# Multiple queries for large scale specific object retrieval

Relja Arandjelović
relja@robots.ox.ac.uk
Andrew Zisserman
az@robots.ox.ac.uk

Department of Engineering Science
University of Oxford
Parks Road
Oxford, OX1 3PJ, UK

**Abstract**

The aim of large scale specific-object image retrieval systems is to instantaneously find images that contain the query object in the image database. Current systems, for example Google Goggles, concentrate on querying using a single view of an object, e.g. a photo a user takes with his mobile phone, in order to answer the question "what is this?". Here we consider the somewhat converse problem of finding *all* images of an object given that the user knows what he is looking for; so the input modality is text, not an image. This problem is useful in a number of settings, for example media production teams are interested in searching internal databases for images or video footage to accompany news reports and newspaper articles.

Given a textual query (e.g. "coca cola bottle"), our approach is to first obtain multiple images of the queried object using textual Google image search. These images are then used to visually query the target database to discover images containing the object of interest. We compare a number of different methods for combining the multiple query images, including discriminative learning. We show that issuing multiple queries significantly improves recall and enables the system to find quite challenging occurrences of the queried object.

The system is evaluated quantitatively on the standard Oxford Buildings benchmark dataset where it achieves very high retrieval performance, and also qualitatively on the TrecVid 2011 known-item search dataset.

## 1 Introduction

The objective of this paper is to retrieve all images containing a *specific* object in a large scale image dataset. This is a problem that has seen much progress and success over the last decade, with the caveat that the starting point for the search has been a single query image of the specific object of interest [2, 5, 11, 17, 18, 21, 24, 26]. In this work we make two changes to the standard approach: first, our starting point for specifying the object is text, as we are interested in probing data sets to find known objects; and second, and more importantly for the development of novel algorithms, we search the dataset using multiple image queries and collate the results into a single ranked list.

It is important to first consider why images containing the target object are currently missed. Addressing this problem has been one of the main research themes in specific object retrieval research with developments in feature encoding to alleviate vector quantization

(VQ) losses [10, 11, 12, 13, 14, 17, 22, 29], and in augmentation of the bag of visual word (BoW) representation to alleviate detector and descriptor drop out (as well as, again, VQ losses) [2, 5, 6, 28].

The limitation of current augmentation approaches, which are based on query expansion (QE) within the data set, is that they rely on the query to yield a sufficient number of high precision results in the first place. In more detail, in QE an initial query is issued, using only the query image, and confident matches, obtained by spatial verification, are used to re-query. There are three problems with this approach: firstly, it is impossible to gain from QE if the initial query fails. Secondly, if the dataset does not contain many images of the queried object QE cannot boost performance. Finally, it is not possible to obtain images from different views of the object as these are never retrieved using the initial query, for example querying using an image of a building façade will never yield results of its interior.

More generally current BoW retrieval systems miss images that differ too much from the query in aspect (side vs front of a building), age (antiquarian photos may be missed if too much has changed between the target image and query), weather conditions, extreme scale changes, etc. Using multiple images of the object to query the database naturally alleviates to some extent all of these problems.

One of the principal contributions of this paper is an algorithm to overcome these current shortcomings by combining multiple queries in a principled manner (section 2). The other principal contribution is the implementation of a real time demonstration system which generates query images automatically starting from text using Google image search (section 3.3).

**Related work.** In content-based image retrieval (CBIR) for categories (but not for specific objects) it is quite common to use a set of images to represent a query specified by text. A standard method is to obtain a set of images from a labelled corpus corresponding to that query [9] or training images from a web search [8, 27]. Other standard approaches in CBIR can also result in a set of images representing the query: in relevance feedback the user selects from a set of images proposed from the target corpus, e.g. in the PicHunter system [7]; in query expansion the original text query can be enhanced (e.g. by synonyms) and thereby result in multiple queries; one form of query expansion is to simply issue new queries using high ranked images from an initial search, a form of blind relevance feedback.

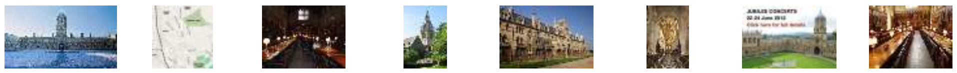
Many methods for combining (or fusing) ranked lists have been developed, these can either use only the rank of the items in the list (e.g. Borda count [3]), or the score as well if this is available [25].

## 2 Retrieval using multiple query images

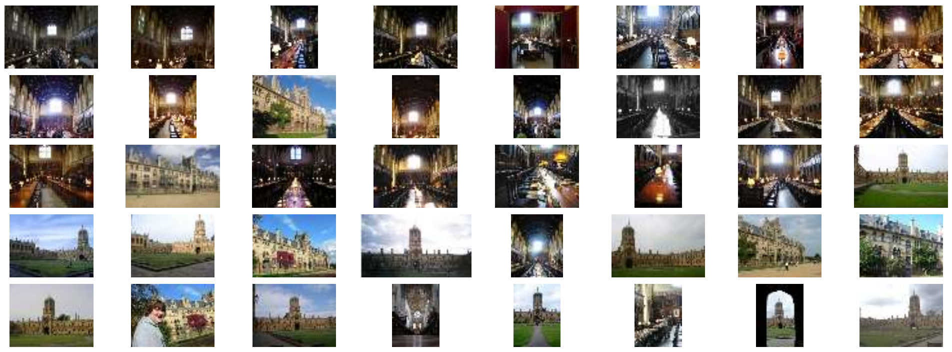
A question arises as to how to use multiple query images (the query set), as current systems only issue a single query at a time. We propose five methods for doing this; methods (i) and (ii) use the query set jointly to issue a single query, while methods (iii)-(v) issue a query for each image in the query set and combine the retrieved results. The five methods are described next.

### 2.1 Retrieval methods

**(i) Average query (Joint-Avg).** Similar to the average query expansion method of [5], the bag-of-words representations of all images in the query set are averaged together. The



(a) Top 8 Google Image results for the textual query "Christ Church, Oxford"



(b) Top 40 retrieved results from the Oxford 5k dataset for the query "Christ Church, Oxford"

Figure 1: **Multiple query retrieval.** Images downloaded from Google using the "Christ Church, Oxford" textual query (a) are used to retrieve images of Christ Church college in the Oxford Buildings dataset (b). All the top 40 results of (b) do show various images of Christ Church (the dining hall, tourist entrance, cathedral and Tom tower). This illustrates the benefit of issuing multiple queries in order to retrieve all images of the queried object. Note that the noise in images retrieved from Google (the second image in (a) shows a map of Oxford) did not affect retrieval.

average BoW vector is used to query the database by ranking images based on the tf-idf score.

**(ii) SVM over all queries (Joint-SVM).** Similar to the discriminative query expansion method of [2], a linear SVM is used to discriminatively learn a weight vector for visual words online. The query set BoWs are used as positive training data, and BoWs of a random set of 200 database images form the negative training data. The weight vector is then used to efficiently rank all images in the database.

**(iii) Maximum of multiple queries (MQ-Max).** A query is issued for each BoW vector in the query set independently and retrieved ranked lists are combined by scoring each image by the maximum of the individual scores obtained from each query.

**(iv) Average of multiple queries (MQ-Avg).** Similar to (iii) but the ranked lists are combined by scoring each image by the average of the individual scores obtained from each query.

**(v) Exemplar SVM (MQ-ESVM).** Originally used for classification [15], this method trains a separate linear SVM for each positive example. The score for each image is computed as the maximal score obtained from the SVMs.

### 2.2 Spatial reranking

Precision of a retrieval system can be improved by reranking images based on their spatial consistency [21, 26] with the query. Since spatial consistency estimation is computationally relatively costly, only a short-list of top ranked results is reranked. We use the spatial reranking method of Philbin *et al*. [21] which reranks images based on the number of visual words consistent with an affine transformation (inliers) between the query and the database image.

Here we explain how to perform spatial reranking when multiple queries are used. For fair comparison of different methods it is important to fix the total number of spatial transformation estimations, we fix it to $R = 200$ per image in the query set of size $N$.

For methods *Joint-Avg* and *Joint-SVM* which perform a single query each, reranking is performed on the top $R$ results. Images are ranked based on the average number of inliers across images in the query set. The number of spatial transformation estimations is thus $N \times R$.

For methods *MQ-Max*, *MQ-Avg* and *MQ-ESVM* which issue $N$ queries each, reranking is performed for each query independently before combining the retrieved lists. For a particular query (one of $N$), reranking is done on the top $R$ results using only the queried image. The number of spatial transformation estimations is thus, again, $N \times R$.

## 3 Implementation description

### 3.1 Standard BoW retrieval system

We have implemented the standard framework of Philbin *et al*. [21] with some recent improvements that are discussed next. RootSIFT [2] descriptors are extracted from the affine-Hessian interest points, we use the recent implementation of the affine-Hessian feature detector [16] by Perd'och *et al*. [1, 20] as it was shown to yield superior retrieval results. The descriptors are quantized into 1M visual words obtained using approximate k-means. Given a single query, the system ranks images based on the *term frequency inverse document frequency* (tf-idf) score [26]. Spatial reranking is performed on the top 200 tf-idf results using an affine transformation [21] as described above.

### 3.2 Implementation details for multiple query method

Here we give implementation details for the proposed methods (section 2). For the discriminative approaches (*Joint-SVM* and *MQ-ESVM* methods), the query set forms the positive training examples, while the negative set comprises 200 random database images. For training of a linear SVM classifier we use LIBSVM [4]. The learnt weight vector is used to efficiently rank all images in the database based on their signed distance from the decision boundary. This can be done efficiently using the inverted index in the same way as when computing the tf-idf score, as both operations correspond to computing the scalar product between a weight vector and the BoW histograms of the database images. In order for retrieval to be fast, the learnt weight vector should be sparse. To ensure this we use the same approach as in [2], namely, the BoW vectors of negative images are truncated (and renormalized) to only include words that appear in at least one positive example.

For the *MQ-ESVM* case, as in [15], scores of individual SVMs have to be calibrated so that they can be compared with each other. This is done by fitting a sigmoid function to the output of each SVM individually [23], to try to map scores to 0 and 1 for negatives and positives, respectively. For the negative data required for calibration we use a set of 200 random images (different from the one used in exemplar SVM training), while for calibration positives we use the spatially verified positives for the given query. Note that it is not possible to evaluate *MQ-ESVM* without spatial reranking, as spatial transformations need to be estimated for the calibration procedure.

### 3.3 Building a real-time system

We have built a system which can respond to user text queries in real-time. After a user enters the query text, a textual Google image search is performed using the publicly available API provided by Google. Each of the top retrieved results, we use eight, is processed independently in a separate thread – the image is downloaded and a bag-of-visual-words description is obtained as discussed in section 3.1. Then, the processed query set is used to present the user with a ranked list of results obtained by using one of the methods introduced in section 2. Note that the methods which issue multiple queries and then merge the retrieved results (*MQ-*) can be easily parallelized as each query can be executed in an independent thread.

The entire process from typing words to retrieving relevant images takes less than 10 seconds. The bottle-neck is the Google API call which can take up to 3 seconds, along with downloading images from their locations on the internet. The actual querying, once the query set BoWs are computed, takes a fraction of a second.

## 4 Evaluation and Results

In this section we assess the retrieval performance of our multiple query methods by comparing them to a standard single query system, and compare them to each other.

### 4.1 Datasets and evaluation procedure

The retrieval performance of proposed methods is evaluated using standard and publicly available image and video datasets, we briefly describe them here.

**Oxford Buildings [21].** This dataset contains 5062 high-resolution images automatically downloaded from Flickr. It defines 55 queries (consisting of an image and query region of interest) used for evaluation (5 for each of the 11 chosen Oxford landmarks) and it is quite challenging due to substantial variations in scale, viewpoint and lighting conditions. The basic dataset, often referred to as *Oxford 5k*, is usually appended with another 100k Flickr images to test large scale retrieval, thus forming *Oxford 105k* dataset. Retrieval performance is measured in terms of mean average precision (mAP).

The standard evaluation protocol needs to be modified for our task as it was originally set up to evaluate single-query methods. We perform 11 queries, one per each predefined landmark; the performance is still measured using mAP.

Our methods are evaluated in two modes of operation depending on the source of the query set: one using the five predefined queries per landmark (Oxford queries, OQ), and one using the top 8 Google image search results for the landmark names (Google queries, GQ), chosen by the user to make sure the images contain the object of interest. The images in the Oxford building dataset were obtained by crawling Flickr, so we append a “-flickr” flag to the textual Google image search in order to avoid downloading exactly the images from the Oxford dataset which would artificially boost our performance.

**TrecVid 2011.** This dataset contains 211k keyframes extracted from 200 hours of low resolution footage used in the TrecVid 2011 known-item search challenge [19] (the IACC.1.B dataset). As there is no ground truth available for this dataset we only use it to assess the retrieval performance qualitatively.

### 4.2 Baselines

Due to the lack of multiple query methods, comparison is only possible to methods which use a single image to query. For the Oxford queries (OQ) case the queries are the 55 predefined ones for the dataset. The two proposed baselines use exactly the same descriptors and vocabulary as our multiple query methods.

**Single query.** A natural baseline to compare to is the system of Philbin *et al*. as described in [21] with extensions of section 3.1. For the Google queries (GQ) case the query is the top Google image result which contains the object of interest.

**Best single query.** The *single query* method is used to rank images using each query from the query set (the same query sets are used as for our multiple query methods) and the best performing query is kept. This method cannot be used in a real-world system as it requires an oracle (i.e. looks up ground truth).

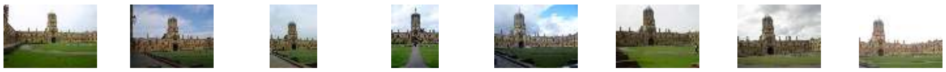
### 4.3 Results and discussion

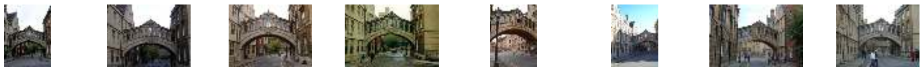
Figure 2 shows a few examples of textual queries and the retrieved results. Note the ability of the system to retrieve specific objects (e.g. the Tom Tower of Christ Church college in figure 2(a)) as well as sets of relevant objects (e.g. different parts of Christ Church college in figure 1) without explicitly determining the specific/general mode of operation.

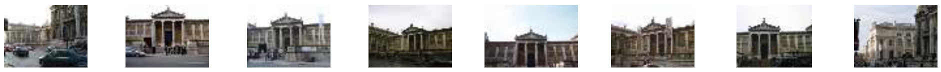
Table 1 shows the retrieval performance on the Oxford 105k dataset. It can be seen that all the multiple query methods are superior to the "single query" baseline, improving the performance by 29% and 52% for the Oxford queries and Google queries (with spatial reranking), respectively. It is clear that using multiple queries is indeed very beneficial as the best performance using Oxford queries (0.937) is better than the best reported result using a single query (0.891 achieved by [2]); it is even better than the state-of-the-art on a much easier Oxford 5k dataset ([2]: 0.929). All the multiple query methods also beat the "best single query" method which uses ground truth to determine which one of the images from the query set is best to be used to issue a single-query.

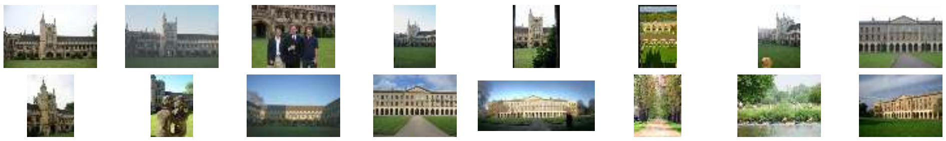
From the quantitative evaluation it is clear that multiple query methods are very beneficial for achieving higher recall of images containing the queried object, however it is not yet clear which of the five proposed methods should be used as all of them perform very well on the Oxford 105k benchmark. Thus, we next analyse the performance of various methods qualitatively on the TrecVid 2011 dataset, and show three representative queries and their outputs in figure 3.

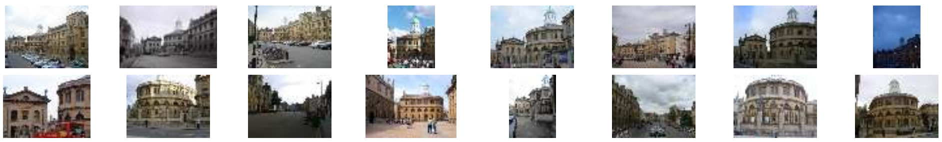
The clear winner is the *MQ-Max* method – this is because taking the maximal score of the retrieved lists enables it to rank an image highly based on a strong match with a single


(a) Tom Tower, Christ Church, Oxford


(b) Bridge of Sighs, Oxford


(c) Ashmolean Museum, Oxford


(d) Magdalen College, Oxford


(e) Broad Street, Oxford

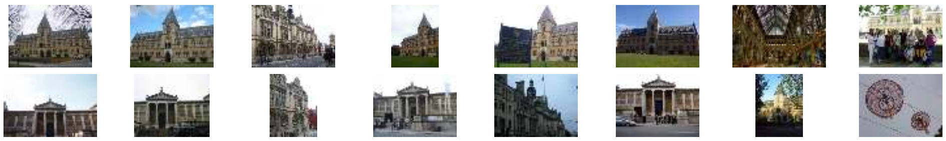
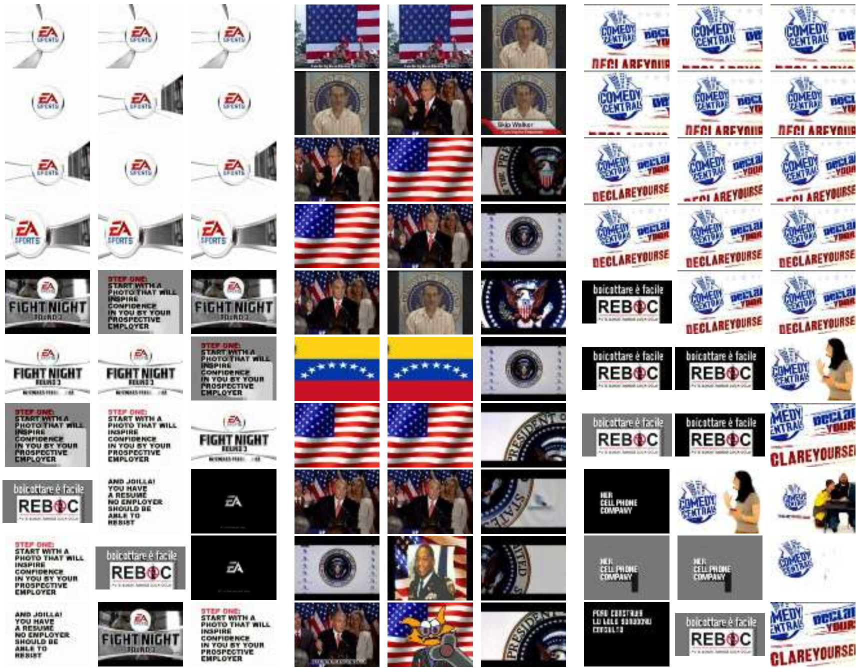

(f) Museum, Oxford

Figure 2: **Query terms and top retrieved images from the Oxford 5k dataset.** The captions show the textual queries used to download images from Google to form the query set. The top 8 images were used, without any user feedback to select the relevant one; the results are generated with the *MQ-Max* method. Specific (a-c) and broad (d-f, figure 1) queries are automatically handled without special considerations; note that (a) is a more specific version of the query in figure 1. (f) searching for "Museum, Oxford", which is a broader query than (c), yields in the top 16 results photos of three Oxford museums and a photo from the interior of one of them.

query image from the query set. The other two methods which average the scores down-weight potential challenging examples even if they match very well with one query image, thus only retrieving "canonical" views of an object. For example, all methods work well for the "EA sports logo" query (figure 3(a)) and retrieve the common appearances of the object (represented in 7 out of 8 images in the query set). However, only the *MQ-Max* method manages to find the extra two "unusual" and challenging examples of the logo in silver on a black background.

| | Google queries (GQ) | | Oxford queries (OQ) | |
|---|---|---|---|---|
| | Without SR | With SR | Without SR | With SR |
| Single query | 0.464 | 0.575 | 0.622 | 0.725 |
| Best single query (“cheating”) | 0.720 | 0.792 | 0.791 | 0.864 |
| Joint-Avg | 0.834 | 0.873 | 0.886 | 0.933 |
| Joint-SVM | 0.839 | 0.875 | 0.886 | 0.926 |
| MQ-Max | 0.746 | 0.850 | 0.826 | 0.929 |
| MQ-Avg | 0.834 | 0.868 | 0.888 | 0.937 |
| MQ-ESVM | N/A | 0.846 | N/A | 0.922 |

Table 1: **Retrieval performance (mAP) of the proposed methods on the Oxford 105k dataset.** SR stands for spatial reranking. The “Oxford queries” (OQ) and “Google queries” (GQ) columns indicate the source of query images, the former being the 5 predefined query images and the latter being the top 8 Google images which contain the queried object. The details of the evaluation procedure, baselines and proposed methods are given in sections 4.1, 4.2 and 2, respectively. All proposed methods significantly outperform the “single query” baseline, as well as the artificially boosted “best single query” baseline.



(a) EA sports logo (b) Presidential seal (c) Comedy central logo

Figure 3: **Multiple query retrieval on TrecVid 2011 dataset.** (a)-(c) show three different textual queries and retrieval results. Within one example, each column shows a ranked list of images (sorted from top to bottom) for a particular method. Left, middle and right columns show *Joint-SVM*, *MQ-Avg* and *MQ-Max* methods, respectively. *MQ-Max* is clearly the superior method.

It is also interesting to compare *MQ-Avg* with *Joint-SVM* in order to understand whether it is better to issue multiple queries and then merge the resulting ranked lists (the *MQ-* approaches), or to have a joint representation of the query set and perform a single query (the *Joint-* approaches). Figure 3 shows that the "multiple queries" approach clearly performs better. The argument for this is similar to the arguments we made in favour of the *MQ-Max* method, namely that it is beneficial to be able find close matches to each individual query image. Furthermore, we believe that the spatial reranking procedure (section 2.2) of the *MQ-* methods is more efficient – estimation of a spatial transformation between a query image and a short-list is conducted on the short-list obtained from the corresponding query image, while for the *Joint-* methods, where only a single "global" short-list is available, many attempts at spatial verification are wasted on using irrelevant query images. Another positive aspect of the "multiple queries" methods is that they can be parallelized very easily – each query is independent and can be handled in a separate parallel thread.

We note that the discriminative methods perform slightly better than the corresponding non-discriminative ones, i.e. *Joint-SVM* and *MQ-ESVM* outperform *Joint-Avg* and *MQ-Max*, respectively. However, the difference in our examples was not significant, so due to ease of implementation we recommend the use of the non-discriminative methods.

Finally, taking all aspects into consideration, we conclude that the method of choice for multiple query retrieval is *MQ-Max*, where each image from the query set is queried on independently and max-pooling is applied to the retrieved sets of results.

## 5 Conclusions

We have investigated a number of methods for using multiple query images and find that approaches that issue multiple independent queries and combine the results outperform those that jointly model the query set and issue a single query. Of the multiple independent query methods *MQ-Max* was found to perform best in terms of retrieving the more unusual instances.

Also, we have built a system which can, in real-time, retrieve images containing a specific object from a large image database starting from a text query. Using Google image search (or Bing or Flickr image search etc) in this way to obtain sample query images opens up a very flexible way to immediately explore unannotated image datasets.

**Acknowledgements** We are grateful for financial support from ERC grant VisRec no. 228180 and EU Project FP7 AXES ICT-269980.

## References

[1] http://cmp.felk.cvut.cz/∼perdom1/code/index.html.

[2] R. Arandjelović and A. Zisserman. Three things everyone should know to improve object retrieval. In *Proc. CVPR*, 2012.

[3] J. Aslam and M. Montague. Models for metasearch. In *Proc. SIGIR*, pages 276–284, 2001.

[4] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM TIST*, 2:27:1–27:27, 2011.

[5] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *Proc. ICCV*, 2007.

[6] O. Chum, A. Mikulik, M. Perd'och, and J. Matas. Total recall II: Query expansion revisited. In *Proc. CVPR*, 2011.

[7] I. J. Cox, M. Miller, T. Minka, T. Papathomas, and P. N. Yianilos. The bayesian image retrieval system, PicHunter: Theory, implementation and psychophysical experiments. *IEEE Transactions on Image Processing*, 2000.

[8] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from Google's image search. In *Proc. ICCV*, 2005.

[9] K. A. Heller and Z. Ghahramani. A simple bayesian framework for content-based image retrieval. In *Proc. CVPR*, 2006.

[10] M. Jain, H. Jégou, and P. Gros. Asymmetric hamming embedding. In *ACM Multimedia*, 2011.

[11] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *Proc. ECCV*, 2008.

[12] H. Jégou, M. Douze, and C. Schmid. Improving bag-of-features for large scale image search. *IJCV*, 87(3):316–336, 2010.

[13] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *Proc. CVPR*, 2010.

[14] A. Makadia. Feature tracking for wide-baseline image retrieval. In *Proc. ECCV*, 2010.

[15] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-SVMs for object detection and beyond. In *Proc. ICCV*, 2011.

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

[17] A. Mikulik, M. Perd'och, O. Chum, and J. Matas. Learning a fine vocabulary. In *Proc. ECCV*, 2010.

[18] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Proc. CVPR*, 2006.

[19] O. Paul, G. Awad, M. Michel, J. Fiscus, W. Kraaij, A. F. Smeaton, and G. Quéenot. Trecvid 2011 – an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *In Proc. TRECVID 2011*, 2011.

[20] M. Perd'och, O. Chum, and J. Matas. Efficient representation of local geometry for large scale object retrieval. In *Proc. CVPR*, 2009.

[21] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proc. CVPR*, 2007.

[22] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Proc. CVPR*, 2008.

[23] J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, 1999.

[24] D. Qin, S. Gammeter, L. Bossard, T. Quack, and L. Van Gool. Hello neighbor: accurate object retrieval with k-reciprocal nearest neighbors. In *Proc. CVPR*, 2011.

[25] J. A. Shaw and E. A. Fox. Combination of multiple searches. In *The Second Text REtrieval Conference (TREC-2)*, pages 243–252, 1994.

[26] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. ICCV*, volume 2, pages 1470–1477, 2003.

[27] L. Torresani, M. Szummer, and A. Fitzgibbon. Efficient object category recognition using classemes. In *Proc. ECCV*, pages 776–789, 2010.

[28] T. Turcot and D. G. Lowe. 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 (WS-LAVD)*, 2009.

[29] J. C. van Gemert, J. M. Geusebroek, C. J. Veenman, and A. W. M. Smeulders. Kernel codebooks for scene categorization. In *Proc. ECCV*, 2008.