Robust Feature Selection for Large Scale Image Retrieval

by

Panu James Turcot

BASc. Honours System Design Engineering, University of Waterloo, 2007

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

THE FACULTY OF GRADUATE STUDIES
(Computer Science)

The University Of British Columbia
(Vancouver)

September 2010

Abstract

This paper addresses the problem of recognizing specific objects in very large datasets. A common approach has been based on the bag-of-words (BOW) method, in which local image features are clustered into visual words, providing memory savings through feature quantization. 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 approach, which we name Robust Feature Selection (RFS), is based on the observation that many local features are unreliable or represent irrelevant clutter. We are able to select "maximally robust" features 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 the Oxford Buildings dataset. 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 by merging them with maximally robust features from neighbouring images. We demonstrate adjacent and 2-adjacent augmentation, both of which give a substantial boost in recognition performance.

Preface

A version of the robust feature selection method outlined in Chapters 4 and 6 has been published. Turcot, P. and Lowe, D. (2009), “Better matching with fewer features: The selection of useful features in large database recognition problems.” 2009 IEEE International Conference on Computer Vision Workshops. All implementation of methods presented and testing was conducted by Panu Turcot. The manuscript was written by Panu Turcot with contributions by David Lowe.

Table of Contents

List of Tables

List of Figures

Glossary

BOW	bag-of-words
DOF	degree of freedom
idf	inverse document frequency
K-NN	K nearest neighbour
NN	nearest neighbour
RFS	Robust Feature Selection
SIFT	Scale-Invariant Feature Transform , a method for generating image descriptors developed by David G. Lowe
tf-idf	term frequency inverse document frequency
tf	term frequency
WGC	weak geometric constraints

Acknowledgments

I would like to thank...

David G. Lowe for his guidance, patience and support during the course of my graduate studies.

My labmates who assisted through countless questions on a neverending variety of topics and provided fun discussions to keep me sane on the longer days.

My girlfriend Jennifer whose work ethic had me in the lab first things in the morning and whose company kept my happy all the while.

The Natural Sciences and Engineering Research Council of Canada (NSERC) and the Canadian Institute for Advanced Research (CIFAR) for their financial support.

Chapter 1

Introduction

1.1 Motivation

Making use of computer vision algorithms to perform image search would allow for content-based retrieval, in which images themselves could be used to find additional images and any associated information. Image-based retrieval would allow a user with a mobile phone camera to easily obtain information of a location or object simply by taking a photo and using the photo to search the internet. A smart shopper in a store might search the internet for similar products, price compare with online retailers and find reviews on an item they find in the store. A curious tourist might want more information on a local attraction, or help identifying something they encounter in their travels.

Internet scale image search is challenging due to the amount of data available. Resources such as online collections contain hundreds of millions of images and are open to the public. Efforts are underway to create sets of millions of hand-labelled images of thousands of objects [7]. When we consider the amount of video available on the internet, the amount of data is further increased. Algorithms able to work on this scale require new techniques and new frameworks.

Current image search methods used by popular internet search engines are effectively text-based retrieval methods making use of web-page content, and image file names to determine image search results. In order for users to search for an image, the subject of the image must be first translated to text in order for an effective query to be formulated, e.g., the name of a person or location, a description of the image contents. While this allows users to find example images on a known topic, identification of unknown image content becomes difficult. In order for content-based retrieval of images on the internet to be possible new image search methods need to be developed.

In this work, we address the task of image retrieval on large collections of images which we refer to as large database image recognition. Similar to text document retrieval, image recognition refers to the task of correctly identifying matches from a large database given a query. The query consists of an image which contains an object of interest and correct matches from the database should contain the same object of interest. We show that feature selection is an effective technique for improving recognition performance while reducing memory requirements. This is achieved by leveraging the large amount of visual data to identify and augment robust features in object images.

1.2 Background

One method which has proven to be successful at object recognition was presented by Lowe [15]. Lowe's Scale-Invariant Feature Transform (SIFT) makes use of local scale and orientation invariant image descriptors to perform matching, making it robust to viewpoint changes, as well as occlusions. A final verification step consisting of a geometric check of tentative correspondences formed during the descriptor matching phase results in a highly reliable set of matched features.

As the number of images increases, the method becomes less suitable as memory requirements increase. Lowe uses a k-d tree and a best-bin first search strategy, resulting in sub-linear increase in search times, however all descriptors must reside in main memory in order for fast searching to be possible. When the number of database descriptors reaches the billions, storage of all the descriptors becomes difficult.

A partial solution to this problem was proposed by Sivic and Zisserman [27] in which nearest neighbour (NN) searching through descriptor space is replaced with a quantization of the descriptor space. This method is now commonly referred to as bag-of-words (BOW) matching. In the BOW matching framework presented by Sivic and Zisserman, descriptors are clustered using k-means and cluster centers are stored. These cluster centers, called visual words, form what is referred to as a visual vocabulary, with each word defining a Voronoi region in descriptor space. The visual vocabulary effectively now becomes the basis for matching between database and query images. Descriptors mapped to the same visual word approximate a K nearest neighbour (K-NN) voting scheme.

Following image descriptor assignment to visual words, individual descriptors are discarded. As many methods still employ the SIFT descriptor, this memory savings corresponds to the compaction of a 128-dimensional vector (128 bytes) into a single integer of a few bytes. Geometric information associated with each descriptor must still be stored.

For matching, Sivic and Zisserman employ a common text-retrieval method known as term frequency inverse document frequency (*tf-idf*) ranking [2] and demonstrate that it is suitable for image recognition. While the exact formulation of term frequency (*tf*) and inverse document frequency (*idf*) terms in the ranking can vary, they represent word frequencies in documents and a weighting to reduce the impact of common visual words respectively. Like many feature based methods, final recognition is improved using a geometric check of the inital matches provided by the *tf-idf* ranking.

The work from Sivic and Zisserman [27] set the basic framework for BOW-based large database image recognition. Many advancements have been proposed to improve recognition performance, speed and memory use. However, these methods consist of the same key components:

- Visual vocabulary generation
- Visual word assignment
- *tf-idf* matching
- Geometric re-ranking

1.3 Contributions

In this thesis we present Robust Feature Selection (RFS), a method to further reduce the amount information stored from each image, while still maintaining strong recognition performance for objects of interest, through the preservation of only a minimal set of image features, which we refer to as *maximally robust* features.

We define a maximally robust 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. Sample images showing maximally robust features following RFS are shown in Figure 1.1.

Our method builds on that of Philbin et al. [23], employing a BOW framework and *tf-idf* ranking. A BOW image database is built using the full feature set. Following construction, database images are used as queries in order to discover additional viewpoints of the objects contained in these images. For each database image, the best *tf-idf* matches are geometrically validated through estimation of epipolar or affine geometry combined with additional image feature constraints. In cases where a match between images is found, geometrically consistent descriptors are labelled and retained while all other descriptors are discarded. In cases where there is limited memory, the number of features used to represent an image can be effectively throttled to a chosen memory size using RFS.

In addition to feature selection, this pre-processing step provides an effective way to discover relationships between database images containing the same object. Rather than discard this information, we store discovered pairwise matches between images in a graph structure which we later use to improve image recognition performance.

In our experiments, testing is conducted on the Pasadena Buildings [1] and Oxford Buildings [23] datasets using a cross validation procedure. Our results show that on the Oxford Buildings dataset, RFS eliminates 97% of image descriptors while maintaining recognition performance. Using the image matching graph, we achieve significantly improved recognition performance without necessitating the storage of any additional features. Results are also presented for the University of Kentucky [20] object dataset.

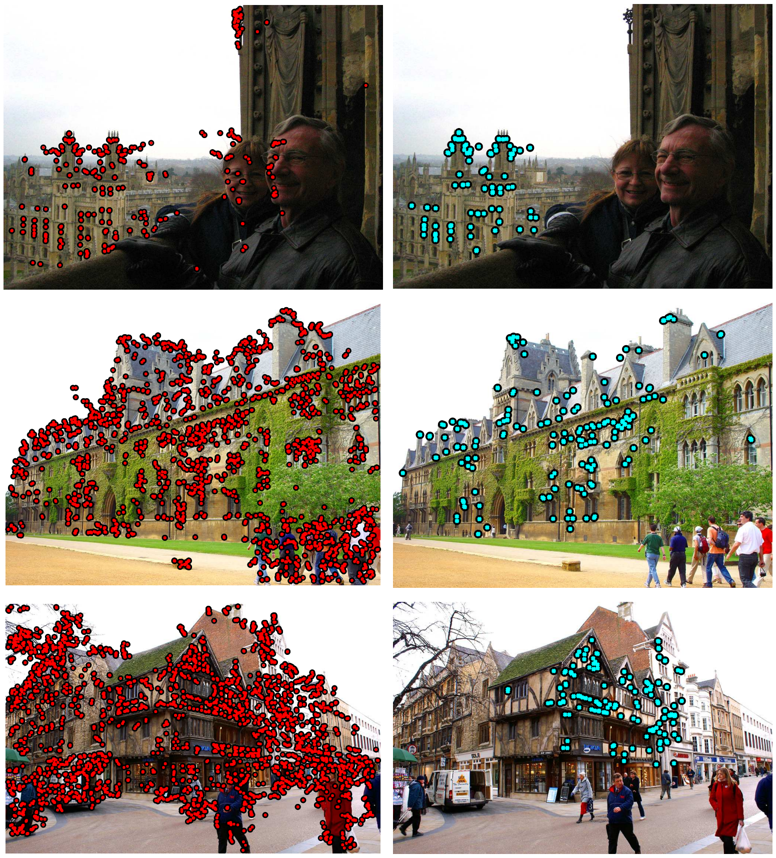

Figure 1.1: Original image features (*left*) and those preserved by robust feature selection (*right*). Transient objects in the foreground and non-distinctive areas of the scenes do not contain robust features.

1.4 Organization

This thesis presents our method for extracting and using maximally robust features as follows. Chapter 2 outlines previous work done in the field of large database image recognition. Section 2.1 presents the BOW framework used in our method used to generate our initial matches. Robust Feature Selection and the geometric validation are discussed in Chapter 3 and Chapter 4. Chapter 5 discusses the image matching graph as well as introduces a new method making use of matching relationships in a BOW framework. The evaluation procedure is provided in Section

Chapter 6 and results are presented in Section Chapter 7.

Chapter 2

Previous work

2.1 Bag-of-words matching

The BOW framework developed by Sivic and Zisserman [27] is built from a group of cluster centers in descriptor space, referred to as visual words $W = \{w_1, w_2, ..., w_i\}$. 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:

$$d \rightarrow w^* = \arg\min_{w} dist(d, w) \tag{2.1}$$

As the number of visual words increases, visual word assignment can become a computational bottleneck. In such cases, approximate nearest word search can be used which provides significant speedup over linear search while maintaining high accuracy. Our implementation of the BOW framework employs the FLANN approximate NN library developed by Muja and Lowe [19].

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 a BOW image database.

While the visual words used in this and other work consist of K-Means clustering of descriptor space using Euclidean distance, it should be noted that this choice of vocabulary is not limited to this formulation. Word assignment in the BOW framework can be conducted using any method that converts high dimensional descriptors into a single index, such as locality sensitive hashing [12].

2.1.1 Querying BOW databases

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}:

$$x_{ij} = \underbrace{\frac{n_{ij}}{\sum_i n_{ij}}}_{\text{tf}_{ij}} \underbrace{\log \frac{N}{\sum_j |n_{ij} > 0|}}_{\text{idf}_i} \tag{2.2}$$

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 $I_j = [x_{1j}, x_{2j}, x_{3j}, ..., x_{ij}]$ which is then normalized to unit length.

$$\overline{I_j} = \frac{I_j}{\|I_j\|_2} = [\overline{x}_{1j}, \overline{x}_{2j}, \overline{x}_{3j}, ..., \overline{x}_{ij}]$$

Distance between images is calculated using the *L2* distance metric or cosine similarity, which are equivalent for length-normalized vectors. Two images that contain no shared visual words are orthogonal and by default have a distance of 2.

$$\begin{aligned} \text{dist}(\overline{I_A}, \overline{I_B}) &= \|(\overline{I_A} - \overline{I_B})\|_2^2 \qquad (2.3) \\ &= \overline{I_A}^T \overline{I_A} - 2\overline{I_A}^T \overline{I_B} + \overline{I_B}^T \overline{I_B} \\ &= 2 - 2\overline{I_A}^T \overline{I_B} \\ &= 2 - 2\sum_i \overline{x_{iA}} \overline{x_{iB}} \qquad (2.4) \end{aligned}$$

While (2.4) is in fact a form of weighted histogram intersection, it is important to note that the number of visual words greatly affects the behaviour of the matching method. *Tf-idf* weight vectors used to represent images have a length equal to the number of visual words, $|W|$. As each image generates roughly 3000 descriptors and it is common for $|W|$ to approach one million in the field of large database image recognition, the *tf-idf* weight image representation is a sparse vector.

As a result, it is common for a given query image to not intersect a large portion of the image database, a property which allows for querying speed as only intersections need be computed by (2.4). In effect BOW matching with a very large visual vocabulary should be interpreted as a method for performing approximate nearest-neighbour search of individual descriptors.

To speed up querying, the *idf* term idf_i and the normalizing factor for *tf-idf* weight vectors I_j of database images are precomputed and stored.

As images contain often repeated and therefore uninformative descriptors, a stop-list of the most common words is generated and those words suppressed, a technique shown by Sivic and Zisserman to be effective at improving recognition performance.

Our tests show that the effect of the stop-list on *tf-idf* performance (Figure 2.1) varies depending on the dataset and vocabulary size. For simplicity, a stop-list size of 1% of the visual vocabulary was used in all reported results as this appeared to perform well in most cases.

2.2 Scaling bag-of-words

Nister and Stewenius [20] demonstrated that a *tf-idf* based approach can scale up to a large number of images with the use of a larger visual vocabulary. This was achieved using recursive k-means clustering to form a vocabulary tree containing up to 16 million leaf nodes. The increased quantization resulted in more accurate matches but came at the cost of a loss of generality; small changes to the image descriptor could cause it to be quantized into a different visual word resulting in a missed match. To counteract this, higher level nodes in the tree were given weights as well, allowing a coarser quantization to influence the *tf-idf* score. Despite this weighting of higher level nodes, performance decreased as the number of visual words was increased above 1 million suggesting 1 million visual words as being a good compromise between over and under quantization.

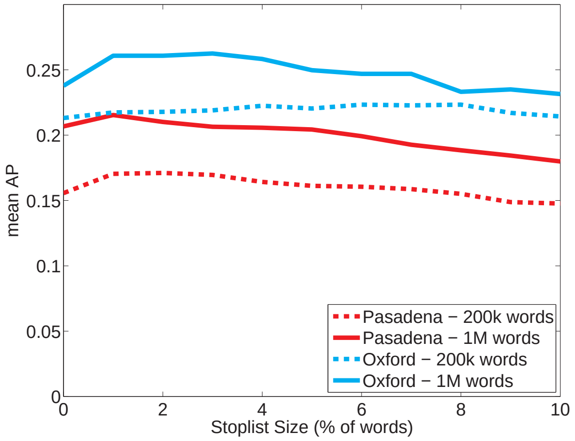

Figure 2.1: Performance of *tf-idf* ranking based on the Oxford and Pasadena Buildings datasets when using 200,000 and 1,000,000 visual words. Recognition varies noticeably with the size of the stop-list used.

Nister and Stewenius also improved search times in BOW matching through the use of an inverted index data structure to store word occurences. With an inverted index, a list of images is indexed based on visual words and only database images containing the same visual words as the query need be accessed. Exploiting the sparsity of a larger number of visual words, scores could be computed much more efficiently and fast matching from a set of thousands of images was possible.

Also using vocabulary trees, Schindler et al. [26] showed that the selection of features which maximized information gain in the tree allowed for the construction of more discriminative visual vocabularies. To do this, a tree was constructed to evaluate how informative features were at discriminating between images of similar locations. A new tree was then trained using only these informative features, improving overall recognition performance. The feature selection method of Schindler et al. relies on labelled images in order to perform informative feature selection and is focused on improving the performance of the visual vocabulary. All features extracted from images are still used.

Philbin et al. [23] compared a vocabulary tree to a flat vocabulary of 1 million words using an approximate NN algorithm to allow for efficient clustering. Their results showed that a flat clustering of the descriptors allowed for a more uniform partioning of the descriptor space and outperformed a vocabulary tree. Again, results suggested that a vocabulary of 1 million visual words yielded the best results, with vocabulary sizes up to 1.25 million being tested. The *tf-idf* framework employed by Philbin et al. was the same as Sivic and Zisserman.

Evaluation of their research was conducted on the original University of Kentucky dataset but also they introduced a new dataset which has proven to be the benchmark on which much future work has been evaluated: the Oxford buildings dataset. Furthermore, to demonstrate the scalability of this method, a background dataset of 1 million images was used while still maintaining fast search times. Our work builds directly upon the method presented by Philbin et al.

2.3 Multiple word assignment

As the number of visual words increases, the accuracy of visual vocabularies as an approximation to the K-NN improves (Figure 2.2). However, the recall of the NN search decreases as there is a loss in generality: small perturbations in descriptors can cause them to be mapped to neighbouring visual words due to very small Voronoi regions. One solution proposed was to allow for multiple word assignment: rather than link a descriptor to a single visual word, store entries in the index for multiple words (Figure 2.2). Nister and Stewenius's use of higher level nodes in the vocabulary tree was a form of multiple assignment, however as the tree structure was generated in a greedy fashion, these higher level nodes were effectively a small visual vocabulary and did not have the improved accuracy of multiple assignment on a flat clustering of the descriptor space. Philbin et al. [24] adapted their earlier work [23] to include multiple assignment. Their approach was to perform a weighted assignment of descriptors to the nearest 3 visual words resulting in improvements to recall without a loss in precision.

Jégou et al. [10] introduced a concept calling Hamming embedding (HE) which allowed for smaller vocabularies (e.g., 200,000 visual words) to perform as well as of 1 million word vocabularies. This was done through further encoding of a descriptor's position within a visual word using a binary code. Query descriptors were then matched to database descriptors by selecting all descriptors in the same visual word within a fixed hamming distance of the query binary code. The bi-

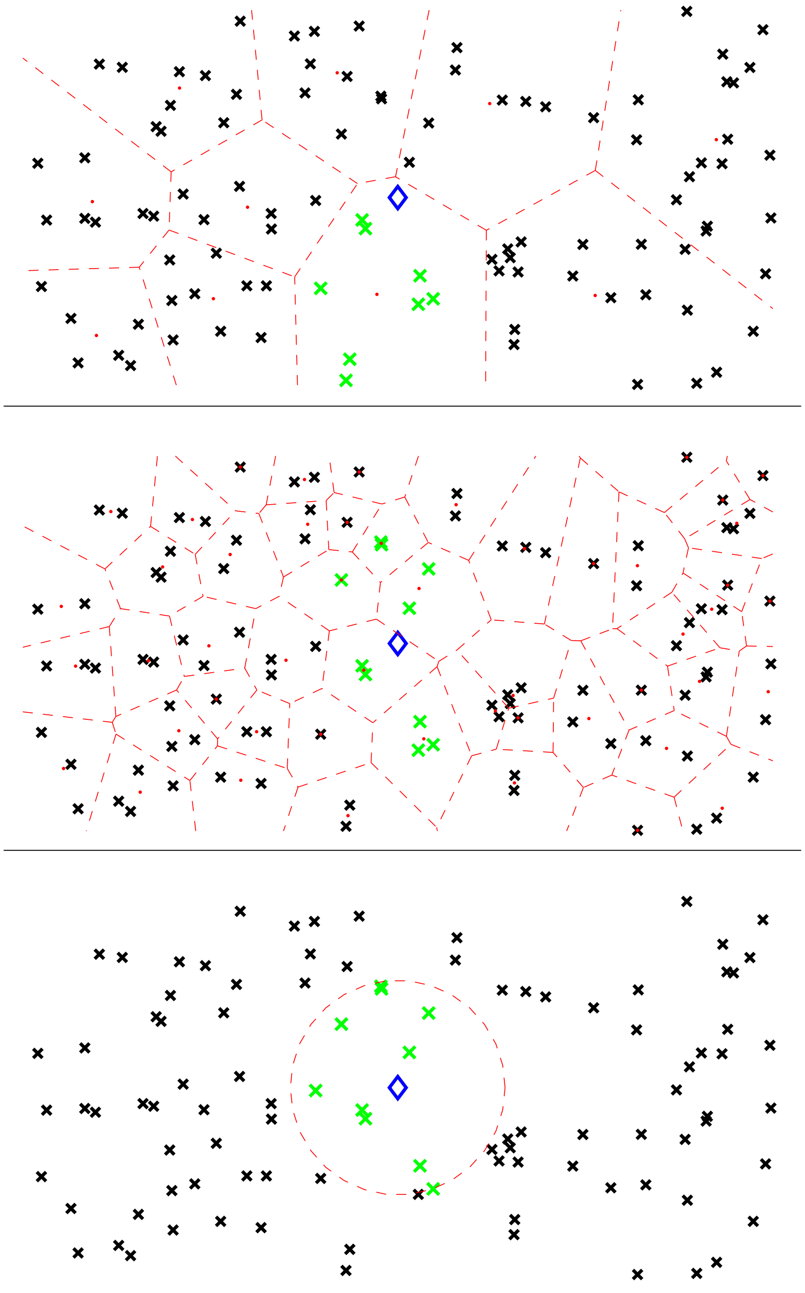

Figure 2.2: A comparison of BOW to K-NN. A query descriptor (*blue*) is matched to database descriptors (*x*) based on Voronoi regions representing visual words. *Top:* Visual vocabulary with small vocabulary; *Middle:* Visual vocabulary with large vocabulary and multiple assignment *Bottom:* K nearest neighbours of the query descriptor.

nary code effectively further quantized each visual word and the hamming distance threshold acted as a criterion for multiple assignment. Later, Jégou et al. [11] incorporated a multiple assignment, making use of a variable number of words for assignment. When combined with Hamminng embedding, it was shown that multiple assignment resulted in improved recognition over Hamming embedding alone. Despite a smaller vocabulary, the variable word multiple assignment employed by Jégou et al. outperformed the multiple assignment scheme used by Philbin et al. when vocabularies were trained on an unrelated set of images. We would like to note that direct comparison of results between methods is difficult due to the differing vocabulary sizes and training images used.

Multiple assignment, while proven to be effective at improving performance [24] [21] [10] is also a trade-off. When conducted on database descriptors, the size of the inverted index is increased as well as the time needed to compute *tf-idf* distances. Due to memory restrictions, a popular approach has been the use of multiple assignment only on the query image descriptors, which increases recognition accuracy at the cost of some extra computation, but without increasing the index size [21] [11].

2.4 Vocabulary generation

Wherever possible, the visual vocabulary should be generated from a sample of images of the objects attempting to be matched [23] [21]. Philbin et al. [23] showed that for the Oxford buildings dataset, recognition of buildings decreased by over 25% when the visual vocabulary was trained on a separate set of images. In this case, the training images were also of buildings of similar architectural style found in Paris.

While many methods are evaluated using vocabularies generated from the set of images being tested [27] [20] [6] [24], we adopt the approach of Jégou et al. [11] and use a vocabulary from an independent set of images taken randomly from Flickr. Using an independent set of images is motivated by the fact that it is not always possible to know the test images at training time, such as when images are added incrementally to an existing database.

Jégou et al. [11] employ exact k-means for visual vocabulary generation, and showed that the use of exact k-means provides a more uniform quantization of descriptors than hierarchical methods, which improves average query time. As vocabulary generation consists of clustering millions of descriptors into hundreds of thousands of visual words, the use of exact NN is very computationally expensive and no clear benefit over using approximate NN has been demonstrated.

As approximate NN libraries such as FLANN[1] offer fast parameter tuning, high NN accuracy and significant speedups over linear search, the use of approximate methods is highly favorable for vocabulary generation. It should be mentioned that visual word assignment is also conducted using approximate NN [10][11] [23] [6] [24] in order maintain fast query times. While Jégou et al. are able to learn a graph structure which outperforms FLANN in fast visual word assignment, their method requires manual parameter tuning and longer training times.

2.5 Geometry

The intial spatial re-ranking conducted by Sivic and Zisserman [27] consisted of a consistency check of the neighbouring regions around each correspondence. Local neighbourhoods surrounding matched features in the query and database images were searched and consistent neighbors from both images voted towards that image as a valid match. While simpler than computing an affine mapping between images, it was found by Sivic and Zisserman that a local neighbourhood check was sufficient.

Philbin et al. [23] introduced fast spatial verification based on a single correspondence between two elliptical regions generated by an affine invariant Hessian regions [16]. The overall transform between two images was limited to vertical shear, scaling in x and y, as well as translation and was computed by transforming both elliptical regions to a unit circle while preserving the vertical orientation. Using a single correspondence allowed for explict checking of every possible correspondence in a matched pair of images, making geometry checking deterministic. The limitations of this single correspondence geometry checking is that it relied on the use of an affine invariant interest point detector and was limited to a 5 degree of freedom (DOF) model.

[1] http://www.cs.ubc.ca/~mariusm/index.php/FLANN/

Jégou et al. [10] successfully integrated feature scale and orientation into the *tf-idf* ranking process using a method they called weak geometric constraints (WGC). When ranking images using WGC, several *tf-idf* scores are computed for each image based on estimated scale and orientation changes. In this way, features with consistent scale and orientation shifts provide evidence for a given image and inconsistent features do not affect the *tf-idf* score. To our knowledge, this is the only work which has attempted to integrate geometric information directly into the *tf-idf* ranking stage of BOW matching. All other work has focused on using geometric information only at the re-ranking stage. Suprisingly, naïve use of WGC results in decreased *tf-idf* performance, and only when priors on scale and orientation changes are introduced does the *tf-idf* ranking benefit from WGC. While our matching method does not incorporate WGC into the *tf-idf* distance computation, we do make use of a similar filtration method based on scale and orientation differences to speed up geometric verification.

Perdoch et al. [21] approached geometric information with the mindset of memory reduction. The same way clustering was used to quantize image descriptors, a method for quantizing geometric data associated with image features was employed. Quantization was achieved through the formation of a geometric vocabulary which was used to represent affine invariant regions surrounding each feature. In doing so, Perdoch et al. was able to quantize geometric information to as few as 8 bits with a negligible reduction in performance, resulting in significant memory savings for fast searching of a database of 5 million images.

One property used by various methods was that objects in many images found in online collections are either upright or rotated by multiples of $\pi/2$. Jégou et al. as well as Perdoch et al. leveraged this fact to improve results. Perdoch et al. showed that if a gravity vector assumption is made, recognition performance improved suggesting many photos found in online collections are already correctly oriented.

2.6 Query expansion

Chum et al. [6] successfully improved recall through the successful application of query expansion, a document retrieval technique, to image retrieval. Their method extended the work of Philbin et al. [23] by making use of the top ranked initial *tf-idf* image to formulate new queries which in turn generate new matches in the database. Following geometric verification of the multiple queries, a final reranking was conducted aggregating the multiple sets of results.

In order for query expansion to succeed, a valid match to the query image must be established through strict geometric verification; naïve use of the *tf-idf* matches to generate a synthetic query results in lowered recognition performance. While more advanced synthetic query generation schemes were proposed, the simplest method proposed was called average query expansion. In average query expansion, all geometrically validated images are averaged together to generate a single additional query. In all cases, the success of query expansion at improving recall hinges on a strong initial *tf-idf* ranking to provide positive matches for synthetic query generation and requires multiple iterations of *tf-idf* searching and geometric re-ranking at query time.

We propose a similar strategy to improve BOW matching called image augmentation. Unlike query expansion, image augmentation improves initial *tf-idf* results and offers recognition performance improvements without requiring additional queries.

2.7 Object discovery

Philbin and Zisserman [22] employed their earlier work [23] in order to perform unlabelled object discovery from a set of images. This was done by discovering matching pairs of images within the image collection and storing the results in a graph structure called a matching graph which we also use in our approach. Each image was used to query the image collection and a geometric consitency check used to validate matches. Following construction, clustering was performed on the graph to isolate interconnected groups of images into distinct objects or scenes. The result was that groups of related images, usually depicting a common object or scene in the image collection, were identified.

Chum and Matas [3] presented a similar concept for object discovery however focused instead on fast computation of the graph. Rather than an exhaustive search, initial image pairs were determined through the use of a fast hashing-based matching. In this way, it is possible to seed the matching graph with candidate images.

While it is possible that some object clusters will be missed, they showed that large clusters will be found with high confidence. Furthermore, no post-processing was conducted on the graph, resulting in very large interconnected groups of images.

In both cases, the purpose of the matching graph was object discovery in large unlabelled image collections. While our matching graph construction is very similar to that of Philbin and Zisserman, we show that the relationships discovered during graph construction can be used to improve recognition in a dataset through our image augmentation method.

The approach taken by Gammeter et al. [8] is again similar to that of Philbin and Zisserman [22], however rather than generating a large image collection they used image meta-data to generate many small collections before attempting to identify objects. This was done by generating a kd-tree using GPS coordinates and subdividing images based on geographic location, simplifying the task of extracting objects by removing many irrelevant images. As part of the object discovery method, a bounding box for objects is estimated in images and they show that discarding descriptors outside the bounding box results in a 33% reduction in the image index size without a significant drop in performance. While this work performs one type of feature reduction, we show that by combining the matching graph with feature selection, we can achieve improved recognition results while reducing the image index size by over 95%.

2.8 Relevance

A summary of relevant work is chronologically arranged in Table 2.1. While a wide array of methods improve and refine image recognition using the BOW framework, few approaches focus on reducing the memory requirements which ultimately determine the ability of this framework to scale to larger datasets. As image descriptors and geometric information associated with individual image features have already been compressed effectively [21], the solution to further memory reduction in a feature based approach is to reduce the number of features stored in the index.

In this work, we apply feature selection to the BOW framework. In doing so we reduce the size of the index stored through the elimination of features which do not aid in the task of object recognition. While some approaches have begun to experiment with the removal of non-informative features using object bounding boxes, effectively identifying important sub-regions of images, we focus instead on feature level selection and attempt to preserve a minimal set of features needed to properly represent an object.

Such an approach has successfully been used by Li and Kosecka [13] in location recognition, allowing for accurate recognition with as few as 10% of the original descriptors. However, Li and Kosecka make use of the original SIFT descriptors and therefore the question as to whether aggressive feature selection can be successfully applied to the BOW matching framework remains open. Furthermore, like Schindler et al., the work of Li and Kosecka requires the location associated with every image to be known in order to perform feature selection.

One important focus of this work is that we make no assumptions that image collections will be annotated with class information and do not rely on class labels to identify features, but instead rely on our unsupervised matching technique. Furthermore, while it has been shown that assumptions on image scale and orientation can be used to improve performance, we attempt to make no such assumptions about the data.

Table 2.1: Summary of related researching employing the BOW framework

Year	Author	Vocabulary size	Vocabulary type	Multiple assignment	Scoring method	Geometry
2003	Sivic and Zisserman [27]	60K	KM -	-	*tf-idf*	local neighbourhood check
2006	Nister and Stewenius [20]	10K - 16M	HKM	weighted high level nodes	Hierarchical *tf-idf*	-
2007	Schindler et al. [26]	1M	Discrim. HKM	-	weighted votes	-
2007	Philbin et al. [23]	50K - 1.25M	AKM & HKM	-	*tf-idf*	3-5 degree of freedom (DOF) from 1 corresp.
2007	Chum et al. [6]	1M	AKM	-	*tf-idf*	3 DOF from 1 corresp.
2008	Philbin et al. [24]	10K - 1M	AKM	nearest 5	*tf-idf*	5 DOF from 1 corresp.
2008	Jégou et al. [10]	20K - 200K	AKM	Hamming embedding	*tf-idf* + WGC	affine
2009	Gammeter et al. [8]	500K	AKM	-	*idf*	homography
2009	Perdoch et al. [21]	1M	AKM	nearest 5 (query only)	*tf-idf*	geometry quantization + gravity vector
2010	Jégou et al. [11]	20K - 200K	KM (AKM query)	Hamming embedding	*tf-idf* + WGC	affine

KM - (Exact) K-means

AKM - Approximate K-means. K-means generated using an approximate NN algorithm.

HKM - Hierarchical K-means. Vocabulary trees using recursive K-means clustering.

Chapter 3

Feature selection

In this work we examine the effectiveness of feature selection in the context of BOW image matching. We define feature selection to be the task of selecting a subset of image features I_{sub} for an image which provide an accurate and efficient representation of the image to allow for later retrieval through *tf-idf* matching.

$$I_{sub} \subset I : \{f_1, f_2, f_3 ...\}$$

The use of a stop-list is a form of feature selection in which only features from uncommon visual words are preserved. In the case of a stop-list, feature selection is conducted based on visual words. The number of features preserved for each image depends entirely upon the distribution of visual words in the database and the individual image.

In order to conduct feature selection on an image level, we must have a method by which to differentiate individual image features according to some metric of robustness. An ideal method would produce a ranked list of the most robust features in an image and would allow the user to throttle the number of features to be preserved from each image. In cases where more memory is available, a stringent number of original image features can be preserved whereas in situations with large memory restrictions, the number of features to be preserved from each image can be small.

3.1 Maximally robust features

When generating features from an image, many image features which are not robust are extracted. When using interest point detectors, interest points that are not present in multiple viewpoints, which we call unstable interest points, are a source of non-robust image features. Feature descriptors that are uninformative and occur frequently in many images are also considered non-robust. Non-robust features could also include those on 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 robust 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 robust feature selection, for large datasets this condition will normally be met. We discuss the special case of images with single views (singleton images) in Section 3.3.

In large database image matching applications, it is assumed that images may not be labelled. Therefore, RFS is fully unsupervised.

3.2 Implementation

In order to perform robust feature selection, a BOW image database containing the full feature set is constructed. Each image in the database, I_j, is used as a query to perform standard *tf-idf* search, ranking the remaining database images. The highest M ranked images are verified for geometrically consistent features, validating possible matches generated by the initial *tf-idf* ranking. If a match is deemed present, only those features which pass the geometric consistency check are flagged as robust. The overall robustness score of a given image feature f_i in query image I_j is simply the number of images in which geometric consistency was established using that feature:

$$R(f_i) = \sum_{\alpha=1}^{M} \begin{array}{ll} 1 & f_i \text{ matched } I_{rank-\alpha} \\ 0 & \text{otherwise} \end{array} \tag{3.1}$$

where $R(f_i)$ defines a robustness function for a given image feature f_i and $I_{rank-1}, I_{rank-2}, I_{rank-3}$ represents the first, second and third best *tf-idf* matches to image I_j respectively.

For our tests, the number of ranked images to check geometrically was set to 30 ($M = 30$). The value of the M parameter affects RFS by influencing search for possible matches. Higher values will lead to longer training times due to increased geometric verification, however they will permit more images to influence feature selection. Lower values of M can result in missed matches, possibly resulting in images failing to find a suitable match.

During RFS, stop-listed visual words are excluded for geometric verification and subsequently never get labelled as robust. This choice is motivated the fact that stop-lists are an effective method for improving *tf-idf* performance. We observed that inclusion of the stop-listed visual words in RFS results in a decrease in final recognition performance.

3.3 Singleton images

Images without any geometrically valid matches are considered singleton images. In the context of object discovery in large image collections, singleton images represent images containing no commonly occuring objects or scenes of interest which we call non-object images. As such descriptors from these images can be 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 selecting features which are robust to affine or other distortions of the image [25].

3.4 Labelled and annotated data

In cases where image labels, or some other form of image annotation is present, image metadata can be used to speed up or improve RFS. In this case BOW databases would be constructed using a subset of images with similar image labels or image properties. In the work of [8], GPS coordinates are used to subdivide the full set of images into geographic regions. Similarly, BOW databases for individual classes can be generated and used for RFS.

Alternatively, class labels can be used to re-rank initial *tf-idf* matches, allowing for images from the same class to be preferentially used for geometric validation. In the case where there are many labelled images from the same class, those with the best *tf-idf* score can be used for validation. If few labelled images are available, robust feature selection will still allow for object discovery from the unlabelled images by validating features against the top *tf-idf* matches. This method is effectively identical to unlabelled robust feature selection, with an improved initial *tf-idf* ranking.

Chapter 4

Geometric verification

The use of visual words to represent image descriptors between images introduces unique challenges to geometric verification of image features. Our BOW framework makes use of the SIFT descriptor [15] and the visual vocabulary is a quantization of SIFT descriptor space. In traditional descriptor-based matching using SIFT, point correspondences between query and database images are determined using nearest neighbor search, often combined with a distance ratio check [15]. This typically results in a rich set of putative correspondences, which we will refer to as SIFT correspondences. SIFT correspondences typically contain few false correspondences resulting in a high inlier ratio, simplifying the task of geometric verification.

In a BOW framework, only visual words are available to differentiate features. Correspondences are established between all features that share the same visual word with no further information by which to filter these initial correspondences. As a result, each feature in the query image can match multiple features in the database image resulting in multiple pairwise correspondences, only one of which is correct. This problem is compounded when multiple features in the query image share the same visual word. The resulting many-to-many point correspondences greatly increase the number of false correspondences in the image posing difficulties in the geometric verification.

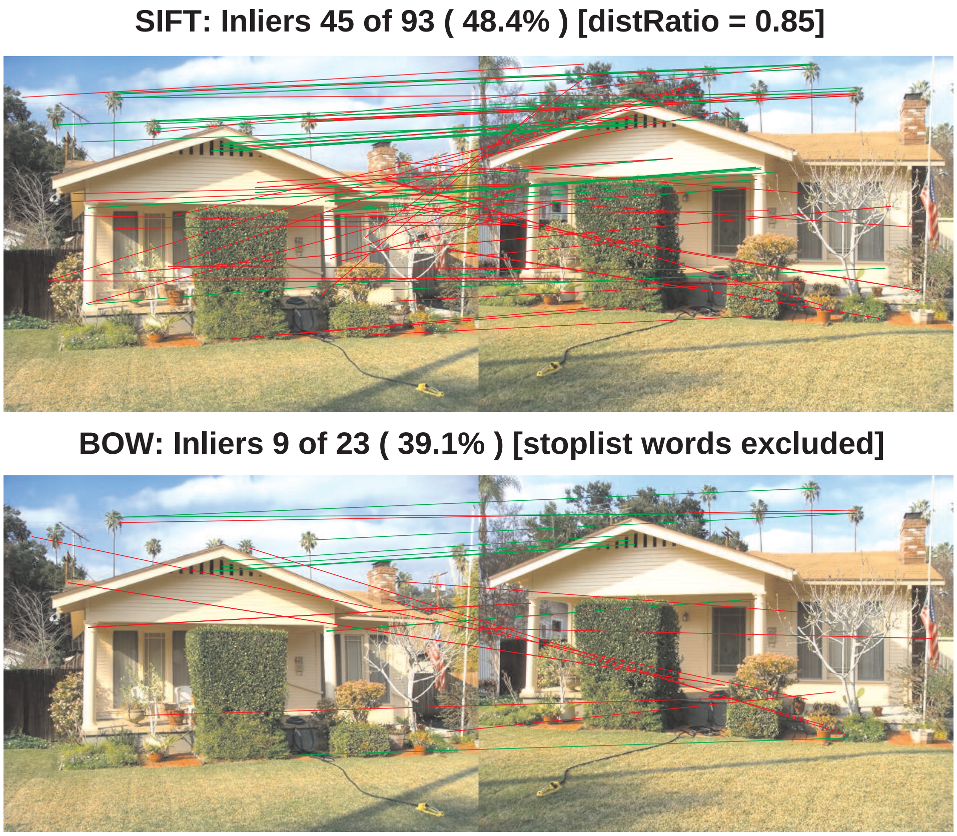
SIFT: Inliers 45 of 93 (48.4%) [distRatio = 0.85]
BOW: Inliers 9 of 23 (39.1%) [stoplist words excluded]

Figure 4.1: Results of geometric verification of traditional descriptor-based feature matching (*top*) where query image features are matched to at most a single database feature (one-to-one). BOW-based feature matching (*bottom*) allows multiple query image features to match multiple database features (many-to-many).

4.1 Correspondence ranking

Geometric verification is performed using a modified version of the LO-RANSAC algorithm [5] to determine either the epipolar or affine geometry between images. In order to improve our chances of finding a correct model using LO-RANSAC, a subset of correspondences with the highest probability of containing an inlier are used for sampling and verification.

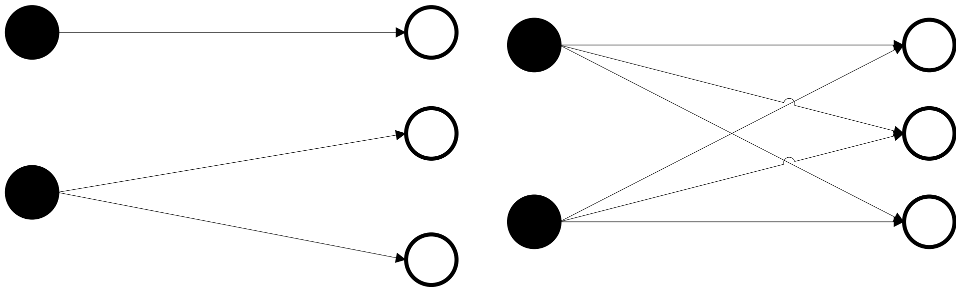

Figure 4.2: Examples of possible bipartite graphs found in bag of words correspondences. Nodes represent features from the query image (*blue*) and database image (*yellow*) while edges represent a shared visual word between query and database feature.

A feature in the query image should only match a single feature in any given database image. When a query image feature shares a visual word with two database image features, two possible correspondences are generated, each with at best a 50% probability of being correct.

In the case of a matching between α query and β database features, we approximate:

$$\text{P(inlier)} \approx \frac{1}{\alpha\beta} \tag{4.1}$$

While this represents only an upper bound on the true probability of finding an inlier in a group of correspondences, it gives a useful ranking of the probability for any pair being a correct correspondence. In our experiments, models were generated using the 200 highest ranked correspondences. This choice was made primarily to improve processing times through the preservation of a relevant subset of correspondences. It was observed that in near duplicate images as well as images with highly repeated structure, the number of putative correspondences could be in the thousands, significantly decreasing run time.

Figure 4.3 shows an image pair with a large number of BOW correspondences. Increasing the number of correspondences allowed for LO-RANSAC to reliably find the correct solution in fewer iterations. The downside however is the computation time needed to validate candidate models, which increases linearly with the

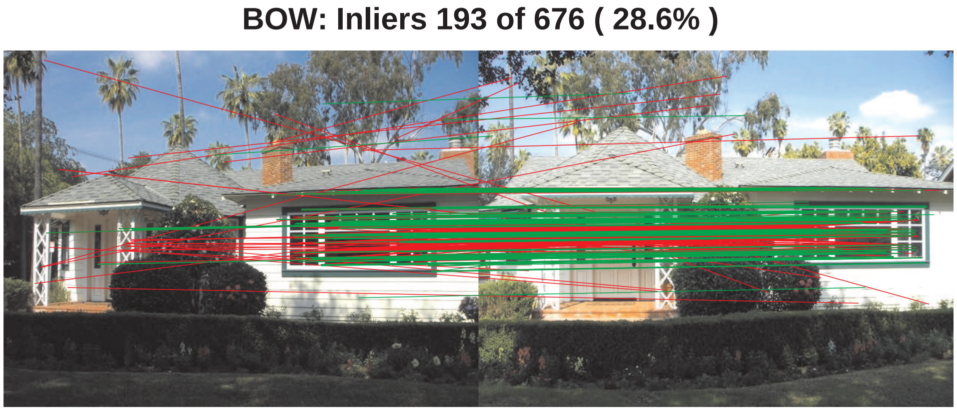

Figure 4.3: Example image pair from the Pasadena Buildings dataset containing many correspondences.

number of correspondences (Figure 4.4). The increased variability in the number of inliers found in the 600 inlier case is due to LO-RANSAC occasionally failing to converge onto a valid solution. Examination of the correspondences confirms that smaller subsets of ranked correspondences do contain a higher proportion of inliers (Figure 4.5).

4.2 Model fitting

As previously mentioned, we consider both epipolar and affine geometry between images during the geometric verification step using the LO-RANSAC algorithm.

The normalized 8-point algorithm [9] is used for epipolar geometry estimation from the initial sampling, and a least-squares approximation is used for the local optimization parameter estimation. Following LO-RANSAC, a least squares solution is estimated from all inliers and a final iteration of local optimization is applied to determine the final solution. In our experiments, the local optimization step in LO-RANSAC resulted a more consistent solution and increased the inlier count when compared to the standard RANSAC algorithm.

An affine fit is determined using a 3-point sample of tentative correspondences, and follows the same steps as estimation of the epipolar geometry. Previous work employed a simplified geometric model along with local affine parameters of the

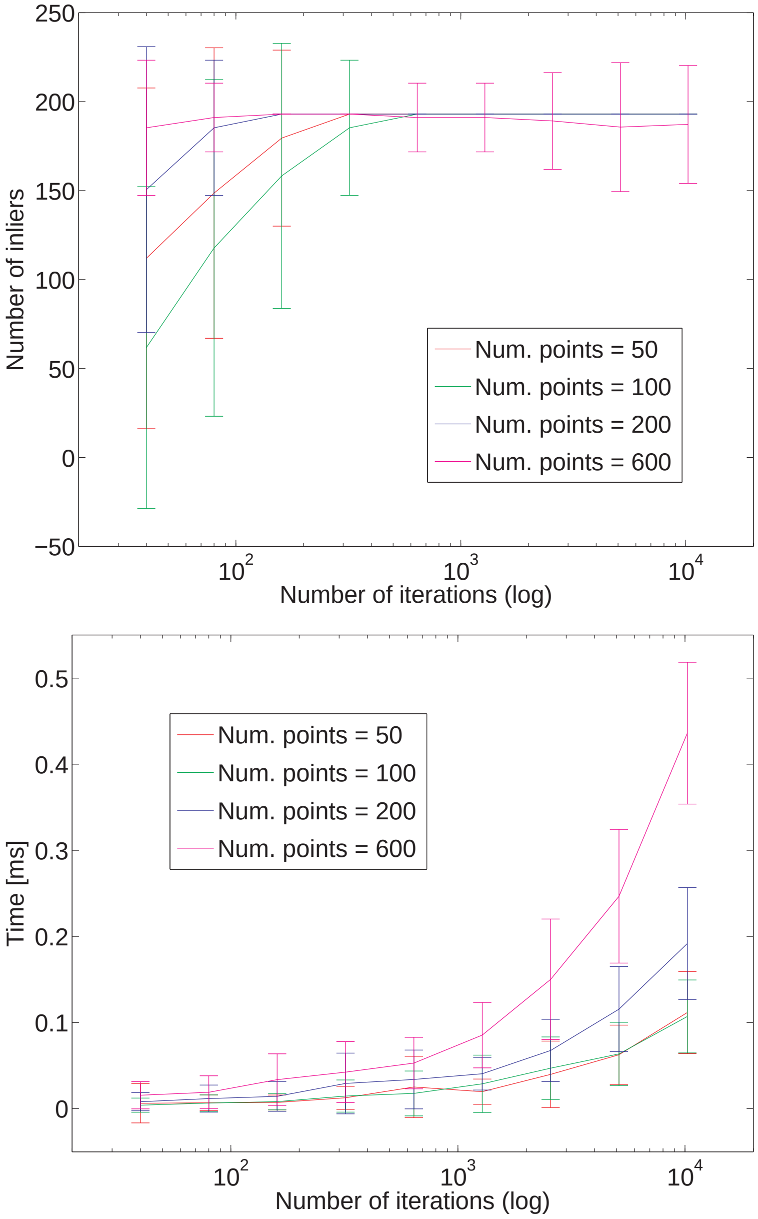

Figure 4.4: The effect of subsampling correspondences on the number in inliers (*top*) and time taken (*bottom*).

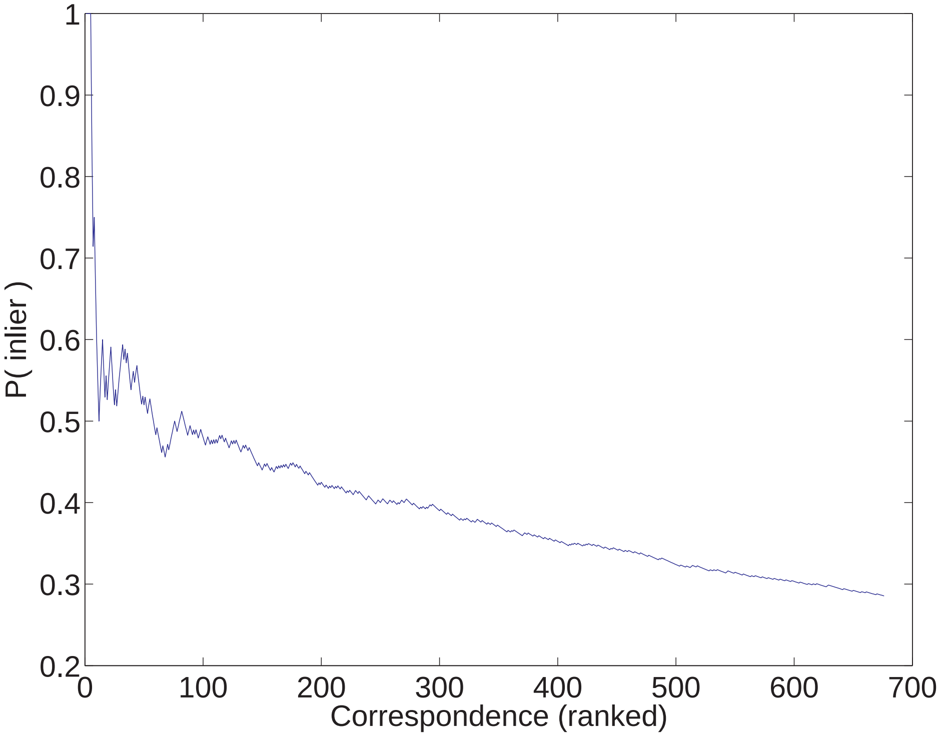

Figure 4.5: Probability of a correspondences being an inlier (inlier ratio) for a given subset size. Note: 200 on the x-axis represents sub-sampling the top 200 of 676 correspondences.

feature to solve the many-to-many correspondence problem [23], however it has been suggested that the storage of local affine invariant patches might not be necessary and that only point correspondences need be preserved [21].

Our sampling and verification also make use of feature geometry to discard matches and speed up model verification. All valid inliers for a given sample must share a scale change that agrees within a factor of 2, as well as an orientation difference within a range of $\frac{\pi}{6}$. Using these constraints allows for fast rejection of outliers, speeding up geometric validation considerably (Table 4.1).

In the presence of many-to-many correspondences, care must be taken so that features are only used to validate one correspondence. This amounts to finding a matching set of a bi-partite graph where nodes and edges represent features and

Table 4.1: Processing time for 100 trials of RANSAC under varying conditions. All methods resulted in correctly identifying 193 inliers from a set of 676 correspondences.

	Time (sec)	
	Top 200 correspondences	All 676 correspondences
Orientation & scale filtering	0.199	0.219
No filtering	1.82	1.86

correspondences which satisfy orientation, scale and model constraints. One valid solution would be to include the correspondences that make up the maximal matching set of the bipartite graph. An alternative would be to select inlier correspondences in a greedy fashion based on which correspondences more closely match the geometric model, which we call best-fit first selection of correspondences. Our tests show that while the maximal matching yields more inliers, RFS performance improves when using a best-fit first selection is used. One possible explanation is that the best-fit first selection choses correspondences that are more likely to be correct as they more closely match the geometric model between images.

Chapter 5

Image adjacency

In addition to filtering out uninformative descriptors, robust feature selection 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 robust feature selection 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 between images v_A and v_B.

A visualization of the image matching graph shows the relationships between database images (Figure 5.1). Even though the matching graph construction is unsupervised, images of the same building naturally group together and form interconnected clusters. New object detection can be conducted by analyzing the matching graph for these clusters of interconnected images, suggesting many photos of the same object. Figure 5.2 highlights the effectiveness of the matching graph at identifying objects from an unlabelled set of images.

5.1 Image augmentation

The construction of the matching 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

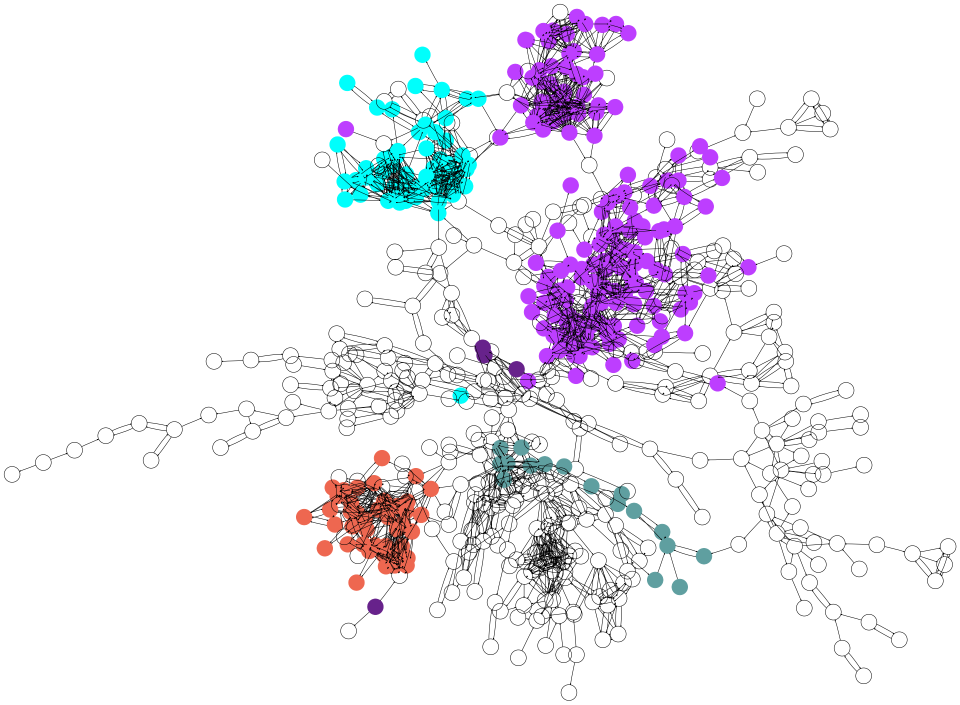

Figure 5.1: Matching graph showing connectivity between database images. This is one connected sub-graph from the full Oxford Buildings dataset. Ground truthed images are coloured by building labels. Images of the same location naturally form highly connected regions in the graph without any prior knowledge about image content. Graph generated using GraphViz[a]

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

nearby viewpoints of the same object. We present a method for integrating multiple viewpoints, also referred to as view clustering [14], which allows images with similar views to share features.

As previously discussed, *tf-idf* matching is effectively a weighted histogram intersection between two sparse vectors which approximates nearest-neighbour searching on individual image descriptors. Making use of the matching graph, we implement a scheme which allows for the transfer of individual feature matches between linked database images. This is conducted through a simple modification of the image representation.

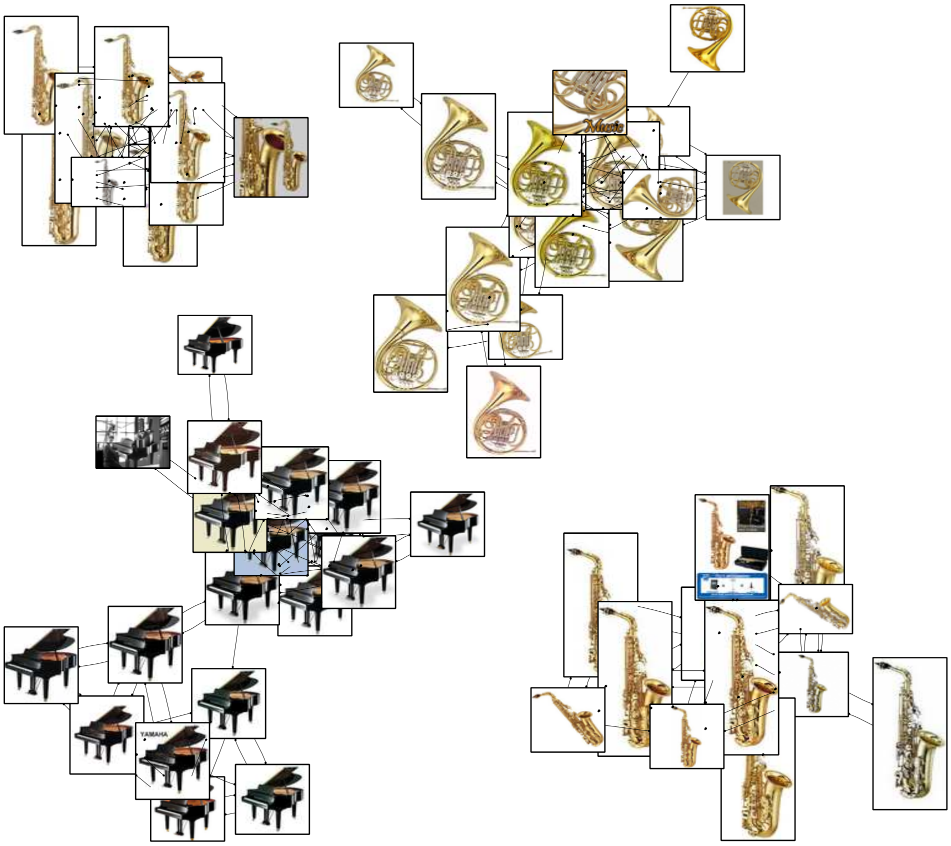

Figure 5.2: Objects discovered from robust feature selection performed on a set of 10,000 musical instrument images taken from Imagenet

For every image I_j in our image database, referred to as the base image, we represent the image not only with its own descriptors, but also include the descriptors of every image within a local neighborhood in the matching graph. We define the local neighborhood of a given base image as all images that can be reached with T-edge traversals on the matching graph (Figure 5.5). This pool of descriptors forms the new augmented representation for the base image I'_j.

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. We use the notation $RFS+1$ to denote the case $T = 1$, i.e., only images immediately adjacent to base image are used form I'_j. Formally, this can be computed

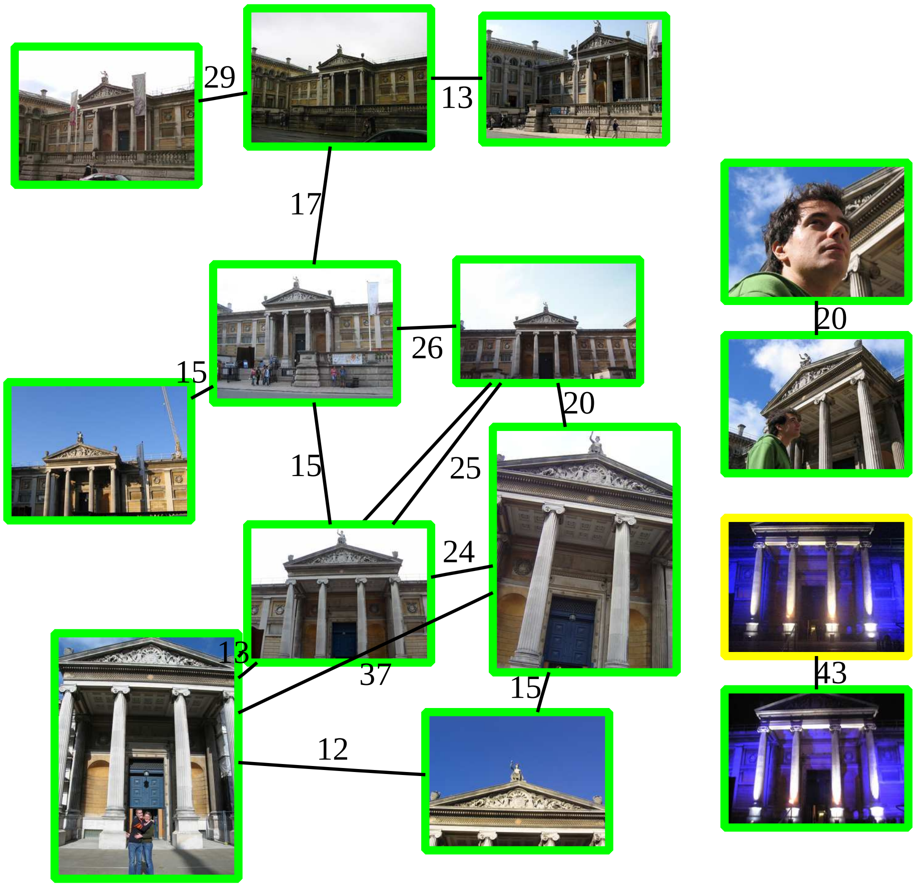

Figure 5.3: Example matching graph showing sub-graphs connecting to Ashmolean images from the Oxford Buildings dataset. Positive examples of Ashmolean images are highlighted green while partial-view images are highlighted yellow. Edge labels denote the number of validated features in the match.

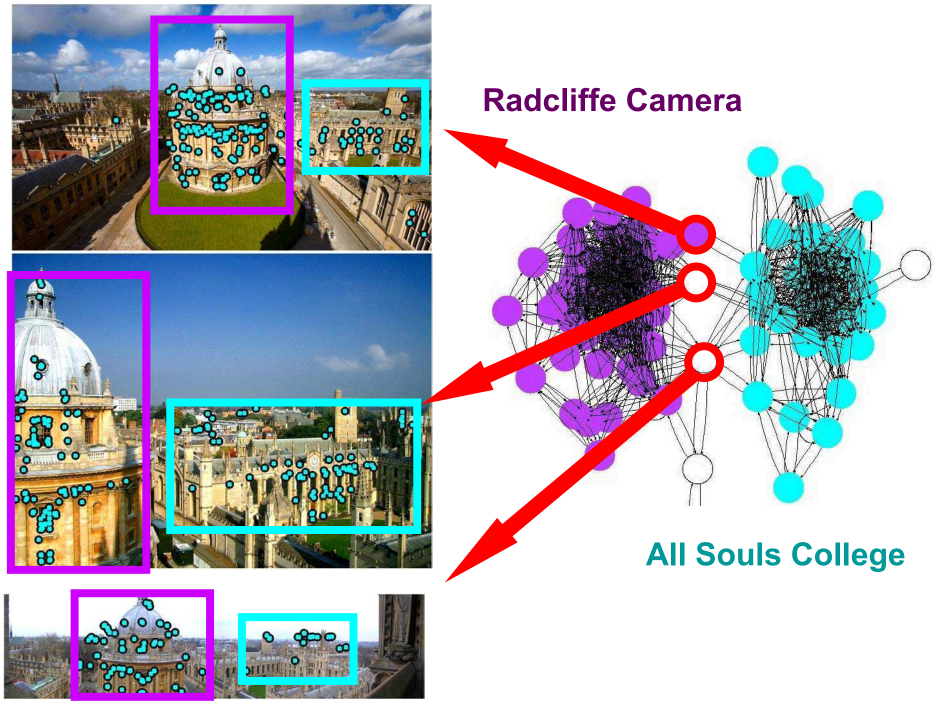

Figure 5.4: Maximally robust features for selected images from the Oxford buildings dataset. Note that these images retain features from both All Souls college and Radcliffe Camera. The corresponding graph shows that the above images connect clusters of All Souls images (*cyan*) and Radcliffe images (*magenta*).

as:

$$RFS+1: m_{ij} = n_{ij} + \sum_{k,\{(j,k),(k,j)\}\in E} n_{ik} \tag{5.1}$$

where m_{ij} is the augmented number of occurrences of word i in image j. The value m_{ij} replaces n_{ij} in (2.2).

While image augmentation is similar in spirit to query expansion [6], 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.

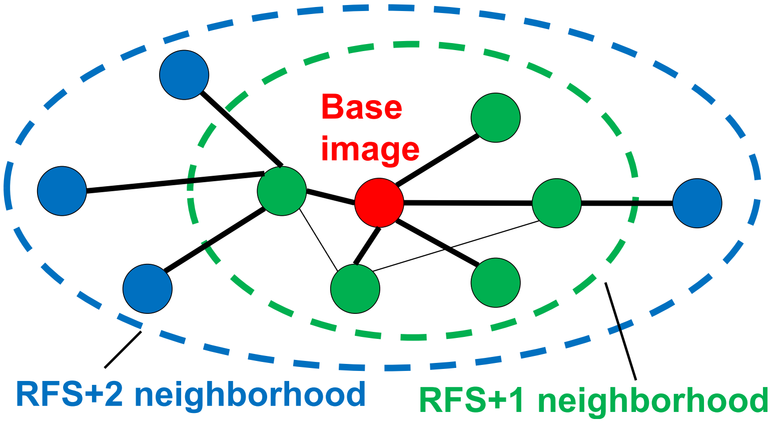

Figure 5.5: Sample graph showing two possible local neighborhoods for use in image augmentation. Following image augmentation, the base image (red) is represented with a *tf-idf* vector formed by pooling all word occurrences from the images in the local neighborhood (orange, yellow).

Augmented image representations can be computed at query time using the base image word occurences and the matching graph. In order to conducted fast querying with image augmentation, the augmented image size, $\sum_i m_{ij}$, for each image needs to be stored along with the inverse document frequency, idf_i, and *tf-idf* normalization factors for the augmented case. This permits fast normalization of term frequencies and calculation of *tf-idf* weights at query-time.

Chapter 6

Performance evaluation

Average Precision

Following the approaches of [6, 10, 24], 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.

In the case of the Oxford buildings, overall mean average precision (mAP) was computed by giving each building equal weight. In this way, overall scores are not biased towards cases with more building images.

Though our results are not directly comparable to those in previous work due to the need to use cross-validation, it should be noted that the BOW framework used as our base case closely follows the approach of Philbin *et al.* [23].

6.1 Robust feature selection

Image features were generated from dataset images using the Hessian-Affine interest point detector [17] along with the SIFT descriptor [15].

For the visual vocabulary, we used the INRIA Flickr60K vocabulary [10], 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.

Unless otherwise specified, a visual vocabulary of 1,000,000 words was used as it has been previously suggested that this yields the best recognition performance. Recognition results using a smaller 200,000 word vocabulary are also presented.

The feature detector, visual vocabulary set and 100K background images were obtained online[1], and are publicly available.

6.2 Dataset

In order to evaluate the effectiveness of RFS, several datasets were used as listed in Table 6.1.

Oxford Buildings

The Oxford Buildings dataset[2] consists of 5062 images taken around Oxford. Images containing 11 different buildings have been manually ground truthed as *Good*, *OK* or *Junk*. *Good* and *OK* images are treated as positive examples of a building, while *Junk* images are ignored when scoring recognition.

- *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.

Pasadena Buildings

The Pasadena buildings dataset [1] consist of 750 images taken of 125 buildings (6 images per building). Building images have varying viewpoint and lighting conditions.

University of Kentucky

The University of Kentucky dataset [20] consists of 10200 images taken of 2550 objects (4 images per object). Object images have widely varying viewpoint and lighting conditions, resulting in a challenging dataset.

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

[2] http://www.robots.ox.ac.uk/$\sim$vgg/data/oxbuildings/

Table 6.1: Summary of datasets used in testing.

Dataset	Images	Descriptors	Cross validation
Oxford	5,062	15.89 M	5-fold
Pasadena	750	1.79 M	6-fold
Kentucky	10,200	10.70 M	-
Flickr100K	100,000	206.75 M	-

Flickr100K

In addition to these object datasets, a background dataset of 100,000 images from Flickr was used. This served to increase the overall number of images to show that the method can scale to large numbers of images, as well as provide distractors for RFS and image recognition.

6.3 Recognition evaluation

Cross validation

In previous work, there has been no explicit separation of the data into training (database) and test (query) images. As RFS 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.

In the case of the Oxford Buildings dataset, only *Good* and *OK* images (566 of 5062) were split using 5-fold cross validation. Unlabelled images and *Junk* images were always included with the background set. Resulting image databases contained an average of 104,950 images.

In the case of the Pasadena Buildings and Kentucky datasets, folds were split so that one image of each object was excluded at a time in the training phase.

Querying

Rather than use features from only object as queries, our work makes use of whole images as queries. In the case of the Pasadena and Kentucky datasets, this is

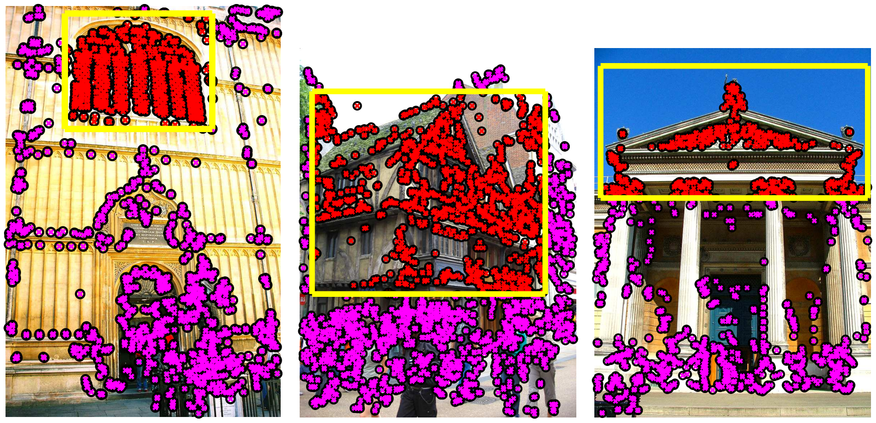

Figure 6.1: Example query images. Unlike previous work, all image features (*red* & *magenta*) are used to formulate queries. Past work employs only object features (*red*) to generate a query.

a natural choice as no object bounding boxes are provided. For the Oxford Buildings dataset, *Good* images excluded from the training set were used as queries. Our view is that the use of whole images are queries more closely resembles a real world query provided by an end user. Figure 6.1 shows example images from the Oxford Buildings dataset used for querying along with previously used query bounding boxes.

Database types

All results are generated from performing BOW querying (section 2.1.1) on image databases. No secondary geometric re-ranking is conducted and AP results reported reflect results from only the *tf-idf* ranking. Database types tested are listed below.

- **Original:** Images represented using all features.
- **RFS:** Images represented using only RFS features.
- **RFS+1:** Images represented using RFS features, and those of adjacent images ($T = 1$)
- **RFS+2:** Images represented using RFS features, and those of 2-adjacent images ($T = 2$)

Unless otherwise specified, all results are conducted with an unlabelled dataset.

Chapter 7

Results

7.1 Feature reduction

We evaluated the effectiveness of both affine and epipolar geometry for RFS using the Oxford and Pasadena Buildings datasets. Both affine and epipolar models are successful in identifying a large number of object images from the ground-truthed datasets, while filtering our the majority of the background images.

Table 7.1 gives a summary of the RFS features found in the two buildings datasets. In both cases, only a small subset of the original images were found to contain RFS features. Of these matched images, only 3.5-4% of the features were preserved by RFS. In order to demonstrate the effectiveness of using only RFS features for recognition, we discard features from singleton images and instead rely on RFS to preserve the important information. As a result, the index for the RFS image database is less than 1% the size of the original image database index. All reported recognition results using RFS features are achieved with a substantially reduced index.

7.2 Recognition performance

Table 7.2 outlines the overall recognition performance of the buildings datasets following RFS. In order to evalute the benefit of labelled data, results are also reported for the case where building image labels are known for training images.

Table 7.1: Image database summary for the Oxford and Pasadena Buildings datasets. Descriptor and file counts reflect the mean across all cross validation folds.

	Images	Original Features	RFS Features
Oxford + Flickr100K			
Total	104,950	222.27 M	0.57 M (**0.26%**)
Singleton	98,401	201.60 M	0 (**0 %**)
Matched	6,549	20.67 M	0.57 M (**2.76%**)
Pasadena + Flickr100K			
Total	100,625	208.24 M	1.01 M (**0.33%**)
Singleton	93,676	186.58 M	0.00 M (**0 %**)
Matched	6,949	21.67 M	0.70 M (**3.21%**)

These labels are used to re-order initial *tf-idf* matches as outlined in Section 3.4 and represent RFS with the ideal *tf-idf* ranking.

It is interesting to note that in both buildings datasets, the affine geometry outperforms the epipolar geometry. We suspect this is due to the nature of the buildings datasets, in which objects consist predominantly planar front-face of a building. The stricter affine solution is sufficient to model the building facade. This suggests that the use of lower DOF may be sufficient for tasks such as landmark recognition.

7.2.1 Oxford Buildings

In the case of the Oxford Buildings test, most building images were properly identified and the recognition score improves from 0.261 to as much as 0.39 despite the drastic reduction in features. Figure 7.1 shows the average precision recall curve which reveals that gains are achieved through improved precision at higher levels of recall.

A detailed view of the building by building performance can be found in Table 7.3 Figure 7.2) shows that buildings such as Radcliffe Camera (RA) and All Souls Col-

Table 7.2: Overall recognition performance on the Pasadena and Oxford Buildings datasets. Results presented for both affine and epipolar geometry. The labelled case indicates where class labels were used to re-order initial *tf-idf* results and represent RFS performance un-affected by *tf-idf* performance

Method (1,000,000)	**Recognition (mAP)**		
	RFS	**RFS+1**	**RFS+2**
Oxford + Flickr100K			
Original (All feats)	0.261		
RFS Affine	0.254	0.359	0.358
RFS Epipolar	0.257	0.347	0.343
RFS Affine + LABELS	0.271	0.365	**0.390**
RFS Epipolar + LABELS	0.254	0.354	0.362
Pasadena + Flickr100K			
Original (All feats)	0.215		
RFS - Affine	0.166	0.205	0.228
RFS - Epipolar	0.179	0.206	0.200
RFS - Affine + LABELS	0.167	0.212	**0.239**
RFS - Epipolar + LABELS	0.180	0.214	0.211

lege (AS) benefit significantly from RFS with scores improving from 0.194 (*Orig.*) to 0.765 (*RFS+2*) in the case of Radcliffe Camera. Images with few positive examples on which to train (PI, KE, BA, CO) underperformed.

One interesting case to note is the Magdalen (MA) performance (Figure 7.3). Despite a large number of images present, poor initial recognition performance caused RFS to fail and no robust features to be detected. This was due to large viewpoint and lighting changes in building images resulting in too few correspondences between images to generate a valid match.

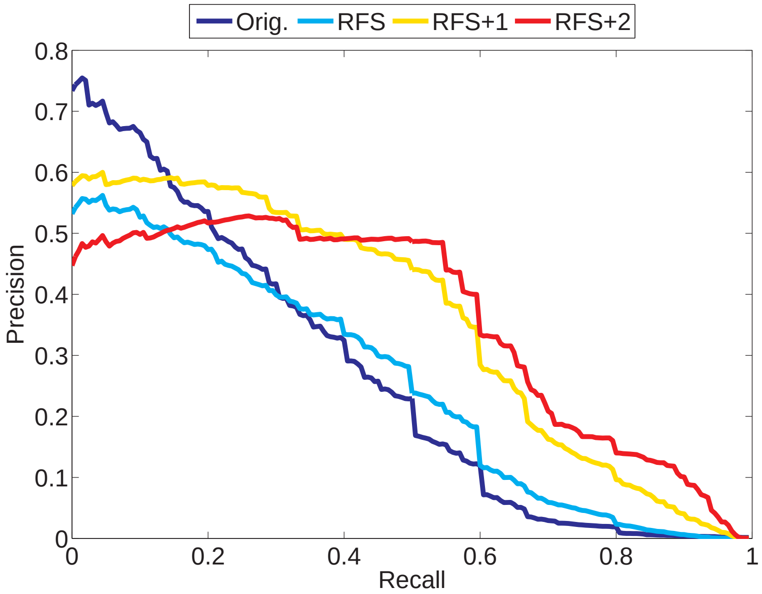

Figure 7.1: Precision-recall curve for Oxford Buildings averaged across all buildings. The introduction of image augmentation (*RFS+1*, *RFS+2*) results in improved precision at higher levels of recall.

7.2.2 Pasadena Buildings

The more challenging Pasadena Buildings showed a smaller improvement from 0.215 to 0.228. As each cross validation fold contains only 5 images of any given building, this further confirms the results from the Oxford Buildings tests in which RFS performs best when a large number of images are present for training. Examination of individual building scores shows that RFS performs performs similarily on average, but that individual building scores can either benefit significantly from RFS, or result in recognition failure. Figure 7.5 shows the diversity of individual building performance found in the Pasadena buildings dataset. As was the case in the Oxford Buildings dataset, RFS performance is dependent on suitable matches being formed in the training phase. As Figure 7.4 shows, some buildings failed to match resulting in features being incorrectly discarded.

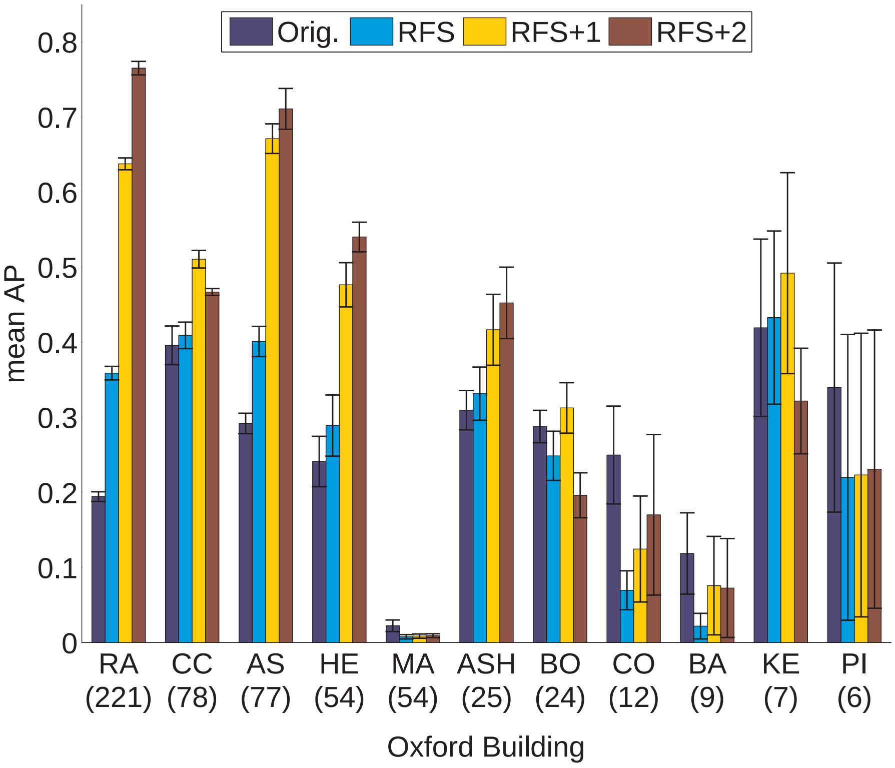

Figure 7.2: Oxford Buildings recognition results broken down by building. The numbers below each building represent the total number of images of that building. Using all features (*Orig.*) underperforms on many buildings when compared to using *RFS*. With the introduction of image augmentation (*RFS+1*, *RFS+2*), buildings with many training images see a significant improvement in recognition. Error bars represent standard error across cross validation folds.

Table 7.3: Query performance by building on the Oxford Buildings dataset. Building values reflect mean AP scores taken on *Good* queries for a given building. Bolded results corresponding to the method with the best performance. *Orig.*—original database, *RFS* —useful feature database, *RFS+1*,*RFS+2*—useful features with image augmentation

Building	Images (Queries)	Singleton: discarded			
		Orig.	RFS	RFS+1	RFS+2
Radcliffe	221 (105)	0.194	0.359	0.638	**0.765**
Christ Church	78 (51)	0.396	0.409	**0.511**	0.467
All Souls	77 (24)	0.292	0.401	0.671	**0.711**
Hertford	54 (35)	0.241	0.289	0.476	**0.540**
Magdalen	54 (13)	**0.022**	0.008	0.009	0.009
Ashmolean	25 (12)	0.309	0.331	0.417	**0.452**
Bodleian	24 (13)	0.288	0.249	**0.312**	0.196
Cornmarket	12 (5)	**0.250**	0.070	0.125	0.170
Balliol	9 (5)	**0.119**	0.022	0.076	0.073
Keble	7 (6)	0.419	0.433	**0.492**	0.322
Pitt Rivers	6 (3)	**0.340**	0.220	0.223	0.231
Average mAP		0.261	0.254	**0.359**	0.358

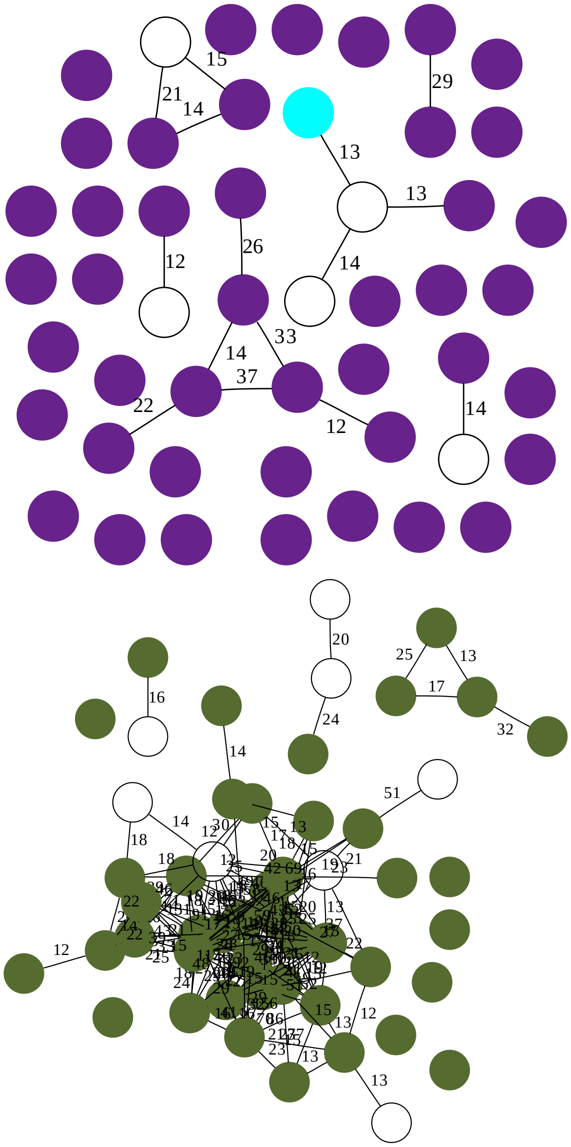

Figure 7.3: Matching graphs for the Magdalen (MA) [*top*] and Hertford (HE) [*bottom*] buildings. An inability to match many of the Magdalen training images caused RFS to fail resulting in many singleton images (30 of 43 building images). Hertford images formed a highly connected matching graph and as a result showed increased benefit from image augmentation.

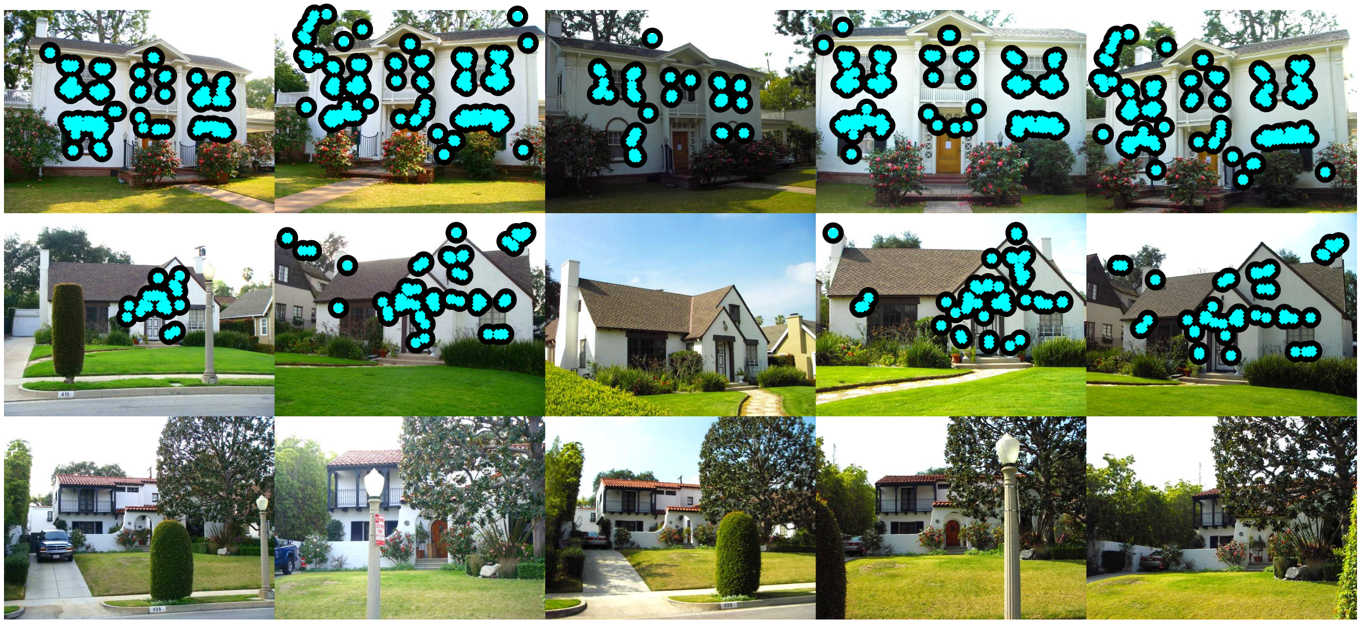

Figure 7.4: Example buildings from the Pasadena Buildings dataset. *Top row:* RFS detects features in all views. *Middle row:* RFS detects robust features in all but one viewpoint. *Bottom row:* Occlusions and lighting changes prevent RFS from detecting features in all views. In this instance, while images 1 and 3 did result in a correct geometric solution, the number of BOW inliers was below the 20 inlier threshold needed for a valid match.

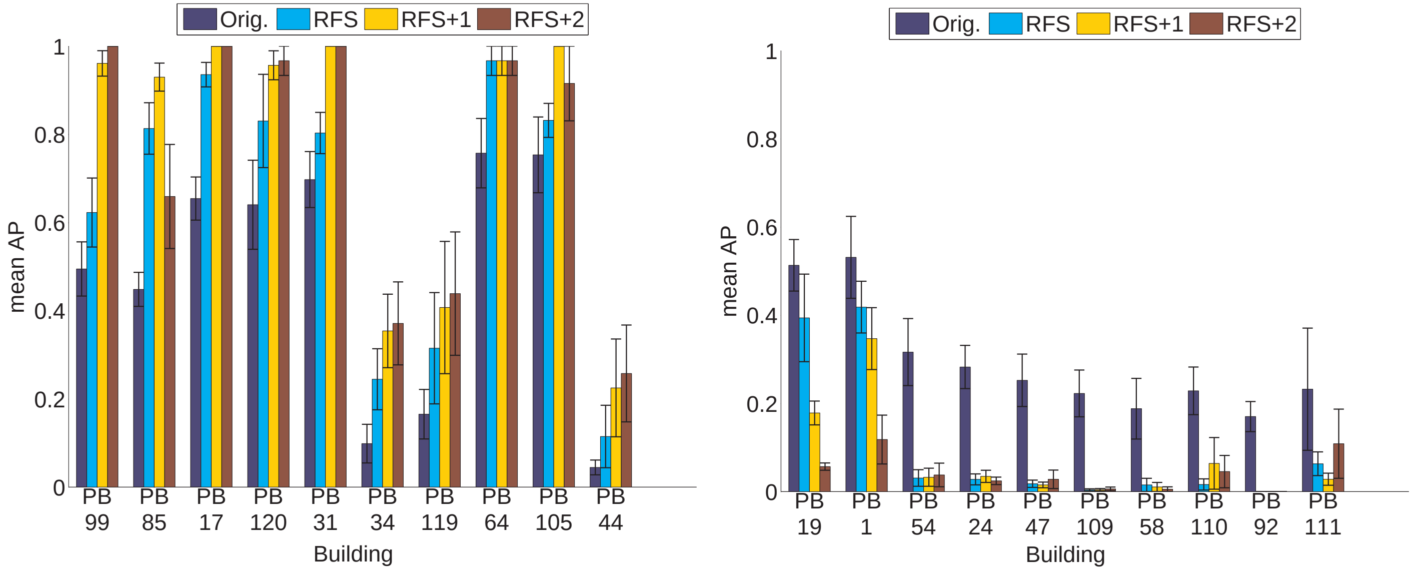

Figure 7.5: A selection of recognition scores for individual buidlings from the Pasadena Buildings dataset. *Left:* Buildings with the largest increase in recognition following RFS. *Right:* Buildings with the largest decrease in recognition following RFS.

7.2.3 Stoplist size

All results presented have made use of a stop-list size of 1% of the visual words. Like the original *tf-idf* ranking, stop-list size impacts RFS performance (Figure 7.6), and while 1% provides good results in both datasets, the Oxford buildings recognition performance peaks at 9%. One positive observation is that the recognition performance following RFS is relatively robust to the initial stop-list size. While the ideal value for a given set of images is data dependent, incorrect values only result in a slight performance degradation.

7.2.4 Feature throttling

The above precision scores for RFS make use of all matched features to represent images. In Section 3 the concept of feature throttling was outlined in which only the best features from each image would be preserved. Figure 7.7 shows that it is possible to maintain improved recognition performance with an even smaller subset of initial image features. Interestingly enough, across both datasets there appear to be no significant gains to extending the representation beyond 300 maximally robust features per image.

7.3 University of Kentucky

The University of Kentucky dataset was used to evaluate the performance of RFS on everyday object images. This proved to be a challenging dataset as each object was represented in only 4 views. Object recognition evaluation was conducted using a score which reflects the average number of the top 4 ranked images which are of the same object. RFS performance is summarized in Table 7.4.

Of the initial 10,200 object images, RFS identified only 6575 using an affine geometry check. As it is important for all images to be matched, the largest scale features from singleton images were preserved yielding a recognition score of 2.84, slightly below the original score of 2.98, using less than 13% of the original image features. The small index is a result of RFS and feature throttling, where image representations were restricted to 200 features per image.

One possible error with unsupervised object detection is highlighted in Figure 7.8. As multiple objects are present in the same location, features from the lo-

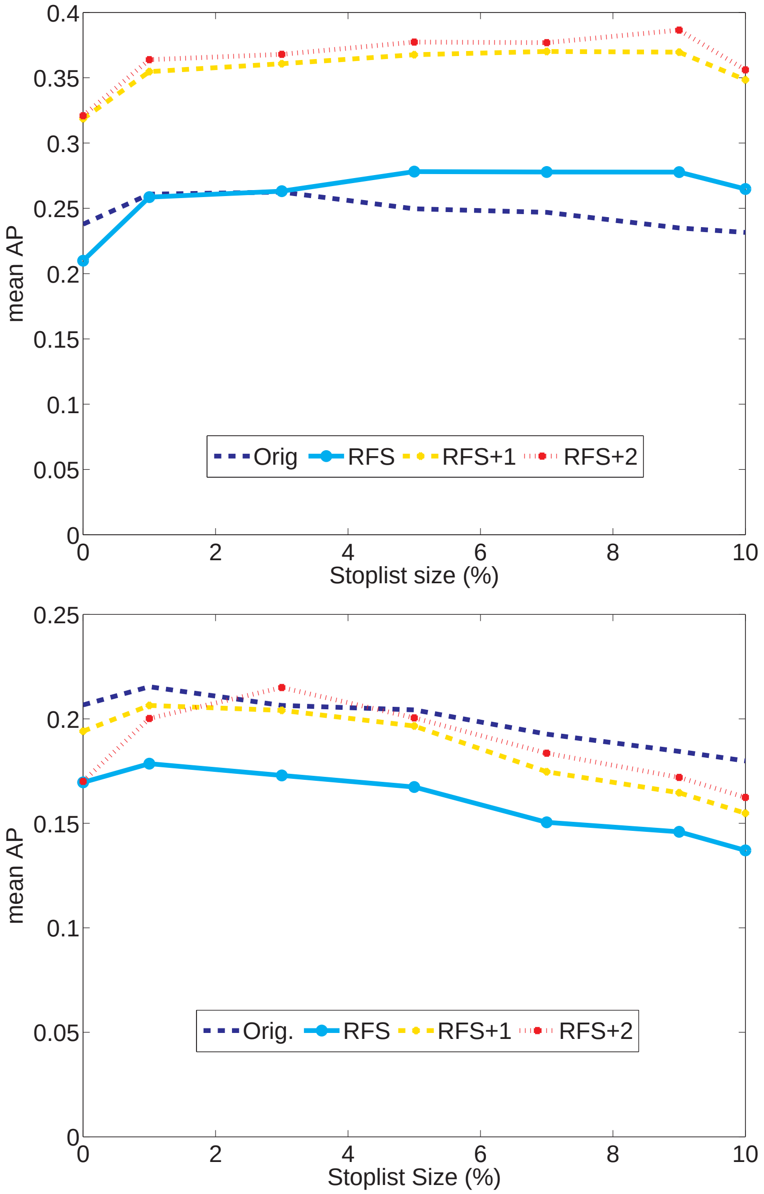

Figure 7.6: The effect of initial stop-list size on RFS performance on the Oxford Buildings (*top*) and Pasadena Buildings (*bottom*) dataset. Changes to the stop-list affect the initial *tf-idf* search, as well as the possible features which can be labelled as robust.

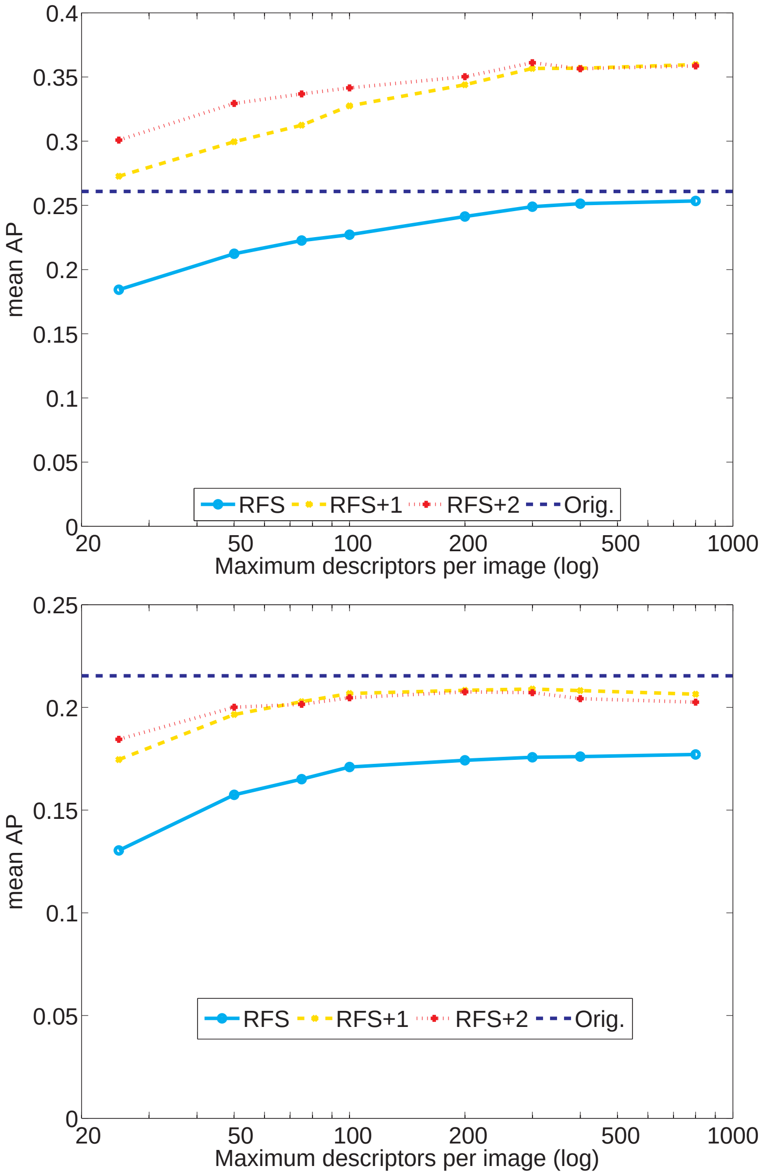

Figure 7.7: Examining the effect of feature throttling on recognition performance on the Oxford (*top*) and Pasadena (*bottom*) building datasets.

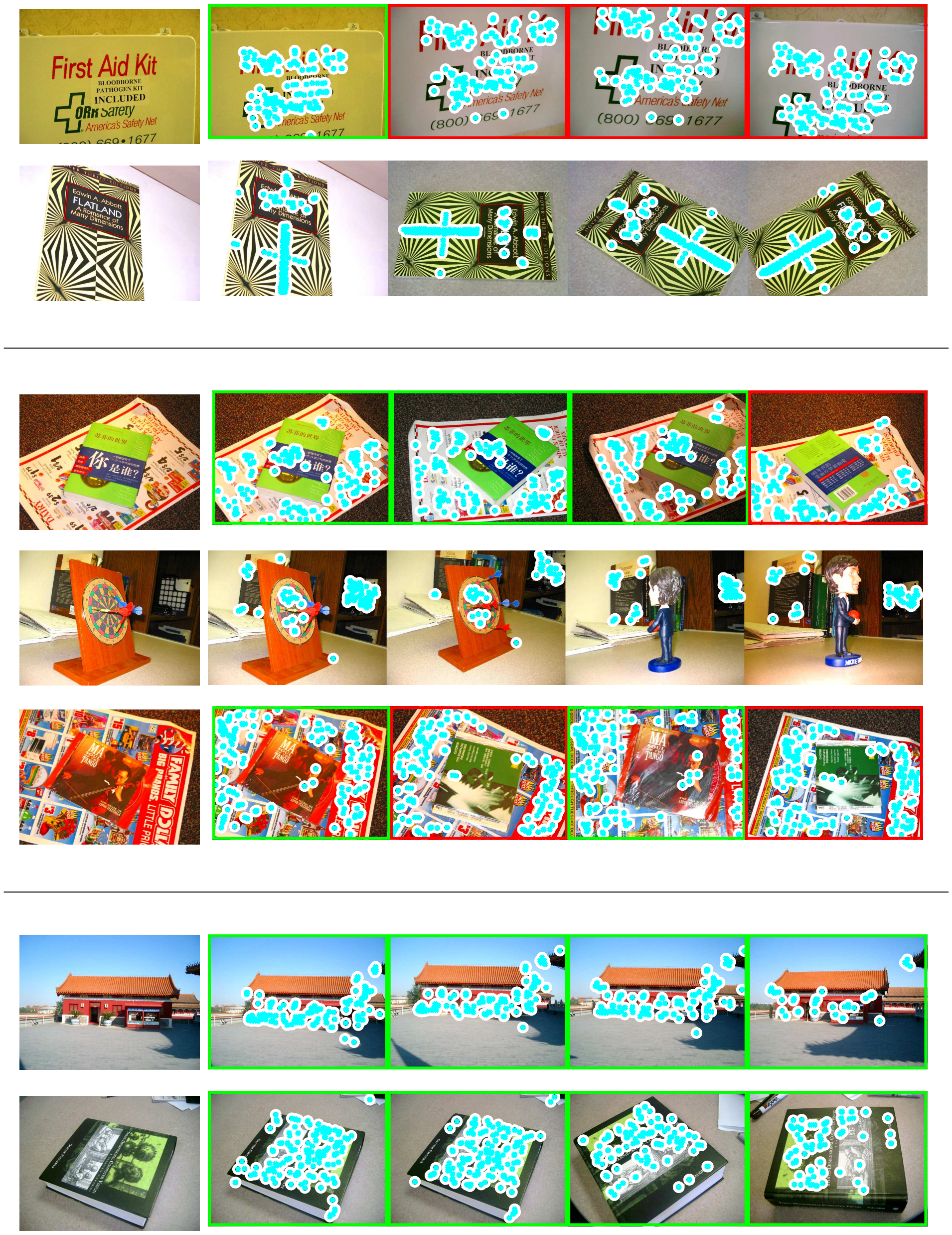

Figure 7.8: Example query images and top 4 matches from the University of Kentucky dataset. *Top:* Mismatches resulting from challenging objects. While these images contain the same object, they are photographed in different conditions and are labeled as different objects. *Middle:* Mismatches caused by RFS identifying the background. *Bottom:* Successful matches.

Table 7.4: RFS performance on the University of Kentucky dataset throttled to 200 descriptors per image. RFS identified objects in 64-65% of images. Preservation of features from singleton images results in comparable recognition to the full set using a fraction of the descriptors.

Method	**Descriptors**	**Recognition (top4)**		
		RFS	**RFS+1**	**RFS+2**
Original	10.7 M	2.98		
Singleton: large scale				
RFS Affine	1.26 M (11.7%)	2.77	2.84	2.81
RFS Epipolar	1.37 M (12.8%)	2.81	2.86	2.79
Singleton: discarded				
RFS Affine	0.61 M (5.7%)	2.27	2.33	2.29
RFS Epipolar	0.71 M (6.7%)	2.29	2.33	2.25

cation have been preserved resulting in a mismatch. Without explicitly tracking matching regions within images, the presence of a uniform background will result some mismatchs when background features dominate the image. Figure 7.8 also highlights some difficult definitions of objects found in the University of Kentucky dataset: the same physical object (e.g., book, first-aid kit) photographed in a slightly changed environment or under different lighting conditions is labeled as a different object.

7.4 Database size

Current state-of-the art methods employ vocabulary sizes ranging from 200,000 to 1,000,000 visual words. All previously reported results make use of a visual vocabulary of 1,000,000 visual words.

In order to establish the effect of varying the size of the visual vocabulary on RFS performance, a vocabulary size of 200,000 visual words was tested. Evaluations of recognition performance conducted on the Oxford Buildings dataset, reported in Section 7.2.1, were repeated making use of the less discriminative vocabulary. Similar to previously published results, the 200,000 visual word case resulted in a decrease in recognition performance, both for the original *tf-idf* matching, as well as for RFS. Figure 7.9, Figure 7.10 and Figure 7.11 show the results for the two cases side by side for comparison.

The positive observation here is that RFS behaviour, relative to the original *tf-idf* matching, does not change with varying vocabulary size. Accordingly, we reccomend the use of a visual vocabulary size of 1,000,000 for performing BOW matching on a large image collections.

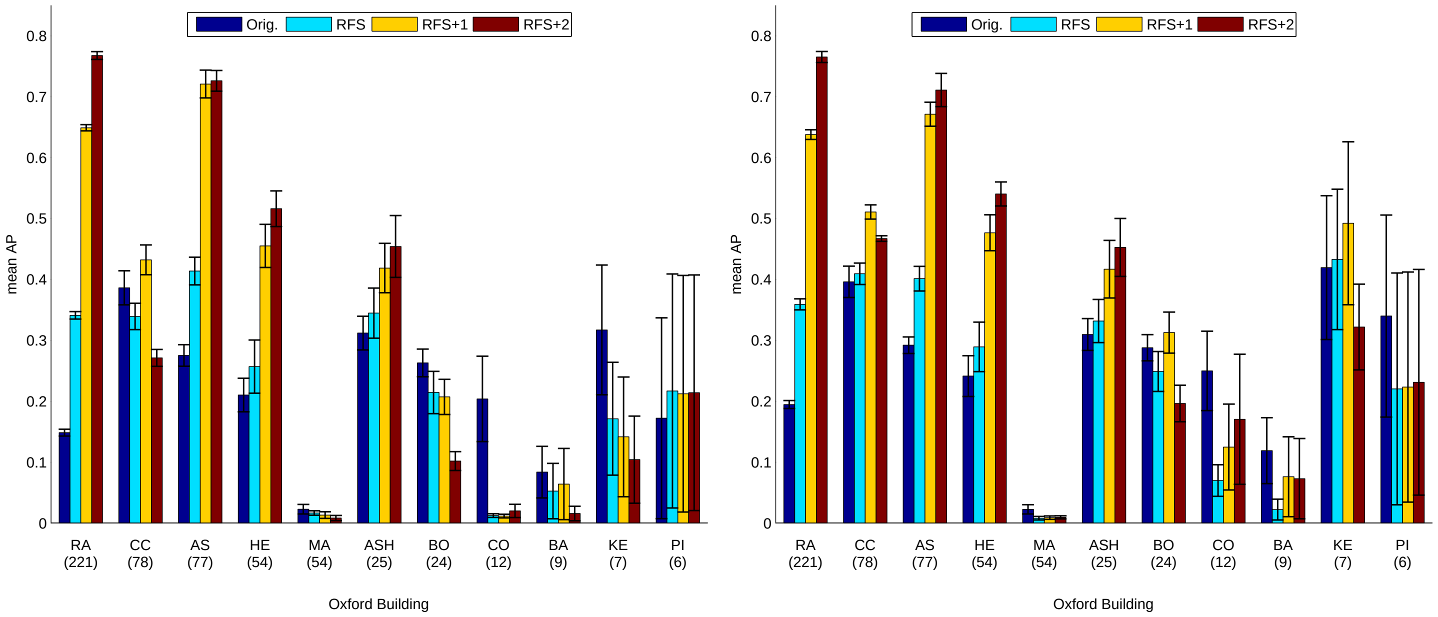

Figure 7.9: Individual building performance when using 200,000 words (*left*) and 1,000,000 words (*right*). While some building scores differ with varying vocabulary size, overall trends reported remain unchanged. [Oxford Buildings]

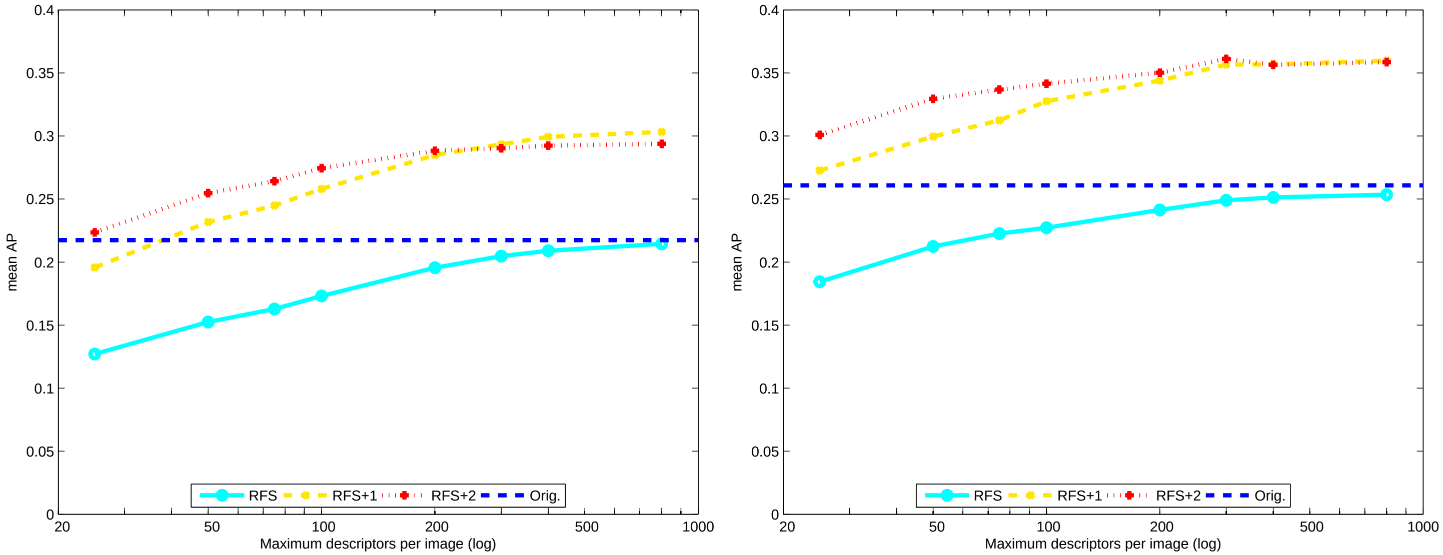

Figure 7.10: Effect of feature throttling on RFS performance when using 200,000 words (*left*) and 1,000,000 words (*right*). Both curves exibit the same rolloff as the number of features is decreased. [Oxford Buildings]

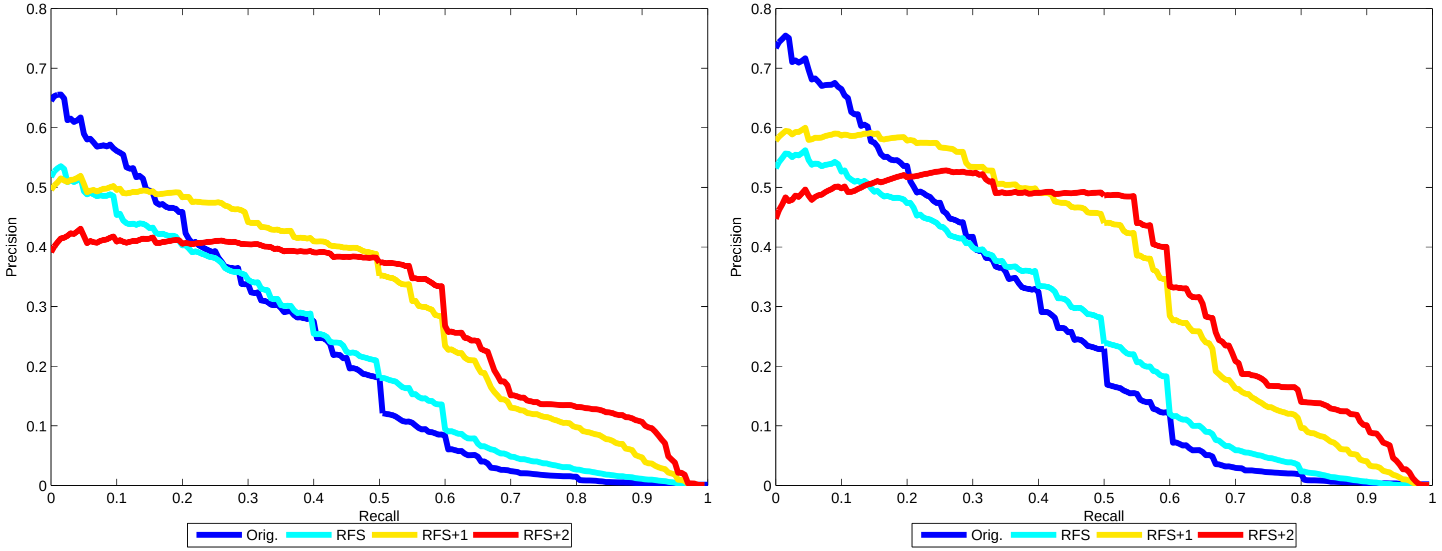

Figure 7.11: Average precision-recall curve for the Oxford Buildings dataset when using 200,000 words (*left*) and 1,000,000 words (*right*). PR-curves for *RFS*,*RFS+1* and *RFS+2* take the same shape with slightly reduced overall performance for the 200,000 case.

7.5 Limitations

While effective at reducing the number of features and maintaing recognition performance, the RFS method suffers from some limitations stemming from the unsupervised object detection.

As was demonstrated with the University of Kentucky dataset, cases where multiple objects are present in the same location can result in only the features from the location being preserved. In the event that feature throttling is employed, RFS will tend to preserve the more commonly occuring object, possibly discarding other objects entirely if they are seldom matched. We argue that for an unsupervised method, this is desireable behaviour as the more commonly occuring objects and locations are preserved.

As we currently do not track regions within an image that match in the image graph, it is not possible to distinguish between multiple objects in an image. This drawback is not specifically due to RFS but rather to the family of BOW based methods which discard spatial layout of image descriptors during the initial match. Partially overlapping images, such as a series of photographs making up a panorama, can result in a series of valid matches relating two images with no overlapping content; a phemonenon we call viewpoint drift. Due to this, use of image augmentation should be limited to smaller values of T (e.g., $RFS+1$ or $RFS+2$), as larger values increase the probability of introducing unrelated content. Figure 7.12 shows an example taken from the Oxford Buildings taken of a statue and window in the Bodleian courtyard. Examination of the matching graph reveals that even this image is connected to images taken of the same courtyard but containing neither object.

It should finally be mentioned that performance of any BOW method is limited to the performance of the underlying image features used.

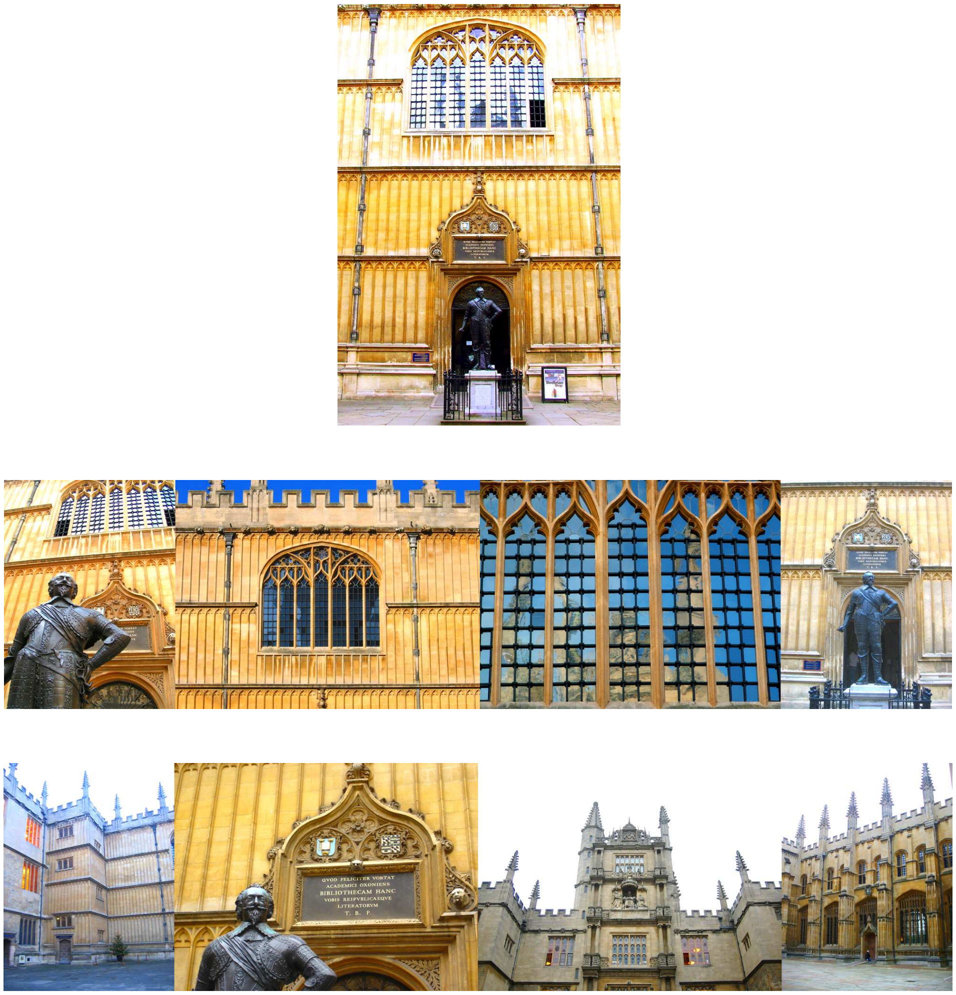

Figure 7.12: *Top:* Image taken in the Bodleain courtyard from the Oxford Buildings dataset (base image). *Middle:* Images directly adjacent to the base image. *Bottom:* Unrelated images connected to the base image via the matching graph.

Chapter 8

Conclusions

We have presented a method for identification of maximally robust 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 using our robust feature selection method can improve recognition performance while reducing memory requirements for image features by as much as 97%. It may seem surprising that recognition is improved by discarding features that fail to match other training images, as at least a few potentially robust features will inevitably be discarded. However, our results show that when sufficient training data is present, 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 at the time of initial 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 relationships within the image matching 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.

The matching 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 unlabelled landmarks. This suggests that the matching graph can be used in other ways to further improve recognition, such as attaching missing labels or correcting mislabelings.

8.1 Future work

There are many possible areas of future work stemming from the work presented in this thesis relating the fields of large scale image retrieval as well as feature selection.

8.1.1 Robust features

While RFS has been shown to be effective on large collections of images containing multiple views of an object, it has limited effectiveness in cases where only single views of objects exist. We believe that the concept of feature robustness can be applied to detecting robust features in single images and should prove to be an interest field of future research.

Recent work on improving the performance of SIFT has shown that the matching performance of SIFT can be improved through the use of synthetic images. Morel and Yu [18] augment the SIFT feature set of an image with SIFT features extracted from affine distorted views of the base image, resulting in a fully affine invariant image representation. We believe that a similar use of synthetic images to detect robust features will result in a compact set of robust features with improved matching performance. These synthetic images are not restricted to affine distortions, but can also be simulated with other methods such as the addition of noise or lens distortion.

8.1.2 Matching graph

We have demonstrated that the matching graph is a useful tool for improving recognition performance even with a simple form of image augmentation. While effective, there is much future work to be done on the formation and application of the matching graph to improve recognition.

Alternate methods for graph construction, such as seeding and growing image clusters [4], can be explored. Inclusion of additional information in the matching graph, such as geometric transforms or matching regions may prove useful for cases where multiple objects exist. More complex generative feature models have been used to improve recognition in query expansion [6] and their application to image augmentation should be explored.

Finally, graph methods which analyze trends and prune the matching graph should be explored. While some work has been conducted on graph clustering using a matching graph [22], the focus was on discovering objects in the images. Applications of graph methods to improve recognition proves to be a interesting field of future research, not only for BOW methods, but also any other methods that are applied to large scale image retrieval.

8.1.3 Large scale image retrieval

As the performance of BOW methods has been shown to vary significantly based on the choice of vocabulary, methods which form better vocabularies should be investigated. To our knowledge only the work of Schindler et al. [26] has investigated learning discriminative vocabularies, however, still rely on a simple unlabelled K-means clustering to form visual words. More complex machine learning methods could yield improved vocabularies, but must be adapted to handle vast amounts of high dimensional data.

Any future work in large scale image retrieval should be guided by practical limitations in computing. Scaling image retrieval to work on the scale of the Internet will require methods that have a low memory foot-print and which can be readily parallelized.

Bibliography

[1] M. Aly, P. Welinder, M. Munich, and P. Perona. Scaling object recognition: Benchmark of current state of the art techniques. In *Proc. IEEE Int. Conf. Comput. Vision Workshops (ICCV Workshops'09)*, 2009. → pages 4, 38

[2] R. A. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley, Boston, MA, USA, 1999. → pages 3

[3] O. Chum and J. Matas. Web scale image clustering: Large scale discovery of spatially related images. Technical report, Czech Technical University in Prague, 2008. → pages 16

[4] O. Chum and J. Matas. Large-scale discovery of spatially related images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32:371–377, 2010. → pages 64

[5] O. Chum, J. Matas, and J. Kittler. Locally optimized ransac. In *Symposium of the German Association for Pattern Recognition (DAGM'03)*, pages 236–243, 2003. → pages 25

[6] 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. IEEE International Conference on Computer Vision (ICCV'07)*, 2007. → pages 13, 14, 15, 19, 35, 37, 64

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*, 2009. → pages 1

[8] S. Gammeter, L. Bossard, T. Quack, and L. Van Gool. I know what you did last summer: object-level auto-annotation of holiday snaps. In *Proc. IEEE International Conference on Computer Vision (ICCV'09)*, 2009. → pages 17, 19, 23

[9] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004. → pages 27

[10] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In D. Forsyth, P. Torr, and A. Zisserman, editors, *Proc. European Conference on Computer Vision (ECCV'08)*, volume I of *LNCS*, pages 304–317. Springer, Oct. 2008. → pages 11, 13, 14, 15, 19, 37

[11] H. Jégou, M. Douze, and C. Schmid. Improving bag-of-features for large scale image search. *International Journal of Computer Vision (IJCV)*, 87(3): 316–336, 2010. → pages 13, 14, 19

[12] B. Kulis and K. Grauman. Kernelized locality-sensitive hashing for scalable image search. In *Proc. IEEE International Conference on Computer Vision (ICCV'09)*, 2009. → pages 8

[13] F. Li and J. Kosecka. Probabilistic location recognition using reduced feature set. In *Proc. IEEE International Conference on Robotics and Automation (ICRA'06)*, 2006. → pages 18

[14] D. G. Lowe. Local feature view clustering for 3d object recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'01)*, pages 682–688, 2001. → pages 32

[15] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004. → pages 2, 24, 37

[16] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *International Journal of Computer Vision (IJCV)*, 60(1):63–86, 2004. → pages 14

[17] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision (IJCV)*, 65(1):43–72, 2005. → pages 37

[18] J.-M. Morel and G. Yu. Asift: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2(2):438–469, 2009. → pages 63

[19] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *Proc. International Conference on Computer Vision Theory and Applications (VISAPP'09)*, 2009. → pages 7

[20] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 2006. → pages 4, 9, 11, 13, 19, 38

[21] M. Perdoch, O. Chum, and J. Matas. Efficient representation of local geometry for large scale object retrieval. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*, 2009. → pages 13, 15, 17, 19, 29

[22] J. Philbin and A. Zisserman. Object mining using a matching graph on very large image collections. In *Proc. IEEE Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP'08)*, pages 738–745, 2008. → pages 16, 17, 64

[23] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07)*, 2007. → pages 4, 10, 11, 13, 14, 15, 16, 19, 29, 37

[24] 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. IEEE International Conference on Computer Vision (ICCV'08)*, 2008. → pages 11, 13, 14, 19, 37

[25] D. Pritchard and W. Heidrich. Cloth motion capture. In *Proc. Conference of the European Association for Computer Graphics*, 2003. → pages 22

[26] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07)*, 2007. → pages 10, 18, 19, 64

[27] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. IEEE International Conference on Computer Vision (ICCV'03)*, volume 2, pages 1470–1477, Oct. 2003. → pages 2, 3, 7, 9, 11, 13, 14, 19