Context Based Re-ranking for Object Retrieval

Yanzhi Chen[1,2], Anthony Dick[2], Xi Li[2,3], Rhys Hill[2]

[1]United Technologies Research Center (China) Ltd., Room 3502, 35/F, Kerry Parkside Office, Shanghai, China
[2]Australian Centre for Visual Technologies, The University of Adelaide, SA 5005, Australia
[3]College of Computer Science and Technologies, Zhejiang University, China

Abstract. We propose a simple but effective re-ranking method for improving the results of object retrieval. Our method considers the contextual information embedded in a dataset. This is based on the observation that if there are multiple images containing the same object in a dataset, then these images can often be grouped into clusters. We make the following two contributions. Firstly, we gain this contextual information by a *random dimension partition* of the dataset. This enables online query model expansion if needed. Secondly, we use the collected contextual information to refine the initial retrieval results by taking into account the context in which each retrieved image occurs. Experimental results on several datasets demonstrate the effectiveness of our method in both accuracy and computation cost: our method refines retrieval results without relying on low-level feature matching or re-issuing the query.

1 Introduction

The goal of object retrieval is to find other images in a dataset containing a target object given in a query image. The target object may appear under varying image conditions, including scale, viewpoint, lighting changes, or partial occlusion of the objects [1]. Therefore, many successful object retrieval systems are based on local invariant descriptors [2, 3, 1, 4, 5], that are robust to scale, affine transformations and partial occlusion [6].

In order to scale to large datasets, an image is reduced into a compressed 'Bag-of-Words" (BoW) format [2] instead of using thousands of local invariant descriptors. In the BoW model, each "word" is a quantised image feature, typically weighted according to its frequency in the image (term frequency, tf) and rarity in the dataset (inverse document frequency, idf). However, the compressed format decreases the retrieval accuracy because of the quantisation process [7].

Over the past decade, there has been considerable improvement on the BoW based retrieval system, focusing on better exploiting low-level feature information [1, 8, 7, 9–12]. It has also been noticed [13, 10, 11] that an image dataset usually contains multiple images showing the same object. Nevertheless, little attention has been paid to the fact that these images containing the same object will appear as clusters in the dataset as opposed to background noise (see Fig. 1

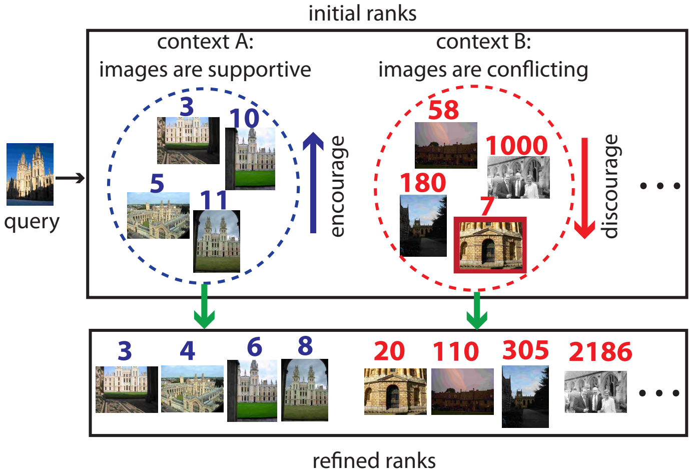


Fig. 1. Illustration of context based re-ranking. The initial retrieval results (sorted by similarity Ψ) contain both true and false positives. The numbers above images are rankings. The accuracy is increased after re-ranking by contextual information. Best viewed in color.

for example). These clustered images can be treated as contextually associated. To this end, we define a "**context**" as a group of images having common visual properties and show its benefits in improving the retrieval performance.

Unlike many previous methods that require an expensive offline training stage, the first contribution of this paper is a simple, efficient clustering method for online context generation. Traditional methods, *e.g.* k-means, are unable to cluster the BoW vectors efficiently, because the vectors are very sparse and high dimensional. Instead, we propose a *random space partition* method, by which the dataset is clustered into groups of images by partitions along randomly selected dimensions. This is effective because of the sparsity of the BoW vectors, and efficient despite their high dimensionality because only a small number of dimensions are selected. The scheme of this simplified clustering method is similar to [14], in which a random projection of high dimensional data is conducted for multiple runs as well.

Our second contribution is to improve the initial retrieval results once the contexts are available, by the analysis of image ranks in each context. The idea is that the similarity between a query and a dataset image should not be determined solely by those two images (as in the standard dot-product similarity [2]), but also influenced by their association with other contextually related images. As illustrated in Fig. 1, images in a context are promoted if they support each other, *i.e.* high retrieval ranks dominate, and therefore ranked highly (context A). Otherwise, images are demoted if there are low or conflicting ranks in their context (context B). Our context based re-ranking method refines the retrieval results by improving the standard one-to-one comparison (dot-product similarity) with the contextual information at query time.

The rest of the paper is organised as follows: Section 2 discusses the related work to our method. Section 3 presents the method to extract the contextual information, which is used in Section 4 to re-rank the dataset. Section 5 reports the re-ranking results on some public datasets and compares them to state-of-the-art. Finally we draw a conclusion in Section 6.

2 Related work

There have been extensive studies aiming to increase the accuracy of BoW based retrieval system. One approach is to improve the BoW image descriptors, for example forming a discriminative visual vocabulary [8], mapping multiple visual words to a single feature [7, 9], or deriving a query adaptive similarity based on feature-to-feature similarity [12]. Another approach is to re-rank the retrieval results as a post-process by analysing an initial set of query results [1, 4, 15, 16]. Compared to the first approach, online re-ranking does not require reconstruction of the visual vocabulary, nor does it require training data. In this paper, we adopt the second approach: *improving the retrieval performance by an online re-ranking process.*

A popular approach to online re-ranking is to utilise low-level spatial information to promote dataset images whose features are spatially consistent with those in the query. Spatial verification [1] examines a truncated list of retrieved results by computing a geometric transformation between features in both query and dataset images. However, the computation of geometric transformation is expensive, so that only a short-list of images can be examined. To speed it up, weak geometric consistency (WGC) [4] filters mismatching descriptors without applying geometric transformation. Instead, the method assumes that matching descriptors are related by a fixed orientation and scale. Re-ranking can also be achieved by testing reciprocal similarity of query and dataset images [13]—that is, whether the images retrieve each other when both used as queries. Reciprocal similarity is discovered by a k-reciprocal nearest neighbour structure that is built offline. In addition, features from verified top ranked results can be added to the query which is then re-issued, in order to improve recall [15, 16]. These methods succeed in finding more query relevant images, but at the cost of online feature matching or query re-issuing, which is computationally expensive.

Rather than examining individual features, the ranking information embedded in a set of top ranked results can be used to further refine the results [17, 18]. In [17], a distance matrix is defined by the similarity of the ranks to take into account the contextual information, while the method proposed in [18] measures the similarity between the query and dataset images based on the idea that images are visually similar if they have intersections among top ranked results when using them as queries. However, these methods need to re-query the dataset in order to re-define the distance between query and dataset images.

In this paper, we present a novel re-ranking method for BoW based object retrieval, which achieves both efficiency and effectiveness. Our work is also inspired by [10] in terms of adjusting the similarity scores of images. Their method applies an offline image graph creation step in which each node represents an image and an edge indicates the connection of same objects in a pair of images, such that neighbouring images are grouped. Our work is different from [10] in two aspects: *i*) our method softly adjusts the similarity scores of dataset images at runtime with the help of the contextual information (although the offline step is optional); *ii*) the adjustments of similarity scores are based on query-specific rank analysis performed at runtime.

3 Context generation

A key ingredient of our method is to generate "contexts" from the dataset. This section describes how to derive such information efficiently from the BoW vectors. The usage of contextual information for re-ranking is discussed in the following Section.

3.1 Random space partition

Let $\mathcal{T}$ be a set of dataset images. The goal of our method is to cluster $\mathcal{T}$ into D groups: $\mathcal{C} := \{\mathbf{c}_k\}_{k=1}^{D}$, where each group $\mathbf{c}_k$ is a context that contains a small number (n_k) of dataset images. The clustering of $\mathcal{T}$ involves two issues: *i*) scalability: the clustering is conducted on high-dimensional BoW vectors, for which standard k-means methods or graph cut of the image dataset [19] are not feasible; and *ii*) efficiency: as it runs at query time, the partition should have low computation and memory requirements.

In order to address these two issues, we use a *random space partition* method which utilises the specific properties of the BoW model. A visual vocabulary is composed of N visual words: $\mathbf{W} := \{w_i\}_{i=1}^{N}$, where N is typically large ($N = 10^6$ in our implementation). The dataset images are represented as a collection of visual word vectors $\mathcal{T} := \{\mathbf{d}_j\}_{j=1}^{V}$, in which V is number of images and $\mathbf{d}_j$ is the corresponding tf-idf image vector.

The dataset vectors $\mathcal{T}$ are very sparse: on average, there are only 2200 non-zero entries in a 10^6 dimensional vector (in our implementation). The high sparsity simplifies the partitioning of $\mathcal{T}$. As illustrated in Fig. 2 (a), the dataset vectors $\mathcal{T}$ are separated into two groups by a random dimension of the image vectors, according to whether each vector contains a non-zero entry in a specific dimension. Note that each dimension of $\mathbf{d}_j$ corresponds to one visual word w_i, so the images can be quickly accessed by an inverted file [2], which maps each visual word to images it appears in. Thus, each "column" of the file, as is shown in Fig. 2 (b), corresponds to a visual word w_i and forms an image group $\mathbf{c}_k$:

$$\mathbf{c}_k = \{\mathbf{d}_j\}_{j=1}^{n_k} \text{ if } F_j(w_i) > 0 \tag{1}$$

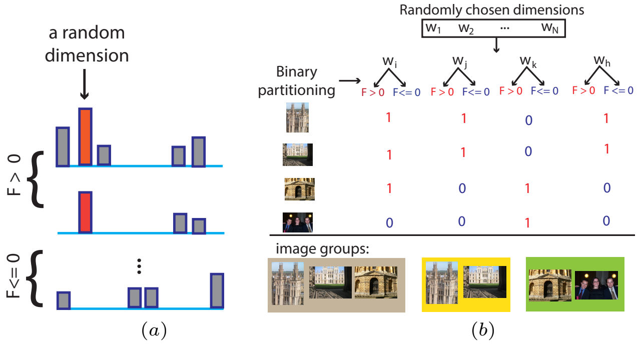


Fig. 2. (a): Image vector separation by a random dimension into two parts. (b): Illustration of random space partition method by using the inverted file. Best viewed in color.

where $F_j(w_i)$ is the frequency of w_i in image j. Scalable clustering of $\mathcal{T}$ 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.

The efficiency of our method is achieved by performing only D ($D \ll N$) data partitions to generate groups $\mathcal{C}$. Intuitively, it is inefficient to use all dimensions (visual words) because only a small number of them are informative. These words are usually the query words $\mathbf{Q}$, thus the random dimension partition is conducted on $\mathbf{Q}$ by default. In addition, an extra set of words $\mathbf{S}$ ($\mathbf{Q}, \mathbf{S} \subset \mathbf{W}$), which are relevant to the query, can also be considered.

Query-relevant words $\mathbf{S}$ can be generated either offline or online as follows: i) **offline**: obtain $\mathbf{S}$ by a thesaurus structure built offline [20], which includes the frequently co-occurring visual words in fixed spatial regions. ii) **online**: obtain $\mathbf{S}$ by query expansion [15], in which the visual words included in the spatially verified regions are appended.

3.2 Expansion of contexts

In order to promote the query-relevant words $\mathbf{S}$ (as well as keep the query words $\mathbf{Q}$), we propose an expansion method during context generation which adopts a weighted random selection scheme to select the dimensions. Because each dimension in the BoW vector is also associated to a visual word, this method can be seen as an expansion of the original query words $\mathbf{Q}$ obtained without re-issuing the query. The expansion proceeds as follows.

Firstly, we randomly choose a subset of words in which $\mathbf{Q}$ and $\mathbf{S}$ are given a higher probability of selection than those words that are non relevant. This is done by associating visual words to random keys under a mapping function f:

$$f(w_i) = \begin{cases} a \cdot x \text{ if } w_i \in \mathbf{Q} \\ b \cdot x \text{ if } w_i \in \mathbf{S} \backslash (\mathbf{Q} \cap \mathbf{S}) \\ x \quad \text{ otherwise} \end{cases} \tag{2}$$

where $w_i \in \mathbf{W}$, x is a uniformly distributed random variable $x \in U(0,1)$ and the parameters a, b are the weights. The D dimensions used for partition are selected in decreasing order of $f(w_i)$. This scheme of random dimension selection is similar to [21].

Secondly, we define three cases based on the values of a and b, such that the query information can be incrementally appended by adjusting the parameters: **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 and the query-relevant set $\mathbf{S}$ are more likely to be selected than others. After obtaining D dimensions, image groups $\mathcal{C} := \{\mathbf{c}_k\}_{k=1}^{D}$ are used to estimate the context scores for re-ranking.

4 Context based re-ranking

This section describes our context based re-ranking scheme. We start with the baseline method that sorts the dataset images according to their dot-product similarity [2] between the tf-idf vectors $\mathbf{q}$ and $\mathbf{d}$, corresponding to query image q and a dataset image d:

$$\Psi(q,d) = \frac{\mathbf{q} \cdot \mathbf{d}}{\parallel \mathbf{q} \parallel \parallel \mathbf{d} \parallel} \tag{3}$$

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 low ranks are likely irrelevant to the query. The ranking is efficient, but neglects contextual information linking the returned results as it only measures similarity between the query and each dataset image in isolation.

As illustrated in Fig. 1, the ranks of all dataset images in a given context can be informative. If many images in a context are relevant to a query, then this supports ranking all images in that context more highly. Conversely, if many images in a context have low rank, then a high ranked exception is likely to be a false positive. Therefore, our method aims to improve one-to-one matching by embedding this information in the similarity measure Ψ (Eq. 3), such that contextually similar images boost each other up. To this end, we use the contextual ranking information to adjust the dot-product similarity Ψ:

$$\Phi(q,d) = \Psi(q,d) \cdot \exp(\Theta(q,d)) \tag{4}$$

where $\Phi(q,d)$ is the improved ranking score. The context factor $\Theta(q,d)$ in Eq. (4) is calculated according to the ranks of result images belonging to each context, and is discussed below. Images are re-ranked by sorting $\Phi(q,d)$ (Eq. (4)). Our method is outlined in Algorithm 1.

Algorithm 1 Context based re-ranking

```
Input: Query image $q$, number of random dimensions $D$.
Output: Retrieval results.
1. Rank dataset images by sorting the dot-product similarity $\Psi$ (Eq. 3).
2. Obtain initial ranks of dataset images.
3. Select $D$ dimensions (Eq. (2)).
4. Generate image groups $\mathcal{C} := \{\mathbf{c}_k\}_{k=1}^{D}$ from inverted file (Eq. (1)).
5. Compute the context score $W(q, \mathbf{c}_k)$ for each image group (Eq. (5)).
6. Compute the context factor $\Theta(q, d)$ for each dataset image $d$ (Eq. (6)).
7. Adjust image similarity and re-rank (Eq. (4)).
Return: Re-ranked results.
```

4.1 Computing the context factor for re-ranking

Our context based re-ranking method introduces the contextual information to the similarity measure by a pair of contextual measures: a context factor and a context score. A query-specific context score $W(q, \mathbf{c}_k)$ describes the association of each image group $\mathbf{c}_k$, while the context factor $\Theta(q, d)$ of a dataset image d is formed by context scores learnt from n_d image groups it has been assigned to. The dataset image d is then re-ranked by the similarity score refined by the context factor (Eq. (4)). Specifically, our method proceeds in two steps.

Firstly, each image group is assigned a context score $W(q, \mathbf{c}_k)$, which summarises the ranks of images in the group. This is composed of two parts:

- The coherence of image ranks in $\mathbf{c}_k$: $\frac{1}{n_k^2}\sum_{j=1}^{n_k}\sum_{s=1}^{n_k} K(\frac{r_j - r_s}{\rho})$, 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 coherence of a context is measured by the association of image ranks in $\mathbf{c}_k$. The parameter ρ is automatically tuned, based on estimating the standard deviation of the input image ranks [22]. Thus, image groups which are distributed widely in the ranking list have less coherence, and will not be weighted strongly in the refined similarity.
- The number of top and 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-H 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 the product of both:

$$W(q, \mathbf{c}_k) = \left[\frac{1}{n_k^2}\sum_{j=1}^{n_k}\sum_{s=1}^{n_k} K(\frac{r_j - r_s}{\rho})\right] \cdot \frac{t_q(\mathbf{c}_k, H) - b_q(\mathbf{c}_k, H)}{n_k} \tag{5}$$

Secondly, the re-ranking process utilises these context scores to improve the similarity score of a dataset image d. We index each dataset image d by a set of D indicators $\{\mathbb{I}_k^d\}_{k=1}^{D}$, where $\mathbb{I}_k^d$ indicates whether d appears in $\mathbf{c}_k$ or not. As each image is assigned to several groups, the re-ranking then makes use of the average context score. The context factor is obtained from these context scores, and is defined for a response image d to query image q as:

$$\Theta(q,d) = \frac{1}{\sum_{k=1}^{D} \mathbb{I}_k^d} \cdot \sum_{k=1}^{D} \mathbb{I}_k^d W(q, \mathbf{c}_k) \tag{6}$$

According to Eq. (4), the initial similarity of images having negative context factor ($\Theta(q,d) < 0$) is decreased, while those having positive context factor ($\Theta(q,d) > 0$) is increased.

5 Experimental results

5.1 Experimental setup

We investigate the performance of our context based re-ranking method in the following aspects: *i*) varying the key parameters applied to context generation (random space partition and context expansion). *ii*) varying the context re-ranking parameters, which include the number of iterations as well as the top (bottom) truncation. *iii*) comparison to state-of-the-art. The details of our experimental settings are as follows.

Datasets: The retrieval experiments are conducted on three public object retrieval datasets: two small scale datasets (Oxford 5K and Paris 6K [23]) and a large scale dataset Oxford 105K consisting of Oxford 5K images and 100K images from MIRFLICKR-1M [24]. Both the Oxford 5K and Paris 6K datasets [23] contain 11 building landmarks for evaluation. Each image within these datasets is represented as a histogram of SIFT words after tf-idf weighting.

Implementation details: The visual words are obtained by quantising the SIFT feature descriptors using approximate k-means [1, 25]. The vocabulary size is 1 million. After that, images are stored in an inverted file structure such that the online process only needs to access those containing query words (or query related words as discussed in Section 3.2). We run our experiments on 2×8-Core Xeon E5-2680 at 2.70GHz with 10G memory.

Evaluation: In order to quantify the retrieval performance, we evaluate the retrieval accuracy by the widely used mean average precision (mAP), as defined in [2]. The mAP scores reported in the following are from our implementation, excepts those cited from other sources.

5.2 Evaluation of random space partition

Initially, we evaluate the effects of various parameter settings in the random space partition. As discussed in Section 3.1, the random space partition involves two key parameters:

1) Query-dependent set Q selection weight, achieved by adjusting the weight a in the random mapping function (Eq. 2). We illustrate the effects of query-dependent set $\mathbf{Q}$ by varying a while fixing the other weighting parameter

Methods		Oxford 5K	Paris 6K
Baseline (without re-ranking)		0.612	0.639
Spatial Re-ranking		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**

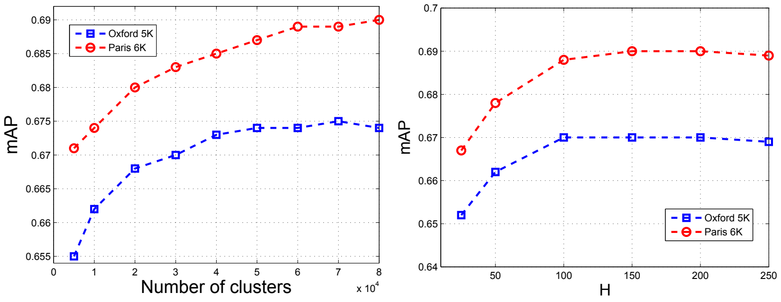
Table 1. Retrieval performance with varying weighting functions in ordering the query words. See text for details. Total number of visual words selected: 3×10^4.

($b = 1$). In this way, the weight a ($a > 1$) gives more priority to the query words than those not in the query ($a = 1$). Table 1 assesses these effects as follows: *i*) (f_1): $a = 1$, random selection of visual words (dimensions). *ii*) (f_2): $a = 10$, query words 10× more likely to be selected. *iii*) (f_3): $a = tf$, similar to f_2 but weight a is proportional to the term frequency of the query word, rather than constant as in f_2. As reported in Table 1, the retrieval results of f_1 are as good as spatial verification on the Oxford 5K dataset, while achieving slightly higher accuracy on the Paris 6K dataset. Note that f_1 can be completed offline, so our random selection method is able to re-rank the dataset effectively and efficiently but with less information required than the standard spatial verification method. In contrast, the mapping functions f_2 and f_3 are query-specific, namely the dimensions are decided according to the given query online. They result in more accurate retrieval accuracy than the offline version (f_1), as well as outperform the spatial verification results on both datasets. The difference between f_2 and f_3 is negligible when a is large. As a result, we set weight $a = 10$ whenever query words require priority in random space partition.

2) Number of randomly chosen dimensions D. Fig. 3 (a) reports the retrieval accuracy for increasing D dimensions selected. Note that the context generation utilises these dimensions to collect contextual information during re-ranking. As illustrated in Fig. 3 (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 datasets. Fig. 3 (a) also validates that the re-ranking performance improves by diminishing amounts as D increases. In addition, Table 2 shows the average CPU time as D increases, in which the CPU time rises consistently with increasing dimension number D. Considering both accuracy and runtime, we set $D = 3 \times 10^4$.

5.3 Effects of context expansion

In this section we illustrate the effects of context expansion in improving the re-ranking performance in two aspects:



(a) (b)

Fig. 3. (a): Retrieval result comparison with increasing dimension D. (b): Retrieval result comparison with increasing top/bottom-H.

Methods		Oxford 5K	Paris 6K	Oxford 105K
Spatial Re-ranking		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 2. Computational cost comparison of spatial verification and our method, where D is the selected dimension (context) number. The results are measured by CPU second.

1) Query-relevant set (S) collection. This can be done offline [20] or online [15], as discussed in Section 3.2. We investigate the effects of various query-relevant set **S** collection methods by applying them to re-rank three retrieval systems: the baseline system (S1), spatial verification (S2), and average query expansion (AQE) (S3). Initially, we investigate the effects of various ways to collect **S** based on re-ranking the baseline system (S1). As seen in Table 3, the retrieval accuracy is 14.5% (9.5%) higher than S1 on the Oxford 5K (Paris 6K) dataset, when **S** is formed by offline expansion. In addition, online expansion is performed by including **S** as all words included in the spatially verified regions **S** (as done by AQE in [15]). 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 a small rise in mAP scores for S1. Similar to S1, offline expansion also enables an increase in retrieval accuracy when re-ranking the results returned by S2 and S3, while the online expansion methods lead to mAP scores close to the offline version but with more expensive computational cost during runtime. Therefore, we use the computationally cheaper offline expansion in the experiments.

2) Query-relevant (S) selection weight. We investigate the effects of query-relevant set (**S**) on the re-ranking results. This is done by enlarging weight b in the random mapping function (Eq. 2) while fixing weight $a = 10$ in Table 1:

Datasets	Retrieval system		Context expansion method		
	System ID	System baseline	Offline [20]	Online [15]	Offline + Online
Oxford5K	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
Paris6K	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 3. Illustration of the effects of various context expansion methods on the re-ranking results. In this process,the query-relevant sets **S** are collected by online or offline expansion, while the re-ranking is based on three kinds of retrieval system, namely: $S1$, Baseline [1]; $S2$, Spatial Re-ranking [1]; $S3$, AQE [15]. The offline expansion is computationally cheaper compared to the online expansion, while its performance is close to the online versions on all the three systems.

i) f_4: $b = \frac{a}{8}$; *ii)* f_5: $b = \frac{a}{4}$; *iii)* f_6: $b = \frac{a}{2}$; *iv)* f_7: $b = a$. The query-relevant words are collected by offline query expansion. Note that the parameters $a > 1$, $b > 1$ in Eq. 2 indicate that both the query and query-relevant words have priority to be selected. We set $b = \frac{a}{2}$ by default in the following experiments as it achieves the best performance on both datasets.

5.4 Evaluation of context re-ranking

In the previous sections, we evaluate various parameters when generating contexts. As expected, the context re-ranking parameters, also affect the re-ranking results, which are discussed as follows:

1) The range of top/bottom ranks. This is set by parameter H in Eq. 5. Fig. 3 (b) reports the retrieval accuracy with the increasing top/bottom-H. Intuitively, the range of top (bottom) ranks need to be relatively small compared to the dataset size so that it indicates whether a context is close to the query. As shown in Fig. 3 (b), we obtain stable retrieval accuracy when H exceeds a threshold, where $H = 200$. Thus, we set $H = 200$ as default.

2) The number of re-ranking iterations. Note that the re-ranking process is an updating scheme: the similarity score of each dataset image is refined according to the contextual information extracted from the ranking list. This process can be repeated such that each iteration generates re-ordered ranks, leading to updated contextual information. Fig. 4 reports the re-ranking accuracy as the iteration number grows, on the Oxford 5K and Paris 6K datasets. As seen in Fig. 4, retrieval accuracy is increased when the iteration number raises from 0 (baseline) to 3. During this process, the highest performance gain occurs at the first iteration. However, the accuracy begins to drop after several iterations. This is because random space partition usually includes noise due to the simplified clustering method. The noisy contextual information is accumulated within several iterations, thus decreasing the accuracy. Based on the above, we

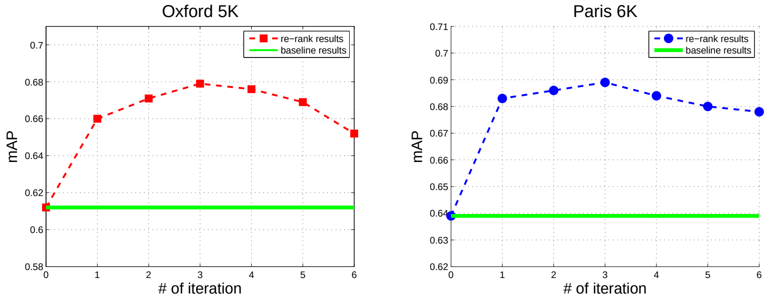


Fig. 4. Illustration of the effects of re-ranking iterations on the Oxford 5K and Paris 6K datasets. The iteration number raises from 0 (baseline) to 6, gradually. Best viewed in color.

Method	Oxford 5K		Paris 6K		Oxford 105K	
	Runtime	mAP	Runtime	mAP	Runtime	mAP
Spatial verification	2.10	0.645	4.71	0.653	4.34	0.571
Our method	0.039	0.701	0.043	0.700	0.48	0.585
AQE	2.21	0.796	4.85	0.769	6.01	0.767

Table 4. Comparison of the effectiveness and efficiency of various re-ranking methods on three datasets, measured by CPU second. Note that the runtime of AQE includes spatial verification and re-issuing query of the dataset. We only calculate the runtime of re-ranking, while do not include the CPU time spent on the initial baseline retrieval.

perform context based re-ranking once only in order to balance the efficiency and effectiveness.

5.5 Comparison to state-of-the-art

This section compares the accuracy and computation cost of our method to state-of-the-art.

1. Computational cost: As our method makes use of the inverted file structure, it requires no extra memory usage compared to the baseline tf-idf matching. Moreover, the runtime of our method consistently increases with D in terms of CPU time (Table 2) and accuracy (Fig. 3 (a)). By truanting the dimension number, our method balances effectiveness and efficiency. In addition, Table 4 compares our method to the spatial verification and AQE methods in terms of both accuracy and runtime. As seen in Table 4, our method outperforms the spatial verification method, while it is not as accurate as AQE. However, as AQE requires re-issuing the query from spatial verified results, our method is able to reduce the computational cost while still increasing the accuracy over the baseline and spatial verification methods.

2. Accuracy: Table 5 compares the accuracy of our method to state-of-the-art in three groups. Group A compares our method to some widely used spatial

Method		Oxford 5K	Paris 6K	Oxford 105K
Group A	Baseline	0.612	0.639	0.515
	WGC [4] (no prior)	0.621	0.644	0.574
	Spatial Re-ranking [1]	0.645	0.653	0.571
	Our method	0.701	0.700	0.585
	iSP [16]	0.741 [16]	0.769 [16]	0.649 [16]
Group B	QE baseline [15]	0.708	0.736	0.679
	AQE [15]	0.796	0.769	0.767
	DQE [11]	0.798	0.783	0.802
	Hello neighbor [13]	0.814 [13]	0.803 [13]	0.767 [13]
Group C	**Our method** + AQE [15]	0.814	0.770	0.757
	Our method + DQE [11]	0.832	0.793	0.790

Table 5. Retrieval performance comparison with state-of-the-art. Our method in this table is based on tf-idf similarity (S1). The results are all obtained from our implementation except those are taken from literatures. Note that our results are slightly different from the results reported in the original paper due to the repetition in implementation

re-ranking methods. Our method is ranked in the second place, although it is based on simply contextual re-ranking. As seen from Table 2 and Table 5, our method outperforms the standard spatial verification and is about 5 times faster. This is because our method uses less information to re-rank, *e.g.* it does not require the spatial consistency test applied to the features. Moreover, our method also outperforms the weak geometric consistency (WGC) method, which aims to verify the consistency between matching features without estimation of a full transformation [4]. Group B compares our method to various query expansion methods, such as AQE and DQE. As shown in Table 5, the accuracy of our method is below these query expansion methods. This is because we are aiming at efficient refinement of initial retrieval results without re-issuing the query as done by these query expansion methods. Compared to the query expansion methods, our method does not need online collection of query relevant visual words. The final group investigates the effect when our context re-ranking method is combined with other re-ranking methods, for example AQE and DQE. The results illustrate that our method can be combined with various query expansion methods, which leads to further improvement of retrieval performance.

6 Conclusion

In this paper, we proposed a simple yet effective re-ranking method for improving the BoW based object retrieval system. In contrast to the standard re-ranking methods, our method analyses the image ranks in terms of shared contextual information rather than expensive spatial consistency examination. We exploit contextual information in two steps. Firstly, we use a random space partition method to cluster the dataset into a large number of image groups. Secondly, the image groups, namely contexts, are used to refine the similarity scores of dataset images by considering their context factors. The experimental results show that our method can provide a significant accuracy boost with minimal computational cost. In future, we plan to test our method on non-rigid object retrieval, since unlike other re-ranking methods we do not rely on spatial rigidity.

References

1. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2007) 1–8
2. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: Proc. Int. Conf. Comp. Vis. (2003) 1470–1477
3. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2006) 2161–2168
4. Jégou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: Proc. Eur. Conf. Comp. Vis. (2008) 304–317
5. Jégou, H., Douze, M., Schmid, C.: Improving bag-of-features for large scale image search. Int. J. Comp. Vis. **87** (2010) 316–336
6. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Van Gool, L.: A comparison of affine region detectors. Int. J. Comp. Vis. **65** (2005) 43–72
7. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2008)
8. Mikulık, A., Perdoch, M., Chum, O., Matas, J.: Learning a Fine Vocabulary. In: Proc. Eur. Conf. Comp. Vis. (2010) 1–14
9. Jegou, H., Harzallah, H., Schmid, C.: A contextual dissimilarity measure for accurate and efficient image search. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2007) 1–8
10. Turcot, P., Lowe, D.: Better matching with fewer features: The selection of useful features in large database recognition problems. In: ICCV Workshop on Emergent Issues in Large Amounts of Visual Data. (2009) 2109–2116
11. Arandjelović, R., Zisserman, A.: Three things everyone should know to improve object retrieval. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2012)
12. Qin, D., Wengert, C., Van Gool, L.: Query adaptive similarity for large scale object retrieval. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn., IEEE (2013) 1610–1617
13. Qin, D., Gammeter, S., Bossard, L., Quack, T., van Gool, L.: Hello neighbor: accurate object retrieval with k-reciprocal nearest neighbors. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2011)
14. Fern, X.Z., Brodley, C.E.: Random projection for high dimensional data clustering: A cluster ensemble approach. In: Proc. Int. Conf. Mach. Learn. Volume 3. (2003) 186–193
15. Chum, O., Philbin, J., Sivic, J., Isard, M., Zisserman, A.: Total recall: Automatic query expansion with a generative feature model for object retrieval. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2007) 1–8
16. Chum, O., Mikulik, A., Perdoch, M., Matas, J.: Total recall ii: Query expansion revisited. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2011)
17. Pedronette, D.C.G., Torres, R.d.S.: Image re-ranking and rank aggregation based on similarity of ranked lists. In: Computer analysis of images and patterns, Springer (2011) 369–376

18. Chen, Y., Li, X., Dick, A., Hill, R.: Ranking consistency for image matching and object retrieval. Pattern Recognition **47** (2014) 1349–1360
19. Philbin, J., Sivic, J., Zisserman, A.: Geometric latent dirichlet allocation on a matching graph for large-scale image datasets. Int. J. Comp. Vis. **95** (2011) 138–153
20. Chen, Y., Dick, A., van den Hengel, A.: Image Retrieval with a Visual Thesaurus. In: 2010 International Conference on Digital Image Computing: Techniques and Applications. (2010) 8–14
21. Chum, O., Philbin, J., Zisserman, A.: Near duplicate image detection: min-hash and tf-idf weighting. In: Proc. British Machine Vision Conference. Volume 3. (2008) 4
22. Bowman, A.W., Azzalini, A.: Applied smoothing techniques for data analysis. Oxford University Press (1997)
23. http://www.robots.ox.ac.uk/∼vgg/data/
24. http://press.liacs.nl/mirflickr/dlform.php
25. http://www.robots.ox.ac.uk/∼vgg/software/fastcluster/