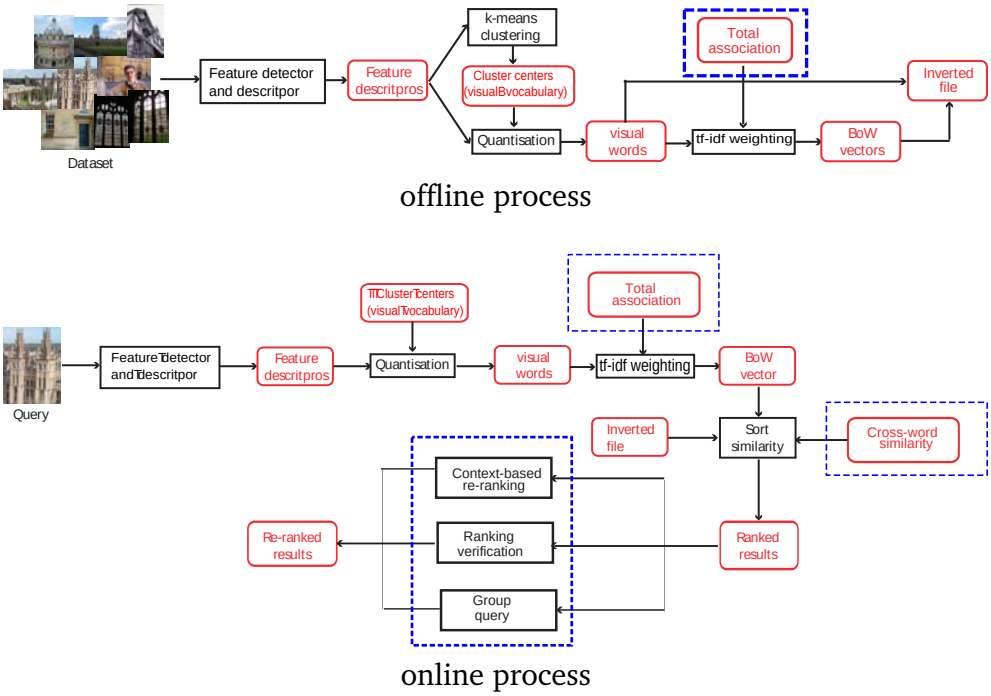


Figure 8.1: Summary of our proposed methods in the BoW based retrieval system. The proposed methods are indicated in dashed boxes.

methods discussed in this thesis. We combine three kinds of re-ranking methods proposed in this thesis with total association and cross-word similarity methods, respectively. As shown in Figure 8.1, these method are placed in different steps in the retrieval pipeline. Therefore, we combine methods that are not located in the same step in Group D. The retrieval accuracy reported in Group D is varied according to the type of combination. As seen in Group D and discussed in previous chapters, the effectiveness of the combination depends on the amount of information provided by each component methods. For example, total association (cross-word similarity) and group query gets little to improve because spatial consistency information is used in both components.

8.3 Future work

The BoW retrieval system has attracted interest in many related areas: computer vision, information retrieval, and machine learning in recent years. However, it is still far from a solved problem.

Adaptive visual vocabulary The most computationally expensive module in a BoW retrieval system is to build a discriminative vocabulary. Current methods, *e.g.* AKM and HKM, are approximate in measuring the feature distance. Therefore, they can not capture the diversity and richness of high dimensional descriptors. Also, the vocabulary building is not adaptive when the dataset images are updated. The proper solution is to make the visual vocabulary **distributed, dynamic and adaptively transferred**. Recent work [10] has proposed a vocabulary adaptation method based on VLAD descriptors. It aims to address the problem of vocabulary sensitivity, such that the visual vocabulary trained in one dataset can be used to represent another dataset. The vocabulary adaptation is also similar to the problem of transfer learning [101]. Transfer learning is useful when training and test data is drawn from different feature spaces. As a result, we can also deem vocabulary adaptation as a transfer learning problem: the vocabulary is built on images different from those need to be searched. However, it needs to carefully scale the transfer learning methods to a large scale vocabulary.

Convergence of retrieval and classification The difference between object retrieval and classification has shrunk in recent years. Typically, the goal of object retrieval is to, given a query image depicting an object, return a list of images containing that same object. It usually requires fast search from large scale dataset, but with little focus on learning the dataset images. Object classification aims to characterize objects with some prior defined classes (labels). It usually needs to learn a classifier with expensive computational cost, and depends heavily on machine learning methods. Recent years have seen convergence of these two problems. Some object retrieval techniques learn a classifier for querying images [9, 30]. Similar to the adaptive visual vocabulary problem, classification techniques used in object retrieval also need to be scale to large dataset, *e.g.* a linear-SVM classier [9] or boosting framework [30]. Conversely, object classification problem recently also pays attention to scalability and efficiency because the availability of large scale dataset, *e.g.* ImageNet. The convergence (divergence) of these two kinds of method will depend on the scalability of these techniques such that object retrieval can borrow them from classification, and vice versa.

Bibliography

[1] http://press.liacs.nl/mirflickr/dlform.php. pages 30

[2] http://wordnet.princeton.edu/. pages 44

[3] http://www.image-net.org/index. pages 30

[4] http://www.robots.ox.ac.uk/∼vgg/data/. pages 29, 30, 105, 169, 170, 174, 177

[5] http://www.robots.ox.ac.uk/∼vgg/software/fastcluster/. pages 33

[6] http://www.vision.caltech.edu/html-files/archive.html. pages 30

[7] AGARWAL, S., SNAVELY, N., SIMON, I., SEITZ, S., AND SZELISKI, R. Building rome in a day. In *Proc. IEEE Int. Conf. Comp. Vis.* (2009), pp. 72–79. pages 38, 49, 100

[8] ARANDJELOVIĆ, R., AND ZISSERMAN, A. Multiple queries for large scale specific object retrieval. In *Proc. Brit. Mach. Vis. Conf.* (2012). pages 160

[9] ARANDJELOVIĆ, R., AND ZISSERMAN, A. Three things everyone should know to improve object retrieval. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (2012). pages 42, 47, 61, 62, 69, 93, 112, 114, 129, 131, 151, 152, 153, 154, 160, 162, 163, 165, 166, 167, 169, 173, 174, 177, 180, 181, 184

[10] ARANDJELOVIĆ, R., AND ZISSERMAN, A. All about VLAD. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (2013). pages 38, 184

[11] BABENKO, B., DOLLÁR, P., AND BELONGIE, S. Task specific local region matching. In *Proc. IEEE Int. Conf. Comp. Vis.* (2007), IEEE, pp. 1–8. pages 36

[12] BARTOLINI, I., AND ROMANI, C. Efficient and effective similarity-based video retrieval. In *Int. Conf. on Similarity Search and Applications* (2010), ACM, pp. 133–134. pages 43

[13] BAUMBERG, A. Reliable feature matching across widely separated views. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (2000), vol. 1, IEEE, pp. 774–781. pages 15

[14] BAY, H., ESS, A., TUYTELAARS, T., AND VAN GOOL, L. Speeded-up robust features (surf). *Comp. Vis. Image Understanding 110*, 3 (2008), 346–359. pages 15

[15] BODEN, M. A. *Mind as machine: A history of cognitive science*, vol. 1. Clarendon Press, 2006. pages 1

[16] BOSCH, A., ZISSERMAN, A., AND MUOZ, X. Image classification using random forests and ferns. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (2007), pp. 1–8. pages 39

[17] BOWMAN, A. W., AND AZZALINI, A. Applied smoothing techniques for data analysis. pages 123

[18] BRADSKI, G. R. Real time face and object tracking as a component of a perceptual user interface. In *Workshop on Applications of Computer Vision.* (1998), IEEE, pp. 214–219. pages 13

[19] BRODER, A. On the resemblance and containment of documents. In *Compression and Complexity of Sequences* (1997), IEEE, pp. 21–29. pages 45, 141

[20] BROWN, M., AND LOWE, D. G. Recognising panoramas. In *Proc. IEEE Int. Conf. Comp. Vis.* (2003), vol. 2, p. 5. pages 15

[21] BUCKLEY, C., SALTON, G., ALLAN, J., AND SINGHAL, A. Automatic query expansion using smart: Trec 3. *NIST SPECIAL PUBLICATION SP* (1995), 69–69. pages 40

[22] CAO, Y., WANG, C., LI, Z., ZHANG, L., AND ZHANG, L. Spatial-bag-of-features. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (2010), pp. 3352–3359. pages 40

[23] CARSON, C., BELONGIE, S., GREENSPAN, H., AND MALIK, J. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Trans. Pattern Anal. Mach. Intell. 24*, 8 (2002), 1026–1038. pages 14

[24] CHANG, C.-C., AND LIN, C.-J. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology 2* (2011), 27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`. pages 169

[25] CHATFIELD, K., LEMPITSKY, V., VEDALDI, A., AND ZISSERMAN, A. The devil is in the details: an evaluation of recent feature encoding methods. In *Proc. Brit. Mach. Vis. Conf.* (2011). pages 35

[26] CHEN, Y., DICK, A., AND LI, X. Visual distance measures for object retrieval. In *Digital Image Computing Techniques and Applicationss* (2012), IEEE, pp. 1–8. pages 5, 48

[27] CHEN, Y., DICK, A., LI, X., AND VAN DEN HENGLE, A. Spatiall aware feature selection and re-weighting. *Image and Vis. Comp. under revision.*. pages 5

[28] CHEN, Y., DICK, A., AND VAN DEN HENGEL, A. Image Retrieval with a Visual Thesaurus. In *Int. Conf. on Digital Image Computing: Techniques and Applications* (2010), pp. 8–14. pages 5

[29] CHEN, Y., LI, X., DICK, A., AND HILL, R. Ranking consistency for image matching and object retrieval. *Pattern Recogn., under revision*. pages 5

[30] CHEN, Y., LI, X., DICK, A., AND VAN DEN HENGLE, A. Boosting object retrieval with group queries. *IEEE Signal Processing Letters 19* (2012), 765–768. pages 6, 160, 184

[31] CHO, M., AND LEE, K. Mode-seeking on graphs via random walks. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (2012), pp. 606–613. pages 48

[32] CHO, M., AND MULEE, K. Authority-shift clustering: Hierarchical clustering by authority seeking on graphs. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (2010), pp. 3193–3200. pages 155

[33] CHUM, O., AND MATAS, J. Large-scale discovery of spatially related images. *IEEE Trans. Pattern Anal. Mach. Intell. 32*, 2 (2010), 371–377. pages 48

[34] CHUM, O., MIKULIK, A., PERDOCH, M., AND MATAS, J. Total recall ii: Query expansion revisited. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (2011). pages 43, 47, 62, 64, 69, 93, 114, 129, 131, 151, 154, 159, 174, 177, 181

[35] CHUM, O., PERDOCH, M., AND MATAS, J. Geometric min-hashing: Finding a (thick) needle in a haystack. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (2009), pp. 9–16. pages 46, 135

[36] CHUM, O., PHILBIN, J., ISARD, M., AND ZISSERMAN, A. Scalable near identical image and shot detection. In *Proc. Int. Conf. on Image and Video Retrieval* (2007), vol. 9, pp. 549–556. pages 45, 46, 141, 145, 148

[37] CHUM, O., PHILBIN, J., SIVIC, J., ISARD, M., AND ZISSERMAN, A. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (2007), pp. 1–8. pages 34, 41, 47, 55, 62, 64, 69, 88, 89, 90, 92, 93, 114, 122, 127, 129, 131, 135, 151, 152, 153, 154, 159, 160, 163, 172, 174, 177, 181

[38] CHUM, O., PHILBIN, J., AND ZISSERMAN, A. Near duplicate image detection: min-hash and tf-idf weighting. In *Proc. Brit. Mach. Vis. Conf.* (2008), vol. 3, p. 4. pages 45, 46, 122, 135, 141, 145, 148, 149

[39] CIOCCA, G., AND SCHETTINI, R. Content-based similarity retrieval of trademarks using relevance feedback. *Pattern Recogn. 34*, 8 (2001), 1639–1655. pages 136

[40] CSURKA, G., DANCE, C., FAN, L., WILLAMOWSKI, J., AND BRAY, C. Visual categorization with bags of keypoints. In *ECCV Workshop on statistical learning in computer vision* (2004). pages 15, 20

[41] DATAR, M., IMMORLICA, N., INDYK, P., AND MIRROKNI, V. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry* (2004), pp. 253–262. pages 44, 45

[42] FALOUTSOS, C., BARBER, R., FLICKNER, M., HAFNER, J., NIBLACK, W., PETKOVIC, D., AND EQUITZ, W. Efficient and effective querying by image content. *J. Intel. Inf. Syst. 3*, 3-4 (1994), 231–262. pages 9, 12

[43] FISCHLER, M. A., AND BOLLES, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM 24*, 6 (1981), 381–395. pages 14

[44] FRAHM, J.-M., FITE-GEORGEL, P., GALLUP, D., JOHNSON, T., RAGURAM, R., WU, C., JEN, Y.-H., DUNN, E., CLIPP, B., LAZEBNIK, S., ET AL. Building rome on a cloudless day. In *Proc. Eur. Conf. Comp. Vis.* 2010, pp. 368–381. pages 49

[45] FREUND, Y., AND SCHAPIRE, R. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Computational Learning Theory* (1995), pp. 23–37. pages 166

[46] GAVVES, E., AND SNOEK, C. Landmark image retrieval using visual synonyms. In *Proc. ACM Int. Conf. on Multimedia* (2010), pp. 1123–1126. pages 40

[47] GIACINTO, G., AND ROLI, F. Bayesian relevance feedback for content-based image retrieval. *Pattern Recogn. 37*, 7 (2004), 1499–1508. pages 136

[48] GIONIS, A., INDYK, P., AND MOTWANI, R. Similarity search in high dimensions via hashing. In *Proc. Int. Conf. on Very Large Data Bases* (1999), pp. 518–529. pages 44

[49] GOAD, C. Special purpose automatic programming for 3d model-based vision. *Readings in Computer Vision* (1987), 371–381. pages 14

[50] GRAUMAN, K., AND DARRELL, T. Efficient image matching with distributions of local invariant features. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (2005), vol. 2, pp. 627–634. pages 44

[51] GRIMSON, W. E. L., AND LOZANO-PEREZ, T. Localizing overlapping parts by searching the interpretation tree. *IEEE Trans. Pattern Anal. Mach. Intell.*, 4 (1987), 469–482. pages 14

[52] GUIMARÃES PEDRONETTE, D., AND DA S. TORRES, R. Image re-ranking and rank aggregation based on similarity of ranked lists. In *Computer analysis of images and patterns* (2011), pp. 369–376. pages 136

[53] HARALICK, R. M. Statistical and structural approaches to texture. *Proceedings of the IEEE 67*, 5 (1979), 786–804. pages 13

[54] HARTLEY, R., AND ZISSERMAN, A. *Multiple view geometry in computer vision*. Cambridge university press, 2003. pages 40, 65, 100

[55] HOWARTH, P., AND RÜGER, S. Evaluation of texture features for content-based image retrieval. In *Image and Video Retrieval*. Springer, 2004, pp. 326–334. pages 14

[56] HU, Y., LI, M., AND YU, N. Multiple-instance ranking: Learning to rank images for image retrieval. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (2008), pp. 1–8. pages 136

[57] HUA, G., BROWN, M., AND WINDER, S. Discriminant embedding for local image descriptors. In *Proc. IEEE Int. Conf. Comp. Vis.* (2007), IEEE, pp. 1–8. pages 37

[58] HUTTENLOCHER, D. P., AND ULLMAN, S. Recognizing solid objects by alignment with an image. *Int. J. Comp. Vis. 5*, 2 (1990), 195–212. pages 14

[59] INDYK, P., AND MOTWANI, R. Approximate nearest neighbors: towards removing the curse of dimensionality. In *ACM symposium on Theory of computing* (1998), pp. 604–613. pages 44, 45

[60] JAIN, P., KULIS, B., AND GRAUMAN, K. Fast image search for learned metrics. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (2008), IEEE, pp. 1–8. pages 45

[61] JAYNES, E. Information theory and statistical mechanics. ii. *Physical review 108*, 2 (1957), 171. pages 78

[62] JÉGOU, H., DOUZE, M., AND SCHMID, C. Hamming embedding and weak geometric consistency for large scale image search. In *Proc. Eur. Conf. Comp. Vis.* (2008), pp. 304–317. pages 30, 38, 105

[63] JÉGOU, H., DOUZE, M., AND SCHMID, C. Improving bag-of-features for large scale image search. *Int. J. Comp. Vis. 87*, 3 (2010), 316–336. pages 12, 22, 24, 38, 74, 98, 99, 136

[64] JEGOU, H., DOUZE, M., AND SCHMID, C. Product quantization for nearest neighbor search. *IEEE Trans. Pattern Anal. Mach. Intell. 33*, 1 (2011), 117–128. pages 37

[65] JÉGOU, H., DOUZE, M., SCHMID, C., AND PÉREZ, P. Aggregating local descriptors into a compact image representation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (2010), pp. 3304–3311. pages 38

[66] JEGOU, H., PERRONNIN, F., DOUZE, M., SCHMID, C., ET AL. Aggregating local image descriptors into compact codes. *IEEE Trans. Pattern Anal. Mach. Intell. 34*, 9 (2012), 1704–1716. pages 38

[67] JIANG, Y., AND NGO, C. Visual word proximity and linguistic for semantic video indexing and near-duplicate retrieval. *Comp. Vis. Image Understanding 113*, 3 (2009), 405–414. pages 44, 75, 94, 96

[68] JOACHIMS, T. Optimizing search engines using clickthrough data. In *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining* (2002), pp. 133–142. pages 136

[69] KADIR, T., ZISSERMAN, A., AND BRADY, M. An affine invariant salient region detector. In *Proc. Eur. Conf. Comp. Vis.* Springer, 2004, pp. 228–241. pages 15

[70] KE, Y., SUKTHANKAR, R., AND HUSTON, L. Efficient near-duplicate detection and sub-image retrieval. In *Proc. ACM Int. Conf. on Multimedia* (2004), vol. 4, p. 5. pages 45

[71] KENDALL, M. A new measure of rank correlation. *Biometrika 30*, 1/2 (1938), 81–93. pages 136, 138

[72] KNOPP, J., SIVIC, J., AND PAJDLA, T. Avoiding confusing features in place recognition. In *Proc. Eur. Conf. Comp. Vis.* Springer, 2010, pp. 748–761. pages 43

[73] KOVASHKA, A., PARIKH, D., AND GRAUMAN, K. Whittlesearch: Image search with relative attribute feedback. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (2012), pp. 2973–2980. pages 136

[74] KULIS, B., AND GRAUMAN, K. Kernelized locality-sensitive hashing for scalable image search. In *Proc. IEEE Int. Conf. Comp. Vis.* (2009), IEEE, pp. 2130–2137. pages 44

[75] KUO, Y., CHEN, K., CHIANG, C., AND HSU, W. Query expansion for hash-based image object retrieval. In *Proc. ACM Int. Conf. on Multimedia* (2009), pp. 65–74. pages 136

[76] LAZAREVIC, A., AND OBRADOVIC, Z. Boosting algorithms for parallel and distributed learning. *Distributed and Parallel Databases 11*, 2 (2002), 203–229. pages 166

[77] LAZEBNIK, S., SCHMID, C., AND PONCE, J. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (2006), pp. 2169–2178. pages 39

[78] LEPETIT, V., AND FUA, P. Keypoint recognition using randomized trees. *IEEE Trans. Pattern Anal. Mach. Intell. 28*, 9 (2006), 1465–1479. pages 36

[79] LI, X., HU, W., SHEN, C., ZHANG, Z., DICK, A., AND HENGEL, A. V. D. A survey of appearance models in visual object tracking. *IEEE Trans. Knowledge and Data Engineering* (2013). pages 12

[80] LI, Y., SNAVELY, N., AND HUTTENLOCHER, D. P. Location recognition using prioritized feature matching. In *Proc. Eur. Conf. Comp. Vis.* 2010, pp. 791–804. pages 49

[81] LIN, W., JIN, R., AND HAUPTMANN, A. Web image retrieval re-ranking with relevance model. In *Proc. IEEE/WIC Int. Conf. Web Intelligence* (2003), pp. 242–248. pages 136

[82] LIU, Y., ZHANG, D., LU, G., AND MA, W. A survey of content-based image retrieval with high-level semantics. *Pattern Recogn. 40*, 1 (2007), 262–282. pages 11, 16, 43

[83] LOWE, D. Distinctive image features from scale-invariant keypoints. *Int. J. Comp. Vis. 60*, 2 (2004), 91–110. pages 15, 18, 20

[84] LOWE, D. G. *Perceptual Organization and Visual Recognition*. Kluwer Acdemic Publishers, 1985. pages 19

[85] LOWE, D. G. The viewpoint consistency constraint. *Int. J. Comp. Vis. 1*, 1 (1987), 57–72. pages 14, 29

[86] LV, Q., JOSEPHSON, W., WANG, Z., CHARIKAR, M., AND LI, K. Multi-probe lsh: efficient indexing for high-dimensional similarity search. In *Int. Conf. on Very Large Data Bases* (2007), VLDB Endowment, pp. 950–961. pages 45

[87] MA, W.-Y., AND MANJUNATH, B. Netra: A toolbox for navigating large image databases. In *Proc. IEEE Int. Conf. Image Process.* (1997), vol. 1, IEEE, pp. 568–571. pages 13

[88] MAKADIA, A. Feature tracking for wide-baseline image retrieval. In *Proc. Eur. Conf. Comp. Vis.* (2010), pp. 310–323. pages 38, 42

[89] MATAS, J., CHUM, O., URBAN, M., AND PAJDLA, T. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vis. Comput. 22*, 10 (2004), 761–767. pages 15, 36

[90] MEHROTRA, R., AND GARY, J. E. Similar-shape retrieval in shape data management. *Computer 28*, 9 (1995), 57–62. pages 14

[91] MELUCCI, M. Weighted rank correlation in information retrieval evaluation. In *Information Retrieval Technology*. Springer, 2009, pp. 75–86. pages 138

[92] MENG, J., YUAN, J., JIANG, Y., NARASIMHAN, N., VASUDEVAN, V., AND WU, Y. Interactive visual object search through mutual information maximization. In *Proc. ACM Int. Conf. on Multimedia* (2010), pp. 1147–1150. pages 162

[93] MIKOLAJCZYK, K., AND SCHMID, C. An affine invariant interest point detector. *Proc. Eur. Conf. Comp. Vis.* (2002), 128–142. pages 15, 36

[94] MIKOLAJCZYK, K., AND SCHMID, C. Scale & affine invariant interest point detectors. *Int. J. Comp. Vis. 60*, 1 (2004), 63–86. pages 33

[95] MIKOLAJCZYK, K., TUYTELAARS, T., SCHMID, C., ZISSERMAN, A., MATAS, J., SCHAFFALITZKY, F., KADIR, T., AND VAN GOOL, L. A comparison of affine region detectors. *Int. J. Comp. Vis. 65*, 1 (2005), 43–72. pages 16, 29, 36

[96] MIKULIK, A., PERDOCH, M., CHUM, O., AND MATAS, J. Learning a Fine Vocabulary. In *Proc. Eur. Conf. Comp. Vis.* (2010), pp. 1–14. pages 38, 47, 62, 69, 91, 93, 114, 131, 151, 174, 180, 181

[97] MUJA, M., AND LOWE, D. G. Fast approximate nearest neighbors with automatic algorithm configuration. In *Proc. Int. Conf. on Computer Vision Theory and Application* (2009), pp. 331–340. pages 23, 33, 39

[98] MURASE, H., AND NAYAR, S. Visual learning and recognition of 3-d objects from appearance. *Int. J. Comp. Vis. 14*, 1 (1995), 5–24. pages 19

[99] NEWMAN, M., AND GIRVAN, M. Finding and evaluating community structure in networks. *Physical review E 69*, 2 (2004), 026113. pages 102

[100] NISTER, D., AND STEWENIUS, H. Scalable recognition with a vocabulary tree. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (2006), pp. 2161–2168. pages 23, 24, 37

[101] PAN, S. J., AND YANG, Q. A survey on transfer learning. *IEEE Trans. on Know. and Data Engi. 22*, 10 (2010), 1345–1359. pages 184

[102] PEDRONETTE, D., AND DA S. TORRES, R. Exploiting contextual information for image re-ranking. *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications* (2010), 541–548. pages 136

[103] PELEG, S., WERMAN, M., AND ROM, H. A unified approach to the change of resolution: Space and gray-level. *IEEE Trans. Pattern Anal. Mach. Intell. 11*, 7 (1989), 739–742. pages 44

[104] PENTLAND, A., PICARD, R. W., AND SCLAROFF, S. Photobook: Content-based manipulation of image databases. *Int. J. Comp. Vis. 18*, 3 (1996), 233–254. pages 9

[105] PERD'OCH, M., CHUM, O., AND MATAS, J. Efficient representation of local geometry for large scale object retrieval. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (2009). pages 40, 47, 62, 69, 93, 114, 131, 149, 150, 151, 174, 181

[106] PHILBIN, J., CHUM, O., ISARD, M., SIVIC, J., AND ZISSERMAN, A. Object retrieval with large vocabularies and fast spatial matching. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (2007), pp. 1–8. pages 12, 20, 22, 23, 24, 26, 29, 33, 34, 36, 37, 40, 41, 42, 43, 46, 47, 51, 61, 62, 65, 69, 88, 89, 90, 92, 93, 112, 114, 127, 129, 131, 134, 135, 137, 145, 147, 150, 151, 162, 163, 174, 180, 181, 182

[107] PHILBIN, J., CHUM, O., ISARD, M., SIVIC, J., AND ZISSERMAN, A. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (2008). pages 23, 39, 47, 55, 61, 62, 69, 91, 93, 94, 99, 112, 114, 131, 151, 174, 180, 181

[108] PHILBIN, J., ISARD, M., SIVIC, J., AND ZISSERMAN, A. Descriptor learning for efficient retrieval. In *Proc. Eur. Conf. Comp. Vis.* (2010), pp. 677–691. pages 34, 37, 47, 61, 62, 65, 66, 69, 87, 91, 92, 93, 114, 131, 151, 174, 180, 181

[109] PHILBIN, J., SIVIC, J., AND ZISSERMAN, A. Geometric LDA: A generative model for particular object discovery. In *Proc. Brit. Mach. Vis. Conf.* (2008). pages 48, 100, 155

[110] PHILBIN, J., SIVIC, J., AND ZISSERMAN, A. Geometric latent dirichlet allocation on a matching graph for large-scale image datasets. *Int. J. Comp. Vis. 95*, 2 (2011), 138–153. pages 48, 69, 100, 102, 120

[111] PHILBIN, J., AND ZISSERMAN, A. Object mining using a matching graph on very large image collections. In *Proc. Indian Conf.Computer Vision, Graphics & Image Processing* (2008), pp. 738–745. pages 47, 48, 100, 102, 155

[112] PLATANIOTIS, K. N., AND VENETSANOPOULOS, A. N. *Color image processing and applications*. Springer, 2000. pages 13

[113] PRITCHETT, P., AND ZISSERMAN, A. Wide baseline stereo matching. In *Proc. IEEE Int. Conf. Comp. Vis.* (1998), IEEE, pp. 754–760. pages 15

[114] QIN, D., GAMMETER, S., BOSSARD, L., QUACK, T., AND VAN GOOL, L. Hello neighbor: accurate object retrieval with k-reciprocal nearest neighbors. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (2011). pages 47, 62, 69, 93, 114, 119, 131, 136, 151, 174, 181

[115] RASTEGARI, M., FANG, C., AND TORRESANI, L. Scalable object-class retrieval with approximate and top-k ranking. In *Proc. IEEE Int. Conf. Comp. Vis.* (2011), pp. 2659–2666. pages 160, 161, 162

[116] ROBERTS, L. G. *Machine perception of three-dimensional solids*. 1965. pages 19

[117] RUBNER, Y., TOMASI, C., AND GUIBAS, L. The earth mover's distance as a metric for image retrieval. *Int. J. Comp. Vis. 40*, 2 (2000), 99–121. pages 18, 44, 94

[118] RUI, Y., HUANG, T. S., AND CHANG, S.-F. Image retrieval: Current techniques, promising directions, and open issues. *Journal of Visual Communication and Image Representation 10*, 1 (1999), 39–62. pages 11

[119] SALTON, G., AND BUCKLEY, C. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science 24* (1999), 288–297. pages 40

[120] SALTON, G., AND MCGILL, M. *Introduction to modern information retrieval*. McGraw-Hill, Inc., 1986. pages 25, 135

[121] SALTON, G., WONG, A., AND YANG, C. A vector space model for automatic indexing. *Communications of the ACM 18*, 11 (1975), 613–620. pages 25

[122] SCHAFFALITZKY, F., AND ZISSERMAN, A. Automated location matching in movies. *Comp. Vis. Image Understanding 92*, 2 (2003), 236–264. pages 15

[123] SCHMID, C., AND MOHR, R. Local grayvalue invariants for image retrieval. *IEEE Trans. Pattern Anal. Mach. Intell. 19*, 5 (1997), 530–535. pages 15, 20

[124] SCHMID, C., AND ZISSERMAN, A. Automatic line matching across views. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (1997), pp. 666–671. pages 15, 20

[125] SHI, J., AND MALIK, J. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell. 22*, 8 (2000), 888–905. pages 102

[126] SHOTTON, J., JOHNSON, M., AND CIPOLLA, R. Semantic texton forests for image categorization and segmentation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (2008), IEEE, pp. 1–8. pages 36

[127] SILPA-ANAN, C., AND HARTLEY, R. Optimised kd-trees for fast image descriptor matching. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (2008), pp. 1–8. pages 24

[128] SIMON, I., SNAVELY, N., AND SEITZ, S. Scene summarization for online image collections. In *Proc. IEEE Int. Conf. Comp. Vis.* (2007). pages 30, 48, 105

[129] SIVIC, J., SCHAFFALITZKY, F., AND ZISSERMAN, A. Object level grouping for video shots. *Int. J. Comp. Vis. 67*, 2 (2006), 189–210. pages 36

[130] SIVIC, J., AND ZISSERMAN, A. Video Google: A text retrieval approach to object matching in videos. In *Proc. IEEE Int. Conf. Comp. Vis.* (2003), pp. 1470–1477. pages 3, 17, 20, 22, 26, 34, 39, 42, 43, 51, 73, 97, 119, 135, 141, 145

[131] SMEULDERS, A., WORRING, M., SANTINI, S., GUPTA, A., AND JAIN, R. Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell. 22*, 12 (2000), 1349–1380. pages 11, 13, 43

[132] SNAVELY, N., GARG, R., SEITZ, S. M., AND SZELISKI, R. Finding paths through the world's photos. In *ACM Transactions on Graphics* (2008), vol. 27, p. 15. pages 49

[133] SNAVELY, N., SEITZ, S., AND SZELISKI, R. Photo tourism: exploring photo collections in 3d. In *ACM Transactions on Graphics* (2006), vol. 25, pp. 835–846. pages 37, 49

[134] SPEARMAN, C. The proof and measurement of association between two things. by c.spearman, 1904. *The American Journal of Psychology 100*, 3/4 (1987), 441–471. pages 136, 138

[135] STOCKMAN, G. Object recognition and localization via pose clustering. *Comp. Vis., Grap., and Image Proc. 40*, 3 (1987), 361–387. pages 14

[136] SWAIN, M., AND BALLARD, D. Color indexing. *Int. J. Comp. Vis. 7*, 1 (1991), 11–32. pages 13

[137] TAMURA, H., MORI, S., AND YAMAWAKI, T. Textural features corresponding to visual perception. *IEEE Trans. on Systems, Man and Cybernetics 8*, 6 (1978), 460–473. pages 13

[138] TANG, W., CAI, R., LI, Z., AND ZHANG, L. Contextual synonym dictionary for visual object retrieval. In *Proc. ACM Int. Conf. on Multimedia* (2011), pp. 503–512. pages 40, 92, 93, 114

[139] TOLA, E., LEPETIT, V., AND FUA, P. A fast local descriptor for dense matching. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (2008), IEEE, pp. 1–8. pages 36

[140] TOLA, E., LEPETIT, V., AND FUA, P. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Trans. Pattern Anal. Mach. Intell. 32*, 5 (2010), 815–830. pages 15

[141] TORRALBA, A., FERGUS, R., AND WEISS, Y. Small codes and large image databases for recognition. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (2008), IEEE, pp. 1–8. pages 45

[142] TURCOT, P., AND LOWE, D. Better matching with fewer features: The selection of useful features in large database recognition problems. In *ICCV Workshops* (2009), pp. 2109–2116. pages 42, 47, 61, 62, 64, 65, 69, 93, 112, 114, 131, 151, 155, 174, 181

[143] TUYTELAARS, T., AND MIKOLAJCZYK, K. Local invariant feature detectors: a survey. *Foundations and Trends® in Computer Graphics and Vision 3*, 3 (2008), 177–280. pages 12

[144] TUYTELAARS, T., AND VAN GOOL, L. Content-based image retrieval based on local affinely invariant regions. In *Visual Information and Information Systems* (1999), pp. 656–656. pages 15, 20

[145] TUYTELAARS, T., AND VAN GOOL, L. Wide baseline stereo matching based on local, affinely invariant regions. In *Proc. Brit. Mach. Vis. Conf.* (2000), vol. 2, p. 4. pages 15

[146] VAN GEMERT, J., VEENMAN, C., SMEULDERS, A., AND GEUSEBROEK, J. Visual word ambiguity. *IEEE Trans. Pattern Anal. Mach. Intell. 32*, 9 (2009), 1271–1283. pages 39

[147] VIOLA, P., AND JONES, M. Rapid object detection using a boosted cascade of simple features. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (2001), vol. 1, IEEE, pp. I–511. pages 166

[148] WANG, J. Z., LI, J., AND WIEDERHOLD, G. Simplicity: Semantics-sensitive integrated matching for picture libraries. *IEEE Trans. Pattern Anal. Mach. Intell. 23*, 9 (2001), 947–963. pages 13

[149] WANG, X., YANG, M., COUR, T., ZHU, S., YU, K., AND HAN, T. Contextual weighting for vocabulary tree based image retrieval. In *Proc. IEEE Int. Conf. Comp. Vis.* (2011), pp. 209–216. pages 91

[150] WEBBER, W., MOFFAT, A., AND ZOBEL, J. A similarity measure for indefinite rankings. *ACM Transactions on Information Systems 28*, 4 (2010), 20. pages 138

[151] WINDER, S., HUA, G., AND BROWN, M. Picking the best daisy. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (2009), IEEE, pp. 178–185. pages 36, 37

[152] WINDER, S. A., AND BROWN, M. Learning local image descriptors. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (2007), IEEE, pp. 1–8. pages 36

[153] WU, L., HOI, S. C., JIN, R., ZHU, J., AND YU, N. Distance metric learning from uncertain side information for automated photo tagging. *ACM Tran. on Inte. Syst. and Tech. (TIST) 2*, 2 (2011), 13. pages 43

[154] WU, L., HUA, X.-S., YU, N., MA, W.-Y., AND LI, S. Flickr distance. In *Proc. ACM Int. Conf. on Multimedia* (2008), pp. 31–40. pages 43

[155] WU, L., HUA, X.-S., YU, N., MA, W.-Y., AND LI, S. Flickr distance: A relationship measure for visual concepts. *IEEE Trans. Pattern Anal. Mach. Intell. 34*, 5 (2012), 863–875. pages 43

[156] YATES, R., AND NETO, B. *Modern information retrieval*. ACM Press, New York, 1999. pages 28

[157] YILMAZ, E., ASLAM, J. A., AND ROBERTSON, S. A new rank correlation coefficient for information retrieval. In *Int. Conf. on Research and Cevelopment in Information Retrieval* (2008), ACM, pp. 587–594. pages 138

[158] YUAN, J., WU, Y., AND YANG, M. Discovery of collocation patterns: from visual words to visual phrases. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (2007), pp. 1–8. pages 40

[159] ZHANG, Y., JIA, Z., AND CHEN, T. Image retrieval with geometry-preserving visual phrases. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (2011). pages 40, 47, 61, 62, 69, 91, 93, 114, 131, 135, 151, 174, 180, 181

[160] ZOBEL, J., AND MOFFAT, A. Inverted files for text search engines. *ACM Computing Surveys 38*, 2 (2006), 6. pages 26, 27

[161] ZOBEL, J., MOFFAT, A., AND RAMAMOHANARAO, K. Inverted files versus signature files for text indexing. *ACM Transactions on Database Systems 23*, 4 (1998), 453–490. pages 26