Better matching with fewer features: The selection of useful features in large database recognition problems

Panu Turcot and David G. Lowe
University of British Columbia
Vancouver, Canada
pjturcot,lowe@cs.ubc.ca

Abstract

There has been recent progress on the problem of recognizing specific objects in very large datasets. The most common approach has been based on the bag-of-words (BOW) method, in which local image features are clustered into visual words. This can provide significant savings in memory compared to storing and matching each feature independently. In this paper we take an additional step to reducing memory requirements by selecting only a small subset of the training features to use for recognition. This is based on the observation that many local features are unreliable or represent irrelevant clutter. We are able to select "useful" features, which are both robust and distinctive, by an unsupervised preprocessing step that identifies correctly matching features among the training images. We demonstrate that this selection approach allows an average of 4% of the original features per image to provide matching performance that is as accurate as the full set. In addition, we employ a graph to represent the matching relationships between images. Doing so enables us to effectively augment the feature set for each image through merging of useful features of neighboring images. We demonstrate adjacent and 2-adjacent augmentation, both of which give a substantial boost in performance.

(a) All image features (b) Useful image features

Figure 1. Original image features (a) and those deemed to be useful (b). Transient objects in the foreground and non-distinctive areas of the scenes are found to be without useful features.

1. Introduction

Large database image recognition refers to the task of correctly matching a query image to an image of the same object selected from a large database. In this context *large* refers to image sets where the amount of data exceeds what can be stored in available memory. Conventional approaches which store individual local descriptors for each image [9] are no longer suitable as the number of images rises into the millions or higher.

One solution to this problem was proposed by Sivic and Zisserman [20], in which image descriptors are quantized into "visual words." Quantized matching is performed using a bag-of-words (BOW) method, in which visual word occurrences alone are used to measure image similarity. Their approach employs a term-frequency inverse-document-frequency (*tf-idf*) weighting scheme similar to that used in text retrieval.

In current BOW methods, *all* descriptors from the initial

image set are quantized and discarded while their geometric data are preserved for later matching. Quantization significantly reduces the storage requirements for features as invariant descriptors do not need to be retained, but can be summarized by a single cluster center for all features in a visual word. However, other information must still be retained for each feature, such as its source image ID, as well as location, scale, and orientation within that image for final geometric checking. In practice, the use of visual words provides, at most, a one order of magnitude reduction in memory usage regardless of the number of features within each visual word.

In this paper, we present a method to further reduce the amount of information stored from each image, while still maintaining strong recognition performance, through the preservation of only a minimal set of image features, which we refer to as *useful* features.

We define a useful feature to be an image feature which has proven to be robust enough to be matched with a corresponding feature in the same object, stable enough to exist in multiple viewpoints, and distinctive enough that the corresponding features are assigned to the same visual word.

Our method builds on that of Philbin *et al.* [15], employing a BOW framework and *tf-idf* ranking. Image descriptors are first extracted and quantized into visual words. These are used to match database images against one another using *tf-idf* ranking. The best *tf-idf* matches are geometrically validated using a method similar to that employed by Chum [5] in which an initial affine model is verified using epipolar geometry. Once validated, geometrically consistent descriptors are labeled and retained while all other descriptors are discarded. Validated image matches are stored in the form of an image adjacency graph where matched image pairs are joined by an edge.

In our experiments, testing is conducted on the Oxford Buildings dataset using a cross validation procedure. Our results show that using only useful features eliminates 96% of image descriptors while maintaining recognition performance. Using image adjacency relationships, we achieve significantly improved recognition performance without necessitating the storage of any additional features.

This paper presents our method for extracting and using useful features as follows. Section 2 outlines previous work done in the field of large database image matching. Section 3 presents the BOW framework used in our method used to generate our initial matches. Useful feature extraction and geometric validation is discussed in section 4. Section 5 introduces the image adjacency graph as well as a new ranking method making use of adjacency relationships. The evaluation procedure is provided in section 6 and results presented in section 7.

2. Previous work

In recent years, many methods have been published making using of a quantized descriptor space and a BOW framework to perform image ranking on large sets of images [7, 8, 14, 15, 16]. Though the methods vary, these recognition systems can all be broken down into the following steps: feature extraction, feature quantization, image ranking, and geometric re-ranking.

Feature extraction from the image is a topic that has been widely researched, with many interest point detectors [12] and descriptors [11] in use.

Feature quantization makes use of clustering to quantize the descriptor space into visual words, which together make up a visual vocabulary. This clustering and word assignment has been conducted using hierarchical k-means [14] as well as approximate flat k-means [8, 15]. Increasing the visual vocabulary size to 1M cluster centers allowed for improvements in recognition on datasets as large as 1M images. Furthermore, it has been shown that the visual vocabulary used can impact the recognition performance of the overall system. Forming the visual vocabulary using a sample set of images effectively trains the BOW to discriminate descriptors from the samples, while using a different set of images results in reduced recognition [8].

Once quantized, matching is performed using methods which borrow heavily from document retrieval. Using a standard term-frequency inverse-document-frequency (*tf-idf*) weighting scheme [4] has been shown to yield good recognition performance [14, 15]. This weighting scheme can be interpreted as an approximation to a K-NN voting algorithm with *tf-idf* weights [8].

The BOW search produces a ranked list of images which can subsequently be re-ranked using geometric information associated with image descriptors. As this is a more computationally expensive step, only a small subset of the images will be candidates for re-ranking, requiring that the initial BOW recognition performance be as precise as possible.

In order to boost recognition performance, novel additions to this base framework have been introduced. In [7], the concept of query expansion is used in which strong image matches are used to generate additional queries. Doing so allows a single query image to undergo several iterations of the image ranking and geometry checking stages before producing a final result. Other improvements include the use of soft word assignment [16] which makes use of multiple word assignment to reduce quantization error in large vocabularies, as well as hamming-embedding [8] in which location within a visual word is encoded to allow for improved accuracy using smaller vocabularies.

In all these methods, word assignment and geometric information associated with every image feature is preserved for possible use in geometric re-ranking.

Recent research [5, 17] has explored the construction and

use of graphs in large image collections. Our approach differs from these by using the image graph for feature selection. In addition, we demonstrate the value of image graphs for augmenting the matched features for each image at query time through use of the adjacency relationships resulting in significant recognition performance improvements.

Previous research showed the value of selecting informative features in a BOW framework [19]. Their approach used ground truth image locations which differs from our unsupervised approach. Our method also incorporates a geometric verification phase that identifies individual correct matches between images.

3. Bag-of-words matching

Our bag-of-words framework consists of a group of cluster centers, referred to as visual words $W = \{w_1, w_2, ..., w_k\}$.

Given a new image I, image descriptors $\{d_1, d_2, d_3...\}$ are extracted. Assignment of descriptors to visual words is performed using a nearest word search:

$$d \rightarrow w = \arg\min_{w} dist(d, w) \tag{1}$$

As the number of visual words increases, visual word assignment can become a computational bottleneck. In such cases, we use approximate nearest word search [13], which provides significant speedup over linear search while maintaining high accuracy.

Following visual word assignment the original image descriptors are discarded, yielding memory savings over conventional image matching methods. A record of word occurrences from each image is kept, and as the name *bag-of-words* would suggest, only this record of visual word occurrences is used for initial querying.

While not used in the initial BOW matching process, geometric information associated with image descriptors can be used on a limited subset of candidate images in a secondary re-ranking of initial query results. The set of cluster centers, image word occurrences and descriptor geometric information form our BOW image database.

3.1. Querying an image database

Images used to query the database follow the same word assignment process and are converted into visual word occurrences. In order to compare image word occurrence histograms, word occurrences are converted to *tf-idf* weights, x_{ij}:

$$x_{ij} = \underbrace{\frac{n_{ij}}{\sum_i n_{ij}}}_{\text{tf}_{ij}} \underbrace{\log \frac{N}{\sum_j |n_{ij} > 0|}}_{\text{idf}_i} \tag{2}$$

where n_{ij} is the number of occurrences of word i in image j and N is the total number of images in the image database. In the IDF term $\sum_j |n_{ij} > 0|$ denotes the number of images in which word i is present.

Tf-idf weights are used in a vector space model, where query and database images I are represented by a vector made up of *tf-idf* weights $I_j = [x_{1j}, x_{2j}, x_{3j}, ..., x_{kj}]$ which is then normalized to unit length. Similarity between images is calculated using the *L2* distance metric or cosine similarity, which are equivalent for length-normalized vectors.

As images contain often repeated and therefore uninformative descriptors, a stop list of the most common words was generated and those words suppressed, a technique shown to be effective at improving recognition performance [20].

4. Useful features

When generating features from an image, many image features which are not useful are extracted. These include features generated around unstable interest points, and feature descriptors that are uninformative and occur frequently in many images. They could also include features of transient occlusions, such as a person or vehicle in the foreground.

Rejection of useless features is motivated by the fact that occlusions and unstable object features will likely exist in only a single image, while useful features are likely to be found in more than one image of the same object or location. Identification of the features that are robust to change of view can be performed by determining which features exist in multiple views and are geometrically consistent with one another. While doing so requires that at least two views of a given object or location exist in the image database prior to useful feature extraction, for most large datasets this condition will normally be met. We discuss the special case of singleton images below.

In large database image matching applications, it is assumed that images may not be labeled. Therefore, our useful feature detection is fully unsupervised.

4.1. Implementation

In order to determine which image features are useful, a BOW image database containing the full feature set is constructed. Following construction, each image in the database is used as a query. The best M images are each geometrically verified, and only features which are geometrically consistent are preserved.

Initial geometry checking is performed by using RANSAC to estimate affine transform parameters between images. Following this initial check, inliers are then used to estimate epipolar geometry using a variation of the LO-

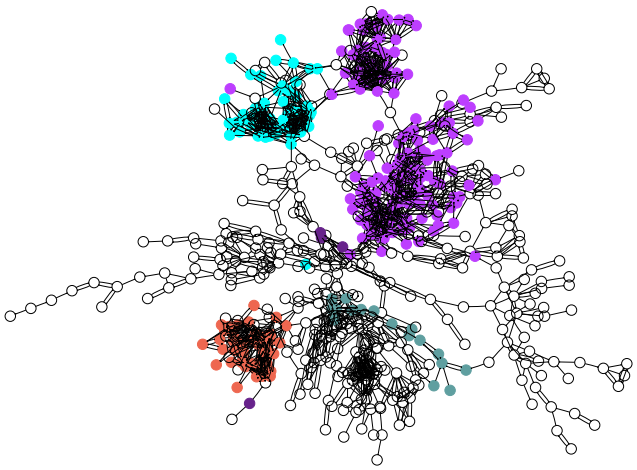


Figure 2. An image graph showing connectivity between database images. This is one connected subgraph from the full Oxford Buildings dataset. Ground truthed images are coloured by building. Images of the same location naturally form highly connected regions in the graph without any prior knowledge about image content. Graph generated using GraphViz[1]

RANSAC algorithm [6]. This two stage process is needed to improve the sometimes poor initial feature correspondence reliability, a result of the many-to-many BOW matching process.

4.2. Singleton images

Images without any geometrically valid matches are considered singleton images. In the context of large database recognition, where items being searched are expected to be common enough that they will appear in multiple images, singleton images can be safely discarded allowing for further memory savings.

In applications where isolated single views of an object may be important, it will become necessary to preserve some features from singleton images. To avoid the memory requirements of preserving all features, a subset of the largest-scale image features in each image can be kept. This is equivalent to preserving low resolution copies of singleton images, with the resolution chosen to achieve a target number of features for each image.

It would also be possible to select a subset of singleton image features based on other criteria, such as keeping features that belong to visual words with high information gain [19], or selecting features which are robust to affine or other distortions of the image [18]. These are topics we intend to examine in future research.

5. Image adjacency

In addition to filtering out uninformative descriptors, useful feature extraction provides information about the relationships between images in the database. We introduce the concept of image adjacency, in which two images that match following geometric verification are said to be adjacent.

To represent these relationships between database images, a graph $G = (V, E)$ is constructed during the useful feature extraction process such that each vertex $v \in V$ represents an image and each edge $e = (v_A, v_B) \in E$ represents a geometrically verified match.

A visualization of the image adjacency graph shows the relationships between database images (Figure 2). Even though the image graph construction is unsupervised, images of the same building naturally group together and form interconnected clusters.

An overview of useful feature extraction and image graph construction is presented in Algorithm 1.

Data: BOW database
Result: Image graph $G = (V, E)$, labeled useful features
foreach *image I in the database* **do**
  G.`addVertex`(v_I)
  Query the database using image I
  $R \leftarrow$ list of database images sorted by rank
  for $i = 1 \rightarrow M$ **do**
    $d =$ `validatedFeatures`(I, R_i)
    if $|d| >$ *numPointThresh* **then**
      G.`addEdge`(v_I, v_J)
      `labelAsUseful`(d)
    end
  end
end

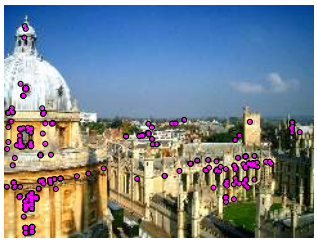
Algorithm 1: Useful feature extraction

5.1. Image augmentation

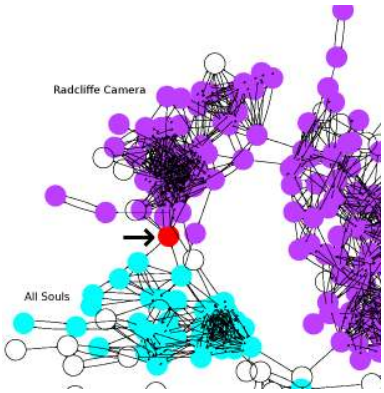
The construction of the image graph allows for improvements to the BOW image matching. Since adjacent images are geometrically verified and are assumed to contain the same object of interest, we can assume adjacent images represent nearby viewpoints of the same object. We present a method for integrating multiple viewpoints together, also referred to as view clustering [10], which allows images with similar views to share features.

For every image I in our image database, referred to as the base image, we represent the image not only with its own descriptors, but also effectively include the descriptors of every adjacent image in the image graph. In our BOW framework, this simple variation on view clustering can be implemented by adding word occurrences of all adjacent images to those of the base image:

(a) All image features



(b) Useful image features



(c) Image graph

Figure 3. All image features (a) and useful features (b) for a selected image. Note that this image retains features from both All Souls college and Radcliffe Camera. In the image graph (c), the node for the above image (*red*) connects clusters of All Souls images (*cyan*) and Radcliffe images (*magenta*).

$$m_{ij} = n_{ij} + \sum_{k,\{(j,k),(k,j)\}\in E} n_{ik} \qquad (3)$$

where m_{ij} is the augmented number of occurrences of word i in image j. The value m_{ij} replaces n_{ij} in Equation (2).

In the case where one image descriptor is used to validate adjacency with multiple images, image augmentation will count an extra occurrence of that descriptor for each match present. This is equivalent to the introduction of importance scaling to the *tf* which weights descriptors by the number of images matched.

While image augmentation is similar in spirit to query expansion [7], it has the advantage of allowing known image relationships to be used in the initial *tf-idf* score. Query expansion can only benefit a query when a correct match has already obtained a high ranking.

6. Performance Evaluation

The dataset used for testing was the Oxford Buildings dataset [2] consisting of 5062 images taken around Oxford. Images containing 11 different buildings have been manually ground truthed as *Good*, *OK* or *Junk*.

- *Good* images: building is fully visible.
- *OK* images: at least 25% of the building is visible.
- *Junk* images: building is present, but less than 25% is visible.

In addition, a set of 100,000 background images taken from Flickr was used.

Image features were generated using the Hessian-Affine interest point detector [12] along with the SIFT descriptor [9]. For the visual vocabulary, we used the INRIA Flickr60K vocabulary [8], generated from a separate set of images. Use of a separate set of images to generate the vocabulary better mimics large database BOW image matching applications where vocabularies cannot be trained to recognize a specific subset of images containing the object of interest.

The feature detector, visual vocabulary set and 100K background images were obtained from [3].

Useful feature generation

For all useful feature databases we constructed, the number of ranked images to geometrically check was set to 30. Images were represented with a maximum of 300 descriptors.

In useful feature images with more than 300 features labeled as useful, the features geometrically validated by the most images were used. Two methods for handling singleton images were tested. In one case, all features from singleton images were discarded. In the other case, the 300 largest-scale descriptors were used.

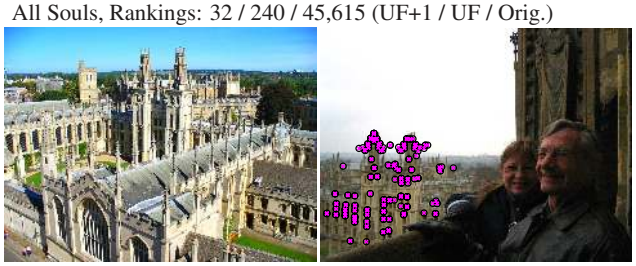
A visual vocabulary of 200,000 words was used. In tests it was shown that results using a smaller 50,000 word vocabulary resulted in similar trends but with a reduction in recognition performance.

Recognition evaluation

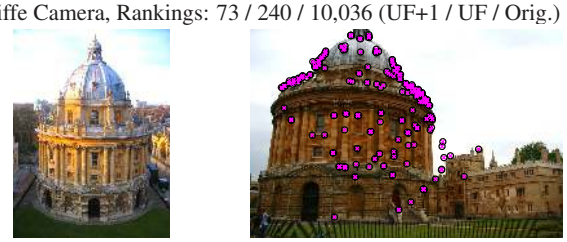
As useful feature detection can be considered a training step, separation of the dataset into testing and training sets was necessary. Failure to do so would result in query images from the test set being used to validate features in the training set. To prevent this, image recognition performance was evaluated using K-fold cross validation, with number of folds set to 5.

All *Good* images were used as query images for a given building. *Good* and *OK* images were considered positive matches, while *Junk* images were ignored. Only *Good* and

All Souls, Rankings: 32 / 240 / 45,615 (UF+1 / UF / Orig.)



Radcliffe Camera, Rankings: 73 / 240 / 10,036 (UF+1 / UF / Orig.)



Ashmolean, Rankings: 39 / 392 / 767 (UF+1 / UF / Orig.)

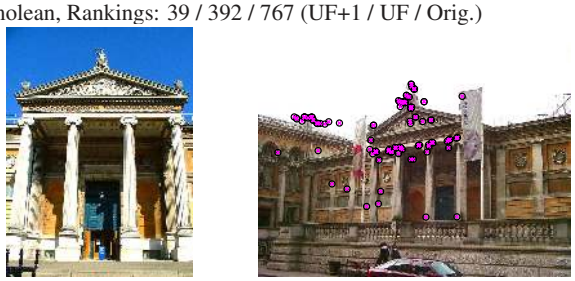


Figure 4. Examples of image matching cases where using only useful features outperformed using all features. These are typically images with significant viewpoint or lighting changes. Query images used (left) and their corresponding matches (right) are displayed.

OK images were split using K-fold cross validation, unlabeled images and *Junk* images were always included with the background set. Resulting image databases contained an average of 104,950 images.

Following the approaches of [7, 8, 16], recognition performance was evaluated using the average precision (AP) score. Average precision is the mean of image precision across all recall rates, providing a single numerical metric for the overall recognition performance of an algorithm. AP can be visually interpreted as the area under a precision-recall curve. AP values range from 0 to 1, with the latter only being achieved when 100% precision is obtained at full recall.

Though our results are not directly comparable to those in previous works due to the need to use cross-validation, it should be noted that the BOW framework used as our base case closely follows that of [15]. Furthermore, as our testing framework makes use of whole images as queries, we can expect lower recognition results than those reported in similar works which make use of only descriptors from the object of interest to query a database.

	Images	Original Descriptors	Useful Descriptors
Singleton images: discarded			
Total	104,950	222.27 M	1.92 M (**0.87%**)
Singleton	88,917	173.13 M	0 (**0 %**)
"Useful"	16,033	49.14 M	1.92 M (**3.92%**)
Singleton images: 300 largest			
Total	104,950	222.27 M	27.06 M (**12.17%**)
Singleton	88,917	173.13 M	25.13 M (**14.52%**)
"Useful"	16,099	49.14 M	1.92 M (**3.92%**)

Table 1. Image database summary for the Oxford (5K) + Flickr (100K) datasets. Descriptor and file counts reflect the mean across all cross validation folds. As expected, the large Flickr background set contains many singleton images.

Database types

All results are generated from performing BOW querying (section 3.1) on image databases. Database types tested are listed below.

- **Original:** Images represented using all features.
- **UF:** Images represented using only useful features.
- **UF+1:** Images represented using useful features, and those of adjacent images
- **UF+2:** Images represented using useful features, and those of 2-adjacent images

7. Results

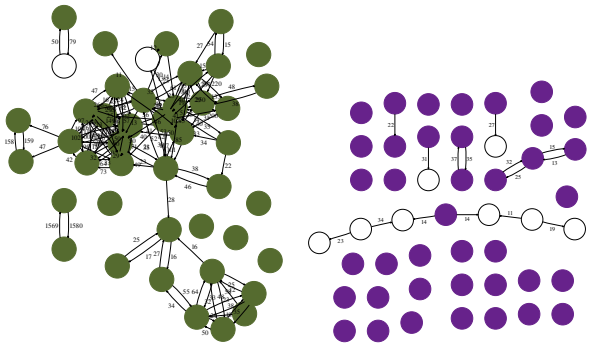
7.1. Descriptor reduction

Generation of the useful feature databases resulted in a large reduction in the number of descriptors present. Table 1 shows the number in image features stored in the original database as well as the useful feature database equivalent. The number of image features from the "useful" images was reduced by approximately 96%, with useful features making up less than 1% of all database features. Rather than being represented by thousands of features, images are now represented by a maximum of 300.

A significant reduction in features is obtained by discarding features from singleton images. In some cases, labeled building images were deemed to be singleton images, resulting in lowered recognition results.

7.2. Recognition performance

In order to fairly compare the performance of BOW ranking using only useful features (*UF*) to BOW ranking using all features (*Orig.*), AP results are summarized by building (Table 2). It is possible to see that using only useful features yields recognition rates comparable to those using all features, with a slight improvement in some cases.



(a) Hertford (b) Magdalen

Figure 5. Building image graphs for (a) *Hertford* (highly connected) and (b) *Magdalen* (poorly connected). Coloured images represent *Good* or *OK* examples of building images. Edge labels denote the number of geometrically verified useful features between images.

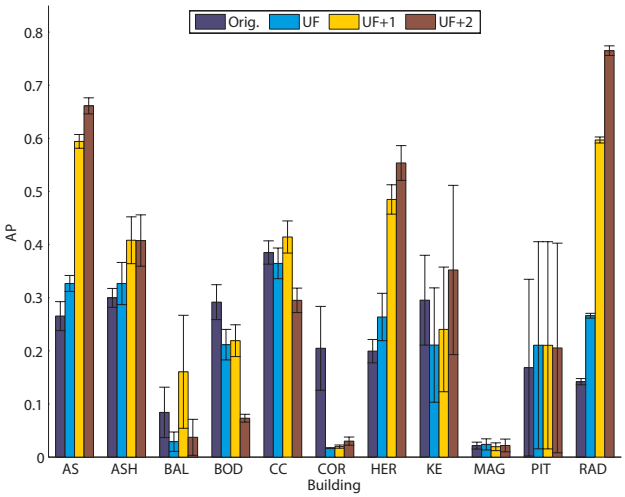


Figure 6. Mean Average precision (AP) performance comparison between database types when using only useful features. Error bars represent standard error across cross validation folds.

The addition of image augmentation (*UF+1*,*UF+2*) improved results significantly, with some buildings having recognition improve dramatically (e.g., *Radcliffe Camera* images improving from **0.142** using all features to **0.765** using the *UF+2* database type). Some example queries with improved *tf-idf* rankings are displayed in Figure 4. Note that these images are all matched despite the change in lighting, viewpoint and the presence of objects in the foreground.

In order to determine the effectiveness of image augmentation when using lower quality queries, *OK* images were used. As expected, overall recognition performance is lower than that of the *Good* images, however the benefit of image augmentation is still clear.

It is clear that useful feature detection requires a minimum level of performance on the initial BOW querying and verification during the image graph construction phase in order to properly identify useful features. In the case of *Magdalen* images (Figure 5(b)), poor initial *tf-idf* ranking yielded very few valid geometrically verified matches resulting in many singleton images. An examination of *Hertford* images (Figure 5(a)) shows that an adequate initial ranking resulted in a highly connected set of images with many useful features being detected.

8. Conclusions

We have presented a method for identification of useful features in a set of images as well as a method for image feature augmentation in a bag-of-words framework. Our results show that pre-processing images to extract useful features can improve recognition performance while reducing memory requirements for image features by 96%. It may seem surprising that recognition is improved by discarding features that fail to match other training images, as at least a few potentially useful features will inevitably be discarded. However, our results show that the enhanced quality of the selected feature set can more than compensate for its reduced size, while at the same time providing large reductions in memory requirements.

Our method for including features from adjacent images while matching gives a substantial improvement in query performance without the need to explicitly define or reconstruct distinct objects. Instead, it efficiently combines features at runtime from related viewpoints based on their matching relationships within the image graph.

The treatment of singleton (unmatched) images should depend on the requirements of the application. For many real-world applications, such as recognition of landmarks from public image collections, it will be appropriate to discard singleton images, as they are likely to contain only transient objects and clutter. However, in cases where it is important to use singleton images, we have demonstrated that one solution is to select a restricted number of large-scale features from the singleton images. In future work, we hope to explore other methods to improve the selection of features from singleton images.

The image graph has been shown to be a useful data structure for improving recognition as well as understanding image data. Our graph visualization has allowed for the identification of new unlabeled landmarks. This suggests that the image graph can be used in other ways to further improve recognition, such as attaching missing labels or correcting mislabelings.

Building	Images (Queries)	Singleton: discarded				Singleton: large			
		Orig.	UF	UF+1	UF+2	Orig.	UF	UF+1	UF+2
All Souls	77 (24)	0.265	0.327	0.594	**0.661**	0.265	0.300	0.588	**0.648**
Ashmolean	25 (12)	0.300	0.327	**0.408**	0.408	0.300	0.321	0.404	**0.405**
Balliol	9 (5)	0.084	0.029	**0.160**	0.037	**0.084**	0.014	0.039	0.069
Bodleian	24 (13)	**0.292**	0.212	0.219	0.073	**0.292**	0.208	0.212	0.075
Christ Church	78 (51)	0.385	0.365	**0.414**	0.295	0.385	0.352	**0.405**	0.287
Cornmarket	12 (5)	**0.205**	0.017	0.020	0.030	**0.205**	0.016	0.019	0.030
Hertford	54 (35)	0.199	0.264	0.485	**0.554**	0.199	0.229	0.466	**0.559**
Keble	7 (6)	0.295	0.211	0.240	**0.352**	**0.295**	0.153	0.268	0.260
Magdalen	54 (13)	0.022	**0.024**	0.020	0.022	0.022	0.022	**0.023**	0.021
Pitt Rivers	6 (3)	0.168	**0.210**	0.210	0.205	0.168	0.204	**0.205**	0.201
Radcliffe	221 (105)	0.142	0.266	0.597	**0.765**	0.142	0.183	0.544	**0.749**
Good Queries (All Buildings)		0.214	0.205	0.306	**0.309**	0.214	0.182	0.289	**0.300**
Good Queries (All Queries)		0.218	0.267	0.464	**0.514**	0.218	0.224	0.436	**0.504**
OK Queries (All Buildings)		0.123	0.125	0.193	**0.223**	0.123	0.109	0.194	**0.219**
OK Queries (All Queries)		0.117	0.161	0.336	**0.418**	0.117	0.123	0.312	**0.407**

Table 2. Query performance by building when singleton images are discarded or represented using large scale features. Building values reflect mean AP scores taken on *Good* queries for a given building. Bolded results corresponding to the method with the best performance. In order to demonstrate that useful features are not sensitive to poor query images, results for *OK* images being used as queries are also shown. *Orig.* - original database, *UF* - useful feature database, *UF+1,UF+2* - useful features with image augmentation

References

[1] http://www.graphviz.org/.

[2] http://www.robots.ox.ac.uk/∼vgg/data/oxbuildings/.

[3] http://lear.inrialpes.fr/ jegou/data.php.

[4] R. A. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[5] O. Chum and J. Matas. Web scale image clustering: Large scale discovery of spatially related images. In *Technical Report CTU-CMP-2008-15, Czech Technical University in Prague*, 2008.

[6] O. Chum, J. Matas, and J. Kittler. Locally optimized ransac. In *DAGM-Symposium*, pages 236–243, 2003.

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

[8] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *European Conference on Computer Vision*, volume I of *LNCS*, pages 304–317, oct 2008.

[9] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[10] D. G. Lowe. Local feature view clustering for 3d object recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 682–688, 2001.

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

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

[13] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Applications*, 2009.

[14] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[15] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[16] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[17] J. Philbin and A. Zisserman. Object mining using a matching graph on very large image collections. In *Computer Vision, Graphics and Image Processing, Indian Conference on*, 2008.

[18] D. Pritchard and W. Heidrich. Cloth motion capture. In *Eurographics*, 2003.

[19] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[20] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *International Conference on Computer Vision*, volume 2, pages 1470–1477, Oct. 2003.