From Images to Scenes: Compressing an Image Cluster into a Single Scene Model for Place Recognition

Edward Johns and Guang-Zhong Yang
The Hamlyn Centre, Imperial College London
ej09@imperial.ac.uk g.z.yang@imperial.ac.uk

Abstract

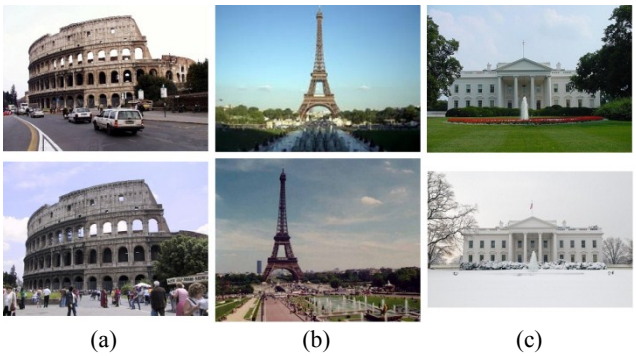
The recognition of a place depicted in an image typically adopts methods from image retrieval in large-scale databases. First, a query image is described as a "bag-of-features" and compared to every image in the database. Second, the most similar images are passed to a geometric verification stage. However, this is an inefficient approach when considering that some database images may be almost identical, and many image features may not repeatedly occur. We address this issue by clustering similar database images to represent distinct scenes, and tracking local features that are consistently detected to form a set of real-world landmarks. Query images are then matched to landmarks rather than features, and a probabilistic model of landmark properties is learned from the cluster to appropriately verify or reject putative feature matches. We present novelties in both a bag-of-features retrieval and geometric verification stage based on this concept. Results on a database of 200K images of popular tourist destinations show improvements in both recognition performance and efficiency compared to traditional image retrieval methods.

1. Introduction

Automatic identification of the place depicted in a single query image is a challenging task due to the instability of images across viewpoint and scale, illumination effects, camera noise, and the dynamic nature of some scenes. In recent years, the growing popularity of photo-sharing websites has provided a vast number of user-generated images from which large datasets can be extracted for research. In particular, image retrieval with collections of popular tourist destinations has been inspired by these datasets for tasks such as recognising famous buildings [1], refining the usability of online collections [2][9], and 3D reconstruction [3]. This paper focuses on the first of these, and explores the following question. Given a query image, how can we efficiently determine the identification of a place depicted in the image, such as a building or structure, by exploring a large database of consumer-generated images?

Early techniques for recognizing the instance of an object [4] or a scene [5] were based on feature-to-feature matching of local invariant features [6, 7]. However, this is not feasible for large-scale search due to the computational cost of comparing many high-dimensional local descriptors. More recently, city-level image retrieval of a few thousand images [8] has been achieved using feature-to-feature matching by prioritising the order of matching, but this remains unsuitable for exploring larger databases with hundreds of thousands or millions of images.

For image retrieval at this scale, methods now typically adopt a two-stage approach. First, features are quantised and images are described as a bag-of-features (BOF), allowing a more efficient means of computing image similarities. The closest k database images to the query image are then passed on to the second stage. Here, geometric verification prunes out false positive feature matches from the first stage. Typically, a query image is compared to every image in the database during the first stage [12], often appending techniques such as query expansion [17] to improve recall. However, whilst image clustering on internet-scale databases has been used for 3D reconstruction or developing semantic representations of databases [2], it is often overlooked for recognition as it is anticipated that a similar-enough individual image will be embedded somewhere in the database. Techniques such as [20] match to iconic images of clusters, but still require the computation of similarity to an individual image, which is susceptible to false positive feature matches from unstable keypoints, as demonstrated in Figure 1.



(a) (b) (c)

Figure 1: Matching local features directly between two images is susceptible to false matches from noisy features or unstable keypoints, due to effects such as (a) pedestrians or vehicles, (b) foreground scene clutter, or (c) long-term dynamic scene changes.

We propose a framework that matches to an entire image cluster, drawing a robust probabilistic model from the rich data associated across clustered images. By learning those features that are likely to re-appear in a query image, the BOF filtering stage can be molded specifically to suit the properties of the scene. Clustering images also helps to filter out "noisy" features by considering their stability across the entire cluster. We also show that by matching to a cluster of several images at once, retrieval time is reduced without impairing recognition performance.

We demonstrate this approach on a database of 200 thousand images representing 200 popular tourist destinations, buildings or structures, acquired from the online photo-sharing website Flikr. We now review the two stages of image retrieval in further detail and their relation to our work.

1.1. Bag-Of-Features filtering

Inspired by text-retrieval developments [10], the BOF method for image retrieval was introduced in [11]. Local features are quantised into a dictionary of visual words, representing discrete partitions in the descriptor space, and normalised histograms of words are computed for each image. Images are compared by computing cosine similarities of these BOF histogram vectors, with each vector element weighted by the inverse document frequency of the respective word, to add importance to discriminative words. One key sacrifice of such feature quantisation is that features representing the same real-world point in space may be assigned to different words. A coarser dictionary reduces this effect, but at the cost of assigning less discriminative power to each word. More recently, this trade-off issue has been addressed with a number of approaches. In [14], a coarse dictionary is complemented by more a discriminative hamming-distance measure between features that are assigned to the same word. Soft assignment was introduced in [13], whereby the impact of each feature is spread over a number of likely words. In [15], this distribution is learned in a more statistical manner.

Whilst these developments reduce the negative effect of quantisation, they still stray from the ideal case whereby a point in real-world space, captured by two different images from similar viewpoints, is assigned to the same partition of feature space. In this paper, we propose a new approach that provides for this without unnecessarily distributing the impact of a feature across multiple words that we know do not correspond to the point. This is achieved by clustering images that represent the same real-world space from similar viewpoints, to form a *scene*, and tracking features across the scene, to form a set of *landmarks*. It is then possible to learn which words a landmark has been assigned to across the scene. The loss of discriminative power by allowing landmarks to span multiple words is reduced by ensuring that clusters are represented by a sufficiently small range of viewpoints. A database of scenes is then formed, with each described by a scene-specific BOF vector representing the expected occurrence rates of these landmarks.

Our approach improves recognition performance for two reasons. First, as stated above, the same point observed in multiple images is assigned to the same set of words that form the respective landmark. Second, the BOF vector is computed by considering only those features that are tracked across multiple images. The rationale behind this is that if a real-world point forms a feature in only one image of a scene cluster, then it is unlikely to appear in a query image representing that same scene. The point can be considered noisy, and appears only once as a result of the instability of its associated keypoint, or due to dynamic objects within the scene. Thus, we only consider matches to points that we already expect to occur, and noisy points that occur only in one image are ignored.

Our approach also speeds up this first stage of image retrieval, again for two reasons. First, query images are matched to a smaller database of scenes, rather than every image that occurs in the database. Second, a very fine vocabulary can be employed, allowing faster lookup in the inverted file system, as each word is linked to a smaller set of plausible landmarks. With previous approaches, if the vocabulary is too fine then landmarks will be prone to multiple assignments across several words. However, in our work, each landmark is explicitly represented by a known set of words, and such multiple assignments are both directly modelled and expected.

1.2. Geometric verification

With the top k images or scenes retrieved from the first stage, it is necessary to ensure geometric consistency across candidate database images, that is not modelled by standard bag-of-features methods. This is typically achieved by estimating an affine transformation between the two images [16], based upon feature-to-feature correspondences, and using an algorithm such as RANSAC to compute inliers for the transformation. The candidate matches to compute the transformation are typically pruned by considering the geometric properties of keypoints [21] or the spatial relationships of features [14].

We propose a new method to prune these candidate features that exploits the expected spatial arrangements of features, learned by considering the distribution of relative positions of neighbouring landmarks across the cluster of images in a scene. Each candidate is verified by considering other landmarks that co-occur most frequently with the candidate. The prior knowledge that the candidate landmark and neighbour landmarks are highly likely to

occur means that a much smaller set of neighbour landmarks needs to be considered, rather than considering all features in the image for verification.

2. Scene Models

2.1. Image clustering

Computing a set of models to represent the database of images requires clustering the images to form a distinct group of visually-similar images, across which features can be tracked. We extract SIFT features [4] and quantise each within a vocabulary tree [12], generated using *k*-means to form a dictionary of 100 million words. We then use soft quantisation [13] of features to compute candidate correspondences across two images. Image clustering is performed across each set of images returned from the search phrase on Flikr.

Most image clustering methods [1, 3] use a RANSAC-based affine transformation computation to verify image geometry and track features across images. We seed the matches for this computation by using a method similar to the spectral technique in [22]. Here, assignment pairs are compared geometrically, and assignments are incrementally accepted or rejected based upon consistency across all other assignments. We use a binary scoring function, setting a similarity of 1 for assignment pairs that agree within a 10° margin for both feature angles and feature orientation differences.

We use hierarchical agglomerative clustering to form a set of scenes, as follows. First, an image similarity matrix M is created, storing the number of feature correspondences between each image pair. We then create a second matrix N, storing the outcome of a binary matching function between each image pair. This function returns 1 if both the number of inlier matches is greater or equal to 7, *and* an affine transformation can be computed between the two images, and returns 0 otherwise. The image pairs are then sorted and processed in order of the number of correspondences in matrix M. Clusters are initially formed by two images pairs when neither of the images has already been assigned to a cluster. Then, if the image pairs are assigned to existing clusters, we compute a cluster similarity and merge the two clusters if this similarity is great enough.

We define the *linkage* between cluster A and cluster B as the fraction of images in A that have a value of 1 in the corresponding element of matrix N, i.e. the fraction of images in A that have been successfully matched to the images in B with an affine transformation. The similarity between A and B is then deemed sufficient to merge the two clusters, is both the linkage between A and B, and the linkage between B and A, is greater than a threshold t. The linkage is required to surpass this threshold in both directions to ensure that each cluster is sufficiently represented by the other. The value of t can be interpreted as dictating the allowable variance in intra-cluster image similarities. A small value will increase the discriminative properties of each cluster but also increase the number of scene clusters, and hence may increase precision but also processing time.

Several values of t were investigated qualitatively by observing the range of images assigned to a cluster. With t less than 0.2, two issue arose that would create difficulties in recognition. First, the range of viewpoints within the cluster was large, and with very low values of t, a chain of images was formed that sometimes covered the full range of viewpoints incident on a scene. The BOF vector for this scene would then suffer from poor discriminative power due to the large variance across the scene. Second, false positive image matches that only matched to a few other images would be included in the scene. This naturally degrades performance because of the introduction of false landmarks to the scene that could later be matched to from a query image.

We chose a value of $t = 0.5$, at which point the scene clusters converged to a representation of a small range of viewpoints. Increasing t further would increase recognition time beyond an appropriate level.

2.2. Landmark generation

Each feature that is tracked across multiple images in Section 2.1 then forms a single landmark. This landmark accumulates the visual words from its constituent features to form a set of words that we expect the landmark to occupy when it is observed in an image. However, due to the fine nature of the dictionary, it is necessary to include those words that lie within the expected descriptor range of the landmark, but were not directly assigned to by the landmark's constituent features. We achieve this by "filling in the gaps" as shown in Figure 2. First, the mean is computed across the word centroids of the landmark's constituent features. Then, the maximum deviation from this mean across all these words is found. Finally, all other features that fall within this deviation are added to the landmark's set of words. In this way, we aim to avoid overfitting of the word distribution by anticipating likely other words that the landmark will occupy.

The probability that a feature corresponding to a landmark is assigned to a particular visual word is an important value used in the Bayesian calculations of Section 4. If we were considering only those words assigned to from the landmark's constituent features, this probability could be computed as the number of occurrences of the word, divided by the total number of features. It is, however, necessary to predict this probability for the new words introduced above without any explicit statistical data from which to compute the probability directly. Therefore, we estimate the probability

by computing a weighted average of the probabilities of surrounding words, with the weight proportional to the proximity of the word centroids. We use a Gaussian weighting of $\sigma^2 = 5000$ and include the $r = 500$ closest words to compute the average. These values are based on the work in [13] and adapted to the visual dictionary we use, which uses 100M words rather than 1M. Probabilities are then normalised across all the words assigned to the landmark.

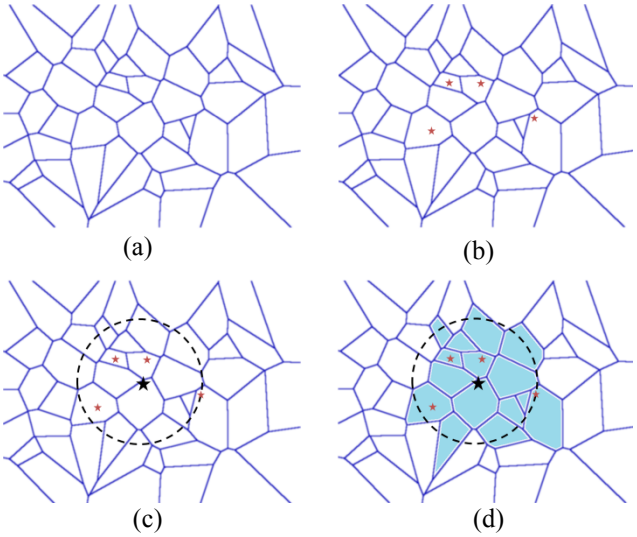


Figure 2: Computing the expected range of visual words assigned to a landmark as a 2-D representation. (a) The visual word dictionary as embedded in feature space. (b) Features corresponding to the same landmark are located in feature space. (c) The mean feature descriptor is computed, together with the maximum distance to the mean across all features. (d) All visual words whose centroids lie within this maximum distance are assigned to the landmark.

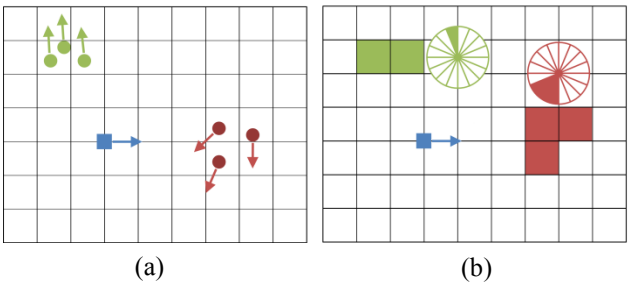


Figure 3: Computing the spatial words between landmarks. (a) The spatial relationships between a landmark (squared) and all co-occurring landmarks (circles) are computed, by considering all features representing the landmark that occur in the same image. The arrows represent the orientations of the features. (b) Spatial words are computed for each co-occuring landmark to represent the range in image space, and range of feature orientation differences, that the landmark's features are expected to occupy.

For each landmark, the co-occurrence statistics are then computed for all other neighbouring landmarks in the scene. The spatial relationship between two landmarks is modelled by a dictionary of three-dimensional *spatial words*. Each word is defined by its minimum and maximum image distance between two features, for both the x- and y-directions, and also the relative orientations between the two features. The image distances are defined as a function of the scale of the feature to retain scale-invariance when modelling the scene. We use 100 divisions in image space for both the x- and y- directions, and 100 divisions in feature orientation space, to give a dictionary of $100 \times 100 \times 100 = 1$ million words. Figure 3 demonstrates the computation of spatial words between features.

3. Bag-Of-Features Filtering

Given a set of scene models, the first stage in recognition of a query image is the computation of similarity between the word frequency vectors of the image, and all scenes in the database. This vector for each scene is computed as the average normalised frequency vector across all views, but votes are only counted for those features that formed landmarks. In this way, noisy features that appeared in the scene cluster due to dynamic objects or unstable keypoints are not considered in the vector, as those features are not expected to appear in another image depicting the scene. We use an inverted file system [11] to efficiently compute these vectors. This is considerably faster than standard image retrieval methods due to the larger dictionary size, because each word in the dictionary has a smaller number of scenes to vote for.

In image-retrieval approaches to recognition, the pipeline then proceeds to take the top k candidate scenes and pass these on for geometric verification. However, the choice of k is often somewhat arbitrary, and is simply set such that the correctly corresponding image is very likely to appear within the top k images. With our method however, we explicitly know the range of BOF vector distances within a single scene. We therefore set a threshold on the BOF vector distance between a query image and a database scene, as the maximum intra-cluster vector distance for the scene. With image-retrieval methods, k is often set to a large value to maximise recall, but many of these false positive candidates can be eliminated in our method prior to geometric verification.

4. Geometric Verification

For each scene, the BOF filtering forms a set of putative feature-to-landmark matches that are now verified geometrically. This is achieved in a two-stage process. First, as is now standard, these putative matches are pruned with weak geometric constraints in a manner similar to [4, 13, 14]. However, because we know the

expected occurrence rates of landmarks and their expected distributions of visual and spatial words, we can achieve this in a probabilistic manner rather than with a voting scheme. The second stage of the verification is a RANSAC-based estimation of an affine transformation between the query image and the candidate scene.

4.1. Pruning putative feature matches

Let us define as $l \in L_y$ a landmark in scene y, and as $n \in N_y$ a different neighbouring landmark in the scene, where N_y is the same set as L_y minus l. For a putative feature-to-landmark mark, the feature in the query image whose visual word matches one of those in landmark l is defined as f_l, and similarly $f_n \in F_N$ is a feature matching neighbour landmark n.

Now, let us define the Boolean variable **L** indicating whether f_l is a true positive match to l, and similarly **N** for the neighbouring landmark. We compute a verification score between f_l and l as the conditional probability that l is present in the query image, given the evidence. This evidence consists of the visual word, w^v_l, assigned to f_l, together with the visual words $w^v_n \in W^v_N$ assigned to all f_n in F_N, and the spatial words $w^s_n \in W^s_N$ assigned to the geometric relationships between f_l and all f_n. The verification score is computed in a Bayesian manner as:

$$p(\mathbf{L}=1 \mid w^v_l, w^v_n, w^s_{ln}) = \frac{p(w^v_l, w^v_n, w^s_{ln} \mid \mathbf{L}=1)p(\mathbf{L}=1)}{\sum_{\mathbf{L}=0,1} p(w^v_l, w^v_n, w^s_{ln} \mid \mathbf{L})p(\mathbf{L})} \tag{1}$$

The value of $p(\mathbf{L} = 1)$ is computed by multiplying the occurrence rate of l in the respective scene with the prior probability of that scene, which is equal across all scenes.

To compute the probability of the evidence, given that $\mathbf{L} = 1$, we must consider the evidence for n arising even when n is not actually present. Therefore, we marginalise the evidence for n over **N**, but with w^v_l depending only on **L**:

$$\begin{aligned} &p(w^v_l, w^v_n, w^s_{ln} \mid \mathbf{L}=1) = \\ &p(w^v_l \mid \mathbf{L}=1) \sum_{\mathbf{N}=0,1} p(w^v_n, w^s_{ln} \mid \mathbf{L}=1, \mathbf{N}) p(\mathbf{N} \mid \mathbf{L}=1) \end{aligned} \tag{2}$$

The value of $p(w^v_l \mid \mathbf{L} = 1)$ is equal to the occurrence rate of visual word w^v_l in the landmark, as computed in Section 2.2. The value of $p(\mathbf{N} \mid \mathbf{L} = 1)$ is equivalent to the co-occurrence rate of landmark n, given that landmark l is present in an image from the scene cluster.

To compute $p(w^v_n, w^s_{ln} \mid \mathbf{L} = 1, \mathbf{N})$, we assume independence between w^v_n and w^s_{ln}, conditional on **L** and **N**:

$$p(w^v_n, w^s_{ln} \mid \mathbf{L}=1, \mathbf{N}) = p(w^v_n \mid \mathbf{L}=1, \mathbf{N}) p(w^s_{ln} \mid \mathbf{L}=1, \mathbf{N}) \tag{3}$$

With $\mathbf{N} = 1$, the two probabilities on the right of Equation 3 are computed by considering the occurrence rate of visual word w^v_n in landmark n, and occurrence rate of spatial word w^s_{ln} between landmarks l and n, calculated across the scene cluster during training. With $\mathbf{N} = 0$, it is necessary to consider the likelihood of w^v_n and w^s_{ln} occurring sporadically in an image, which are treated as independent given that $\mathbf{N} = 0$. The sporadic occurrence rate of w^v_n is computed as the average number of features in an image that contain at least one feature with visual word w^v_n. In this way, rather than using the global occurrence rate of visual words, we account for the fact that some words can exhibit "burstiness" [18] and occur more frequently in a single image than would otherwise be expected. The sporadic occurrence rate of w^s_{ln} is computed by assuming independence between the spatial location of a feature and its orientation, and so we multiply together the two sporadic rates of occurrence of location and orientation. Again, these sporadic rates are learned statistically from the images in the scene cluster.

We now return to Equation 1 to compute the probability that the evidence arises when l is not present. This is similar to the case when l is present, except that $p(w^v_l \mid \mathbf{L} = 1)$ is computed as a sporadic probability:

$$\begin{aligned} &p(w^v_l, w^v_n, w^s_{ln} \mid \mathbf{L}=0) = \\ &p(w^v_l \mid \mathbf{L}=0) \sum_{\mathbf{N}=0,1} p(w^v_n, w^s_{ln} \mid \mathbf{L}=0, \mathbf{N}) p(\mathbf{N} \mid \mathbf{L}=0) \\ &= p(w^v_l \mid \mathbf{L}=0) \sum_{\mathbf{N}=0,1} p(w^v_n, w^s_{ln} \mid \mathbf{N}) p(\mathbf{N} \mid \mathbf{L}=0) \end{aligned} \tag{4}$$

In image retrieval techniques, it is often necessary to consider many neighbouring features when verifying a candidate feature [13, 14]. However, with our approach, we learn the co-occurrence rates of landmarks and their discriminative properties, both in terms of visual word and spatial word distributions. As such, we can select a smaller set of neighbouring landmarks, $n^* \in N_y^*$ such that when these are used together to compute **L**, the overall likelihood of the presence of l is greater or less than a specified threshold.

Let us define each $e^* \in E^*$ as the evidence provided by f_n^*, the feature tentatively matched to n^*. This evidence consists of the visual word of f_n^* and the spatial word between f_n^* and f_l. We compute the overall likelihood that l is present in the query image by treating the presence or absence of each n^* as independent, conditionally on the presence or absence of l:

$$\begin{aligned} p(\mathbf{L}=1 \mid w^v_l, E^*) &= \frac{p(w^v_l, E^* \mid \mathbf{L}=1)p(\mathbf{L}=1)}{\sum_{\mathbf{L}=0,1} p(w^v_l, E^* \mid \mathbf{L})p(\mathbf{L})} \\ &= \frac{p(w^v_l \mid \mathbf{L}=1)p(\mathbf{L}=1) \prod_{e^* \in E^*} p(e^* \mid \mathbf{L}=1)}{\sum_{\mathbf{L}=0,1} p(w^v_l \mid \mathbf{L})p(\mathbf{L}) \prod_{e^* \in E^*} p(e^* \mid \mathbf{L})} \end{aligned} \tag{5}$$

The value of $p(e^* \mid \mathbf{L})$ is computed as in Equations 3 and 4, and thus the overall likelihood of the presence of l can be computed, based on the set of neighbouring landmarks N_y^*.

We select this set by ensuring that $p(\mathbf{L} = 1 \mid w^v_l, E^*)$ is above or below a threshold, after considering all neighbours in the set. This is achieved incrementally by adding each n^* to the set in order of the co-occurrence rate of n^* with l, conditional that l is present. In this way, those neighbouring landmarks out of the full set N_y that are more likely to be present in the image are considered first and act to verify l more strongly than those neighbouring landmarks that are not expected to occur.

Rather than computing $p(\mathbf{L} = 1 \mid w^v_l, E^*)$ at run time, we calculate values of $p(e^* \mid \mathbf{L})$ for all n, and for all expected values of w^v_n and w^s_{ln}. In this way, each landmark can be verified efficiently by looking up the probabilities from memory and updating the value of $p(\mathbf{L} = 1 \mid w^v_l, E^*)$. Once the threshold t has been reached, the outcome of the verification of l is returned. We use a threshold of $t = 0.99$, such that if $p(\mathbf{L} = 1 \mid w^v_l, E^*) > t$, the landmark is positively verified, and if $p(\mathbf{L} = 1 \mid w^v_l, E^*) < t$, it is negatively verified. All positively verified landmarks are then passed on for further global geometric verification.

4.2. RANSAC affine transformation

Given the pruned set of feature-to-landmark matches, geometric verification is now made holistically across the entire image. In standard image retrieval, this is achieved by taking pairs of features that putatively correspond between the query image and the database image. In our case however, it is necessary to compute the locations of these database features from the scene model, rather than any specific image. We achieve this by computing the median image coordinates of each landmark in the scene, and embedding points at these coordinates in a synthesised image. This synthesised image is then used to provide seed correspondences, along with the query image features, for the RANSAC algorithm. The median image coordinates are used rather than the mean to reduce the impact of any incorrect feature matches during landmark generation that might dramatically corrupt the relative coordinates of landmarks.

5. Experimental Results

Images for 200 popular tourist destinations, buildings and structures were acquired by searching for terms in Flikr such as “Colosseum Rome” or “Eiffel Tower Paris”. 1000 images were downloaded for each search, and image clustering was performed on each group to form a set of scenes. A total of 2786 scenes were computed, each of at least two images in the cluster, with an average of 198 landmarks per scene. Figure 4 shows the six clusters with the largest number of images for the search term “Sydney Opera House”. Each represents a distinct viewpoint, but illumination conditions were also apparently a factor in clustering. For testing, a further 100 images were taken for each search term, for a total of 20000 test images. Whilst the database set contained noisy images, any image in the query set that was not suitable (such images included maps, artwork, scale models of buildings, or text) was replaced with a suitable image.

Figure 4: The six clusters with the largest number of images for the Sydney Opera House. Each cluster represents a distinct viewpoint incidental on the place of interest.

We compared our framework to two state-of-the-art image retrieval approaches for place recognition. First, we implemented a full image retrieval framework based on the soft quantisation of [13]. Here, the detrimental effect of quantisation is tackled by assigning each visual word a set of neighbouring words, and including those words when computing feature matches. Second, we implemented the iconic images work of [20]. Here, images are clustered as with our framework, and the image with the greatest number of feature matches across the cluster is defined the iconic image. Then, query images are matched to the iconic images rather than the full database. We again use the soft quantisation of [13] for BOF indexing. For both these two implementations, we took the top 200 images returned from the BOF filtering stage and passed these on for geometric verification. In our method, as discussed before, we only pass on those images that fall

within the maximum BOF distance across the scene cluster.

All three methods used a vocabulary tree [12] for feature quantisation. In the image retrieval implementations, a vocabulary size of 1 million was used, as this has been shown to give best results [21]. For our method however, we use a vocabulary size of 100 million. The quantisation errors that typically harm image retrieval methods are not as apparent in our method because we explicitly compute the expected range of visual words for each landmark. We can therefore choose a larger dictionary to give a finer description of landmark visual words, and to take advantage of the efficiency boost this provides during the BOF inverted file structure [11].

5.1. Precision-Recall

The precision-recall graph is a good indicator of the performance of recognition engines, by demonstrating the ability to accurately recognise the content of an image whilst doing so across a large number of query images. Ideally, it is desirable to have a high precision and a high recall, but naturally as the recall requirement increases, the overall precision drops due to false positives.

For each method, we compute the average precision-recall statistics across all 20000 test images. With each test image, the database images, or in the case of our method, database scene models, were ranked in order of the number of inliers detected in the geometric verification stage. Average precision and recall across all test images was then calculated based on these rankings, and the results are shown in Figure 5.

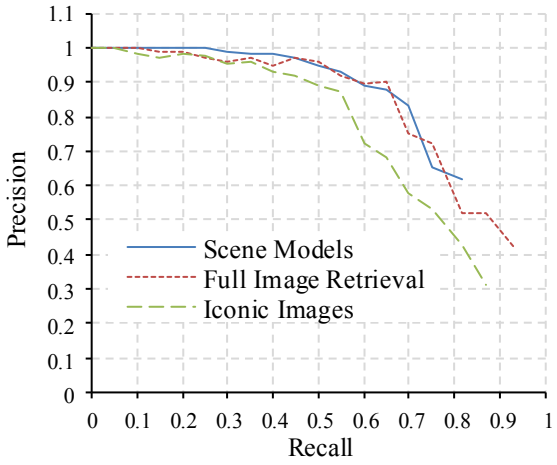


Figure 5: Precision-recall curves for place recognition our method and traditional image retrieval methods on a database of 200K images.

The performance of the iconic images method is the poorest of the three. Given that the iconic images are a subset of the database for full image retrieval, these two methods perform similarly for low recall. However, as recall increases, the lack of a richer dataset causes the precision to drop further than with the full database.

Our method nonetheless outperforms the iconic images method, despite both using the same image clusters for training. Furthermore, it provides a marginal performance increase over image retrieval even using the full database. Particularly at low recall, the precision of our method is promising and allows accurate recognition from a large number of query images. This performance increase can largely be explained by considering that false positive matches in standard image retrieval occur largely for two reasons. First, a match is attempted to all features in an image, despite the fact that typically well over half of those features are never actually matched to another image, even if the viewpoints are very similar. Therefore, these features are susceptible to inducing false positive matches, whilst in our method, these unstable features are never considered for matching. The second reason for the failure of image retrieval is due to quantisation. Whilst using soft quantisation increases the likelihood of achieving a true positive feature match, it also increases the likelihood of introducing a false positive match due to the extra set of visual words that are considered. In our method, we explicitly learn this distribution of words without increasing the range of words to be considered beyond necessary means.

5.2. Efficiency

As our method performs similarly to a full image retrieval approach in terms of recognition performance, it might be argued that the simplicity of an image retrieval framework outweighs the added cost of learning the scene models prior to recognition. However, one of the key benefits of using iconic images rather than a full database is to reduce the redundancy in the database and increase efficiency, and this similarly applies to our method. The computational efficiency was explored across all three implementations, by computing the average recognition time for each stage of the pipeline: feature quantisation, BOF filtering, and geometric verification. The results of this can be seen in Table 1.

Feature quantisation is marginally slower in our method, due to the larger visual dictionary size, but this is negligible with respect to the overall recognition time. Furthermore, this is a constant time and does not increase as the size of the database increases, and so is a less important measure of overall system efficiency.

Compared to the full image retrieval, we achieve a dramatic increase in efficiency for BOF filtering. This is due to the fact that BOF vectors are compared to 2786 scene models, rather than the full 200K database of individual images. Furthermore, our system efficiency even outperforms the iconic images implementation, despite the fact that both methods compute BOF vector

similarities across the same number of candidates. This arises because in our method, we only consider visual words in the BOF vector that occur due to detected landmarks, rather than all the features in an image. The number of landmarks in an image is typically under half the number of features, hence the increase in efficiency by over a factor of two.

Geometric verification, the most time-consuming stage overall, also sees improvements in efficiency with our method. Whilst the RANSAC algorithm for computing the affine transformation is consistent across all techniques, the number of candidate feature correspondences is typically lower in our case. This arises because incorrect candidate correspondences are eliminated more effectively by considering the probabilistic scene model, and using likely co-occurring landmarks for verification or rejection, rather than considering all features equally to eliminate incorrect candidates, as is necessary in standard image retrieval approaches.

	Scene Models	**Full Image Retrieval**	**Iconic Images**
Feature Quantisation	67	48	51
BOF Filtering	3	261	7
Geometric Verification	134	187	221
Overall	204	496	279

Table 1: Comparing the computational time of our method with two image retrieval implementations, in milliseconds, for place recognition in a database of 200K images.

6. Conclusions

We have presented a novel framework to recognise the place depicted in a query image by learning from a database of 200K images acquired from an online photo-sharing website. Database images were clustered into visually-similar scene models, with features tracked across clusters to form landmarks representing real-world points in space. Recognition was then performed relative to these scene models, rather than to individual images as is typically done. The probabilistic models enable a more accurate estimation of the distribution of bag-of-features vectors, and by observing a landmark's descriptor statistics and inter-landmark spatial relationships, a rigorous geometric verification stage is introduced that efficiently verifies feature-to-landmark matches. Recognition performance is similar to or marginally better than full image retrieval approaches, whilst being significantly more efficient.

References

[1] Y.-T. Zheng, M. Zhao, Y. Song, H. Adam, U. Buddemeier, A. Bissacco, F. Brucher, T.-S. Chua, and H. Neven. Tour the World: building a web-scale landmark recognition engine. In *Proc. CVPR*, 2009, pp. 1085–1092.

[2] K. Heath, N. Gelfand, M. Ovsjanikov, M. Aanjaneya, and L. Guibas. Image Webs: Computing and Exploiting Connectivity in Image Collections. In *Proc. CVPR*, 2010, pp. 343 –2429.

[3] S. Agarwal, N. Snavely, I. Simon, S. Seitz, and R. Szeliski. Building Rome in a Day. In *Proc. ICCV*, 2009, pp. 72-79.

[4] D. Lowe. Object Recognition from Local Scale-Invariant Features. In *Proc. ICCV*, 1999, 1150-1157.

[5] F. Li. Probabilistic location recognition using reduced feature set. In *Proc. ICRA*, 2006, pp. 3405-3410.

[6] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, and T. Kadir. A Comparison of Affine Region Detectors. In *Proc. IJCV*, 65(1), 2005, pp. 43-72.

[7] K. Mikolajczyk and C. Schmid. A Performance Evaluation of Local Descriptors. In *Trans. PAMI*, 27(10), 2005, 1615-1630.

[8] Y. Li, N. Snavely, and D. Huttenlocher. Location Recognition using Prioritized Feature Matching. In *Proc. ECCV*, 2010.

[9] N. Stavely, S. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM Transactions on Graphics*, 2006.

[10] R. Baeza-Yates and B. Ribeiro-Neto. Modern Information Retrieval. ACM Press, 1999.

[11] J. Sivic and A. Zisserman. Video Google: A Text Retrieval Approach to Object Matching in Videos. In *Proc. ICCV*, 2003.

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

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

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

[15] A. Mikulik, M. Perdoc, O. Chum, and J. Matas. Learning a Fine Vocabulary. In *Proc. ECCV*, 2010.

[16] R. Hartley and A. Zisserman. Multiple View Geometry in Computer Vision. Cambridge University Press, 2003.

[17] O. Chum, J. Philbin, J. Sivic, M. Isard, A. Zisserman. Total Recall: Automatic Query Expansion with a Generative Feature Model for Object Retrieval. In *Proc. ICCV*, 2007.

[18] H. Jegou, M. Douze and C. Schmid. On the burstiness of visual elements. In *Proc. CVPR*, 2009.

[19] C. M. Bishop. Pattern Recognition and Machine Learning. Springer, August 2006.

[20] X. Li, C. Wu, C. Zach, S. Lazebnik and J-M. Frahm. Modeling and Recognition of Landmark Image Collections Using Iconic Scene Graphs. In *Proc. ECCV*, 2008.

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

[22] M. Leordeanu and M. Herbert. A spectral technique for correspondence problems using pairwise constraints. In *Proc. ICCV*, 2005