Noname manuscript No.
(will be inserted by the editor)

Generative Methods for Long-Term Place Recognition in Dynamic Scenes

Edward Johns · Guang-Zhong Yang

Received: date / Accepted: date

Abstract This paper proposes a new framework for visual place recognition that incrementally learns models of each place and offers adaptability to dynamic elements in the scene. Traditional Bag-Of-Words image-retrieval approaches to place recognition typically treat images in a holistic manner and are not capable of dealing with sub-scene dynamics, such as structural changes to a building façade or seasonal effects on foliage. However, by treating local features as observations of real-world landmarks in a scene that is observed repeatedly over a period of time, such dynamics can be modelled at a local level, and the spatio-temporal properties of each landmark can be independently updated incrementally.

The method proposed models each place as a set of such landmarks and their geometric relationships. A new Bag-Of-Words filtering stage and geometric verification scheme are introduced to compute a similarity score between a query image and each scene model. As further training images are acquired for each place, the landmark properties are updated over time and in the long term, the model can adapt to dynamic behaviour in the scene. Results on an outdoor dataset of images captured along a 7km path, over a period of 5 months, show an improvement in recognition performance when compared to state-of-the-art image retrieval approaches to place recognition.

Keywords Scene Recognition · Image Retrieval · Topological Localization · Appearance-based Localization · Simultaneous Localization and Mapping

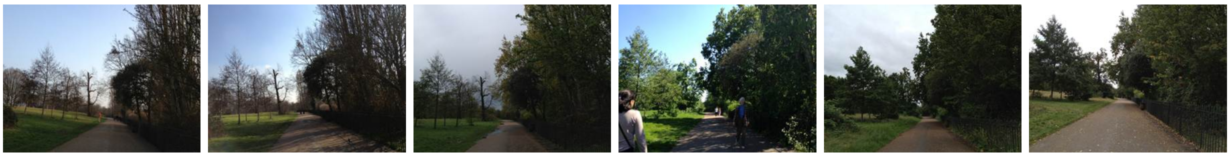
Edward Johns
Imperial College London, UK
E-mail: ej09@imperial.ac.uk

Guang-Zhong Yang
Imperial College London, UK
E-mail: g.z.yang@imperial.ac.uk

1 Introduction

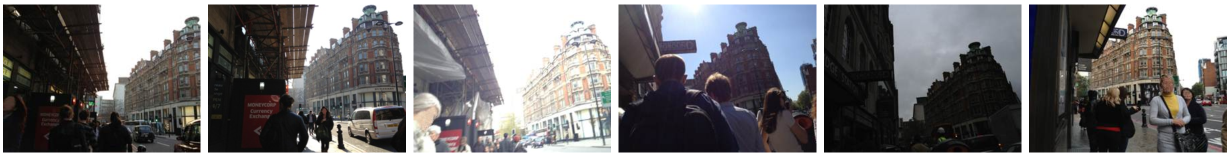
The recognition of a place instance depicted in an image has seen a wide range of applications including object retrieval (Arandjelovic and Zisserman, 2012), loop closure, topological localisation and appearance-based Simultaneous Localisation and Mapping (SLAM) (Johns and Yang, 2013a; Cummins and Newman, 2009), 3D reconstruction (Agarwal et al, 2009) and building recognition for tourists (Zheng et al, 2009; Johns and Yang, 2011a). Typically, the approach to large-scale tasks (Schindler et al, 2007; Nister and Stewenius, 2006) is based on adaptation of image retrieval methods, whereby a query image is compared to all images in a database, each representing a distinct location, to find the closest match. In recent years, efficient matching has been inspired by the Bag-Of-Words (BOW) model (Sivic and Zisserman, 2003) where comparisons of histograms of quantised features select candidate images for stronger geometric verification. In this paper, we present a new framework for place recognition that improves both the BOW filtering and geometric verification components of traditional approaches.

Databases for image retrieval often have significant redundancy due to dynamic behaviour influencing an image. In this paper, we define two types of dynamics: feature dynamics and scene dynamics. Feature dynamics arise due to the instability of a keypoint when the same real-world point is viewed under different viewpoints or illumination conditions. Scene dynamics arise due to long-term structural changes in a scene, such as renovations of building façades or seasonal effects on foliage, and short-term dynamic bodies such as pedestrians or cars. As a result of both these types of dynamic behaviour, many features exist in the database that are never matched to by features in a query image, causing



March April May June July August

Fig. 1: Natural long-term dynamics due to seasonal effects on foliage. The lack of consistent local features between the images causes problems when matching a query image directly to a database image.



March April May June July August

Fig. 2: Example man-made long-term dynamics due to building renovations. Occlusion from short-term dynamics, such as pedestrians, and dramatic illumination dynamics, also causes problems.

significant redundancy and inefficiency in both memory and computational time. Figures 1 and 2 demonstrate two examples of dynamic behaviour in our dataset that cause difficulties for image-retrieval-based recognition, due to a lack of stable features over a period of time.

We propose a model-based recognition framework based on (Johns and Yang, 2011c) that compresses database images into a set of scene models, each representing a place of interest, whilst maintaining the ability to match images from the full range of viewpoints and illumination conditions expressed in the training images. This is achieved by tracking features across multiple images to form a set of *spatio-temporal landmarks*, each representing a real-world point, and learning the distribution of descriptors across the landmark's constituent features. Greater importance can then be assigned to those landmarks that are more likely to appear and those that are assigned to more discriminative descriptors. Similarly, sets of co-occurring landmarks can be learned that co-occur frequently and with rigid spatial relationships.

Feature dynamics are thus accounted for by eliminating those features that occur infrequently, and scene dynamics can be incorporated into the model incrementally by introducing new landmarks into the database as they begin to appear in subsequent visits to a place. This models the dynamics of a scene at the sub-scene level, rather than at the image level as in the case of traditional image retrieval, whereby the entire image must be updated to reflect changes in the environment.

In this paper, we also introduce a new generative BOW filtering stage that learns distributions of visual words rather than a fixed point estimate, and a new probabilistic voting stage for geometric verification that considers all candidate scenes simultaneously, verifying the presence of landmarks in a query image by incrementally incorporating further co-occurring landmarks until a scene has been matched to with sufficient confidence.

1.1 BOW Image Retrieval

Image retrieval based on the BOW model typically involves two main stages. In the first *BOW filtering* stage, a *BOW vector* is created for each image, storing the frequency of visual word occurrences. Vectors are then compared typically by computing their cosine similarity (Sivic and Zisserman, 2003), but other techniques are also available, including min-hash functions (Chum et al, 2008) and Principal Component Analysis (PCA) dimensionality reduction (Jegou and Chum, 2012). Several methods have been introduced to circumvent the effect of feature quantisation (Chatfield et al, 2011), such as soft assignment (Philbin et al, 2008), learning an explicit likely distribution of alternative words (Mikulk and Perdoch, 2010) and computing feature similarities with Hamming distances (Jégou et al, 2010).

In the second *geometric verification* stage, the top k images from the first stage are analysed for geometric consistency between matching features of the two images. Successful attempts have been made to encode weak geometric information in the BOW vectors themselves, by computing several vectors over different spatial windows (Cao et al, 2010; Marszalek and Schmid, 2006), or by including scale and orientation information in the vector (Jégou et al, 2010). Stronger geometric verification is often required for larger-scale searches, which typically involve generating candidate feature correspondences via a Hough-based voting stage (Lowe, 2004) followed by verifying candidates through an estimation of epipolar geometry (Hartley and Zisserman, 2004). Generation of candidates has been addressed by voting for image transformations based on translation, scale and orientation shifts between features (Philbin et al, 2007; Zhang et al, 2011), and with additional spatial weighting between voting bins (Tolias and Avrithis, 2011). Query expansion can also help to increase the level of recall (Chum et al, 2011).

1.2 Model-Based Place Recognition

Model-based place recognition has been successfully applied to small indoor environments (Ni et al, 2009; Pronobis and Caputo, 2007), but large-scale modelling has not been addressed in this way. Attempts to improve the efficiency of retrieval have included matching to iconic images of a scene (Raguram et al, 2011), but these still require direct image-to-image matching, and as such feature redundancies remain. The work in this paper is related to the approach in (Johns and Yang, 2011a), whereby scene models are learned from image clusters, and we expand upon this to demonstrate incremental learning and adaptation to dynamic scenes.

Existing approaches to learning of dynamic scenes typically adopt an incremental approach to Support Vector Machine (SVM) classification (F. Orabona and Caputo, 2010; Luo et al, 2007). However, online training of SVMs remains computationally expensive and is not suitable for real-time applications such as robotics. Furthermore, these approaches are applied to small indoor training sets where discriminative methods are suitable, whereas for large-scale recognition this level of complexity is often not viable.

Direct feature-to-feature matching approaches (Se et al, 2001; Lik and Kosecka, 2006) have been successful on small-scale databases, and more recently this has been speeded up by considering the order of feature matching (Li et al, 2010), but these methods still require expensive computation of feature-to-feature descriptor distances. Feature tracking to extract stable features has been applied previously in simple frameworks (E. Arnaud and Verri, 2006; Li et al, 2010). However, none of these methods can learn feature descriptor distributions in a robust probabilistic manner, nor do they exploit the observed spatial relationships between features as they are tracked.

1.3 Key Contributions

In this paper, we present three key technical contributions within our place recognition system. They include:

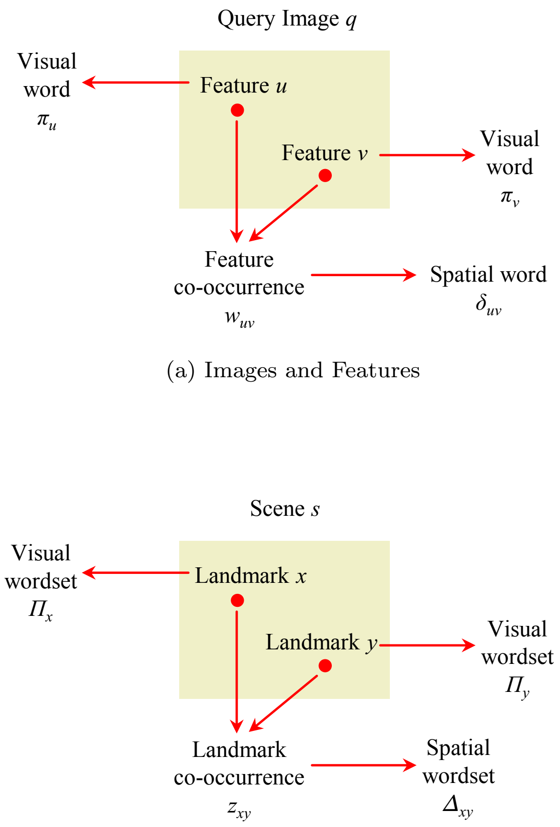
- a new *generative BOW* filtering stage based on the mean and variance of elements in the BOW vector learned over several training images;
- a new *probabilistic voting* system for geometric verification of features based on quantisation in both feature and image space;
- the ability to adapt to dynamic environments where objects are moving into or out of the scene.

2 The Scene Model

In this section, we introduce the model used to describe each location in a map of the environment. We provide definitions for key components of this model, and then describe how the models are learned from a set of training images representing a scene.

2.1 Definitions

- A *query image*, q, is an input image which is to be recognised.
- A *scene*, s, is a model of a place's appearance stored in the database or map.
- A *landmark*, x, is a real-world point in 3D space that is observed in an image viewing the point.
- A *neighbouring landmark*, y, is another landmark that co-occurs in at least one training image as x.
- A *feature*, u, is a local region in an image detected at a corner, that is described by its scale, orientation and visual word. An observed landmark causes a feature to appear on an image. We used Scale-Invariant Feature Transform (SIFT) features in our experiments (Lowe, 2004).
- A *neighbouring feature*, v, is another feature that appears in the same image as u.
- A *landmark co-occurrence* z_{xy} is a co-occurring pair of landmarks, x and y.
- A *feature co-occurrence* w_{uv} is a co-occurring pair of features, u and v.
- A *visual word*, π_u, is a quantised portion in feature space that describes the texture in the local area surrounding feature u, such as is used in the BOW model (Sivic and Zisserman, 2003).
- A *spatial word*, δ_{uv}, is a quantised portion in image space that describes the geometric relationship of feature co-occurrence w_{uv}, defined by the angle, distance (to scale), orientation difference, and scale difference, between the two features.



(a) Images and Features

(b) Scenes and Landmarks

Fig. 3: Notation used in this paper.

- A visual wordset, Π_x, is the set of visual words assigned to landmark x from its observed features.
- A spatial wordset, Δ_{xy}, is the set of spatial words assigned to landmark co-occurrence z_{xy}, from its observed feature co-occurrences.

Figure 3 demonstrates the notation that is used in this paper, showing parallels between components of an image and components of a scene.

2.2 Spatial Words

To enable fast recognition with low memory requirements and a tractable probabilistic model, the geometric relationship between two features is quantised into discrete spatial words. The spatial dictionary can be considered as a regular grid of $a \times b$ squares on an image, relative to a particular feature we are evaluating, with each grid square further quantised into c scale divisions and d orientation divisions, to form a dictionary of $a \times b \times c \times d$ spatial words. Each square represents the relative x- and y- distances between two features, the scale division represents the ratio of scales between the two features, and the orientation division represents the difference in orientation between two features.

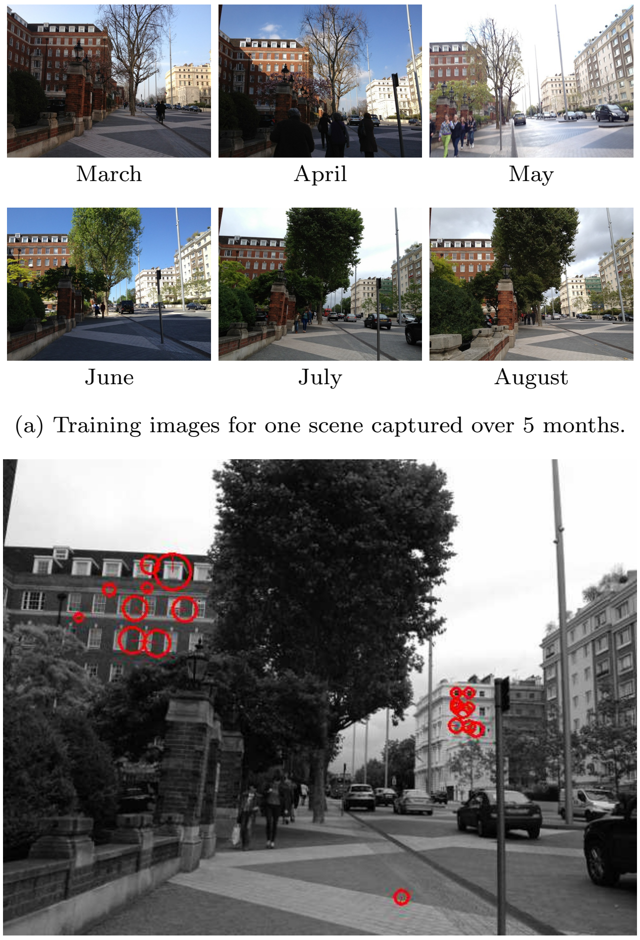
2.3 The Scene Model

We follow the approach of (Johns and Yang, 2011c) and represent every location by the appearance of its scene s, with each scene in the map described by a set of landmarks X and their geometric relationships. Each landmark is described by its visual wordset, the set of visual words over which the landmark has been observed in training images. Each co-occurrence of landmarks is described by its spatial wordset, the set of spatial words over which the two landmarks have been observed in training images. Given a query image, the task then becomes to find candidate matches between query features and scene landmarks based on their visual words, and then verify these candidates based on their spatial words.

2.4 Generating Landmarks

A landmark is formed by finding feature correspondences across a set of training images. Those features that are detected in at least two images are retained, and the entire track of correspondences then forms a landmark. This stage is performed for three reasons. First, many features in a scene are unstable, due to dynamic bodies in the scene or weak image gradients, and will not be observed again in further images of the scene. Second, locating each landmark independently is necessary to enable geometric verification and learning dynamic properties of landmarks if more than one feature in an image is assigned to the same visual word. Otherwise, if only the raw statistics of visual words was recorded, then two landmarks represented by the same visual word will corrupt each other's respective statistics. Finally, tracking across multiple images in a dense topology, and learning a generative model of each landmark, allows for unobserved locations, that exist between observed locations, to be incorporated within this model, as shown in (Johns and Yang, 2011b).

Figure 4 demonstrates learning landmarks from training images captured over a period of 5 months. Whilst standard image-retrieval-based recognition engines would be confused by dynamic scene elements such as those from pedestrians and leaves in the trees, our approach filters out these dynamics and focuses on static elements. As can be seen in this figure, the most likely landmarks are typically detected on the sides of



(a) Training images for one scene captured over 5 months.

(b) The most stable landmarks for this scene, with observation probabilities greater than 0.5.

Fig. 4: Detecting landmarks from a set of training images over a period of time enables filtering out of dynamic bodies and learning which landmarks are most stable.

buildings, rather than on the foliage which changes in appearance over time.

Several methods exist for computing the necessary feature correspondences (Tolias and Avrithis, 2011; Li et al, 2010; Leordeanu and Hebert, 2005), which typically involve a coarse filtering out of inconsistent features, followed by a stricter rigid stage such as computing a homography between the images, based on the candidate feature correspondences from the first stage. We use the Hough Pyramid Matching method (Tolias and Avrithis, 2011) to generate candidates, followed by a Random Sample Consensus (RANSAC) (Hartley and Zisserman, 2004) to compute strict inliers. Initial candidate matches are formed by using soft assignment in the visual dictionary to account for the same landmark being assigned to different words under varying illumination and viewpoint conditions. As in (Philbin et al, 2008), we therefore assign five ”soft” visual words to each word in the dictionary and form candidate matches to a word whenever one of its soft words is detected in a different image.

3 Generative BOW

In this section, we present the first stage of our recognition system. BOW filtering stages typically compare images by computing the cosine similarity between their BOW vectors (normalized vectors of word visual word frequences) (Sivic and Zisserman, 2003). However, comparing explicit BOW vectors in this way is sensitive to dynamic scenes where the vectors may change over time. One solution is to average the BOW vector to eliminate dynamic features (Johns and Yang, 2011c) and compare the average BOW vector to the query. A more advanced approach is to consider variances and co-occurrence statistics of visual words rather than a simple point estimate. Discriminative BOW models have demonstrated this in object classification (Csurka et al, 2004) and image retrieval (Arandjelovic and Zisserman, 2012). However, the BOW filtering stage should not discriminatively classify query images before any geometric verification has taken place; it exists to efficiently generate a smaller set of candidate scenes.

We propose that if several training images are available for a scene, then a generative model can be learned from the visual word distributions across these images, which both filters out dynamic features and allows for a deeper representation of visual word statistics. First, in the BOW vector for a scene, we include visual word counts only from those features that have been tracked across two or more training images. Contributions from other features are assumed to be from noisy or dynamic bodies, and as such are much less likely to be represented in another BOW vector of the scene. A further bonus from this is that comparison of BOW vectors is far more efficient, as we only compare a very small number of elements which correspond to the landmarks in the scene. Second, we consider the variance in the BOW vector, rather than taking a point estimate at the mean, or considering point estimates from all training images independently. By building a generative model that incorporates both the mean and variance, we are able to predict more accurately the likelihood that a scene may produce the observed BOW vector in a query image.

Consider the example in Figure 5, where circles represent BOW vectors for images of two different scenes, and the square is our query image. Using the standard cosine similarity measure, the query would be assigned to the blue scene - whether we use the nearest-

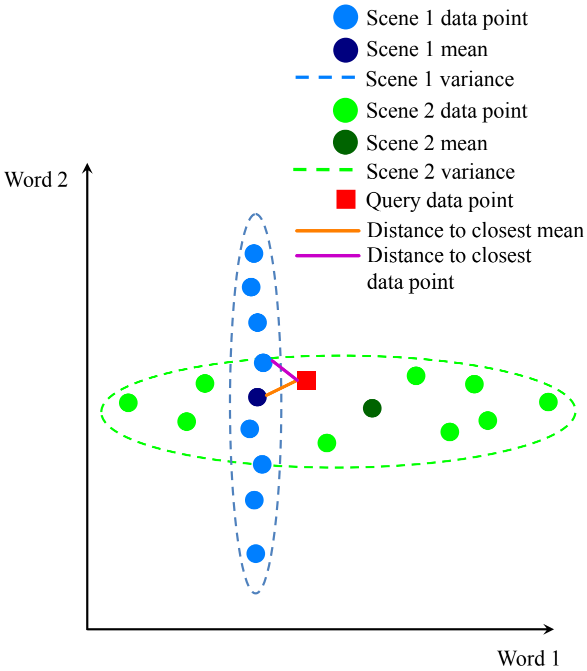


Fig. 5: Our generative BOW similarity score takes into account the variance of visual word frequencies in an image, rather than just a point estimate. The query data point is most appropriately matched to the green scene when the variance is considered, whereas it is assigned to the blue scene when the variance is overlooked.

neighbour individual image or the scene average. However, it is clear that due to the blue scene's low variance in the 'Word 1' direction, the query image is in fact much more likely to be representative of the green scene. Our method considers this variance and correctly assigns the query to the green scene due to its wider variance in the respective direction.

Let $\mathbf{a}_q$ represent the standard normalised (L2-norm) BOW vector for query image q, with *tf-idf* weighting (Sivic and Zisserman, 2003). Then, let $\mathbf{b}_s$ be the generative BOW vector for scene s, which expresses both the mean and variance in the vector, in a normally-distributed manner, across all training images for the scene. The BOW score, $S_{\text{bow}}(q, s)$, is computed by considering how likely $\mathbf{a}_q$ is to be generated from scene s, compared with all other scenes:

$$S_{\text{bow}}(q, s) := p(q \mapsto s | \mathbf{a}_q, \mathbf{b}_s) \tag{1}$$

$$= \frac{p(\mathbf{a}_q | q \mapsto s, \mathbf{b}_s)}{\sum\limits_{s \in S} p(\mathbf{a}_q | q \mapsto s, \mathbf{b}_s)} \tag{2}$$

where $q \mapsto s$ indicates that q is an observation of s, and we assume that all scenes have equal probability (the global localisation application).

The likelihood of observing $\mathbf{a}_q$ given that q represents s is then computed by considering the probability density function of scene s's normally-distributed BOW vector:

$$p(\mathbf{a}_q | q \mapsto s, \mathbf{b}_s) = \frac{\exp\left(-\frac{1}{2}(\mathbf{a}_q - \boldsymbol{\mu}_s)^{\mathrm{T}} \boldsymbol{\Sigma}_s^{-1} (\mathbf{a}_q - \boldsymbol{\mu}_s)\right)}{(2\pi)^{\frac{n}{2}} |\boldsymbol{\Sigma}_s|^{\frac{1}{2}}} \tag{3}$$

where $\boldsymbol{\mu}_s$ and $\boldsymbol{\Sigma}_s$ are the mean vector and covariance matrix components of $\mathbf{b}_s$, and n is the number of visual words in the dictionary. Ideally, we would like to be able to store the full covariance matrix, but this is not possible due to the vast memory and computational requirements. For a 100K visual dictionary, each scene would require ~20 gigabytes of memory, which is clearly impractical. One solution would be to store only the largest set of covariances for each word and compute Equation 3 based on these. However, this suffers from overfitting as co-occurrence rates are typically much lower than occurrence rates of individual landmarks. Furthermore, it is not so important to model the co-occurrence statistics at the BOW filtering stage, as these are then brought into play in the later geometric verification stage. For these reasons, we store only the diagonal elements of $\boldsymbol{\Sigma}_s$ and forego the covariances, and calculate the probability of each element in $\mathbf{a}_q$ based only on its own variance in scene s.

4 Probabilistic Landmark Voting

In this section, we present the second stage of our recognition system: geometric verification by probabilistic landmark voting. For each candidate scene s that passes through the Generative BOW filter, a geometric score is computed, $Score_{\text{geo}}(q, s)$. This score reflects the geometric consistency between features U_q in query image q and landmarks X_s in scene s.

We define the geometric score as the normalised sum, over all landmarks in the scene, of the *landmark observation probability*, $p(u \mapsto x|E)$, which is conditional on *landmark evidence*, E. This is the probability that landmark x has been observed in q as feature u, and we take the maximum of this probability over all features in the query image:

$$Score_{geo}(q, s) := \frac{\sum\limits_{x \in X_s} \max\limits_{u \in U_q} p(u \mapsto x | E)}{\eta} \tag{4}$$

The normalising term η is the average number of landmark observations in s's training images. The score is similar to the standard method in image retrieval of counting inlier feature correspondences, but in our case we weight each count with a score relating to the probability that the feature-to-landmark correspondence is correct, and the normalisation ensures that scenes with fewer stable landmarks are not penalised in the score.

4.1 Landmark Evidence

For each query feature u, an inverted index is used to form a set of *candidate landmarks* X^c whose visual wordsets contain the query feature's visual word. For each query feature u that forms candidate landmarks, the landmark evidence E represents all features in the query image that are used to calculate the landmark observation probability in Equation 4. It is broken down into two components: e_x, the evidence provided by feature u, and the set E_Y, the evidence provided by neighbouring features V which are candidate matches to neighbouring landmarks Y_x. E_Y in turn is broken down into one component, e_y, for each neighbouring landmark. y. If a neighbouring feature v's visual word matches y's visual wordset, and feature co-occurrence w_{uv}'s spatial word matches landmark co-occurrence z_{xy}'s spatial wordset, then evidence e_y is set to this feature v. If there are no features matching y, then e_y is set to $\emptyset$, indicating that the landmark co-occurrence z_{xy} has not been observed.

As an example, consider Figure 6. Feature u and landmark x form a candidate correspondence based on consistency of visual words. Evidence e_x is therefore defined by feature u. Now, x's neighbouring landmarks y_1 and y_2 form candidate matches with u's neighbouring features v_1 and v_2 respectively, based on consistency of visual words. Further, the geometric relationship between u and v_1 matches the geometric relationship between x and y_1, based on spatial words; therefore, e_{y_1} is set to feature v_1 as a candidate match to co-occurrence z_{xy_1}. However, the geometric relationship between u and v_2 does not match the geometric relationship between x and y_2. Therefore, e_{y_2} is set to $\emptyset$, and the co-occurrence z_{xy_2} is defined as absent. In this way, when calculating the observation likelihood of a landmark, geometric matching is localised and all geometric constraints are relative to that landmark. As such, only the relative geometry is of importance - not the absolute position in the image - and this allows for viewpoint-invariant recognition.

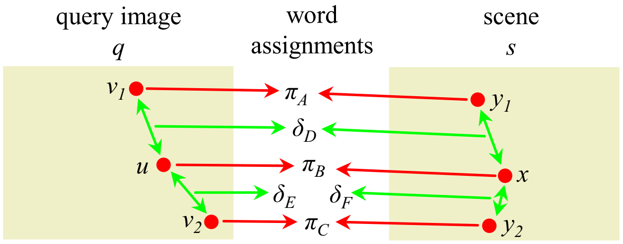


Fig. 6: Feature u forms a candidate match to landmark x based on visual word consistency. Additionally, neighbouring features v_1 and v_2 form candidate matches to neighbouring landmarks y_1 and y_2. In terms of spatial words, v_1 is consistent with y_1, whereas v_2 is not consistent with y_2. Therefore, evidence e_{y_1} is set to feature v_1 as a candidate match to z_{xy_1}, whilst evidence e_{y_2} is set to $\emptyset$.

4.2 Landmark observation probability

Consider that u may be a visual word match to several candidate landmarks in difference scenes, and it is necessary to find which candidate landmark is most likely to have been observed. The *prior match probability*, $p(u \mapsto x)$, and the *evidence likelihood*, $p(E|u \mapsto x)$, are therefore calculated for each of these candidate landmarks to form a probabilistic score. Furthermore, it is necessary to consider the probability $p(E|u \mapsto \emptyset)$ that the evidence is observed when u is in fact not a true match to any of the candidate landmarks. The landmark observation probability can then be expressed as:

$$
p(u \mapsto x|E) = \frac{p(E|u \mapsto x)p(u \mapsto x)}{\left(\sum_{x \in X^c} p(E|u \mapsto x)p(u \mapsto x)\right) + p(E|u \mapsto \emptyset)p(u \mapsto \emptyset)} \tag{5}
$$

4.3 Prior Match Probability

The prior probability that a random feature in the query image is a true match to landmark x takes into account three factors. First, the landmark observation likelihood $p(x|s)$, which is the likelihood that the landmark will be observed in an image representing scene s, reflecting the stability of the landmark. Second, the number of features in the query image, $|U_q|$. The fewer features in the image, the more likely that a random feature is the one representing the landmark. Third, the prior probability of observing the scene, $p(s)$, which we assume to be equal across all scenes in S for a global localisation application. Combining these three components together then yields the prior match probability:

$$p(u \mapsto x) = p(x|s)\frac{1}{|U_q|}\frac{1}{|S|} \tag{6}$$

4.4 Evidence Likelihood

For a fully probabilistic calculation of Equation 5, the evidence $E = \{e_x, E_Y\}$ for each candidate landmark must contain the same features, such that the evaluation of $p(u \mapsto x|E)$ is based upon the same data. Now, e_x is already set to the same feature u that is a candidate match to each landmark in X^c. Then, for each candidate landmark x, we divide E_Y into two components: E_{Y_x}, the candidate features matching to x's neighbouring landmarks, which we call the *neighbourhood evidence* for x, and $E_{Y_{x'}}$, the neighbourhood evidence for all other candidate landmarks x', which we call the *competing landmarks*. Finally, the evidence likelihood for a given candidate landmark can be expressed as:

$$\begin{aligned} p(E|u \mapsto x) = p(e_x|u \mapsto x) \times p(E_{Y_x}|u \mapsto x) \\ \times \prod_{x' \in X^c, x' \neq x} p(E_{Y_{x'}}|u \mapsto x) \end{aligned} \tag{7}$$

where the evidence likelihood of e_x is based upon the visual words assigned to u:

$$p(e_x|u \mapsto x) = p(\pi_u|\Pi_x) \tag{8}$$

4.4.1 Neighbourhood evidence for x

To calculate the likelihood of x's neighbourhood evidence, E_{Y_x}, we adopt a naive Bayes model and assume independence between each neighbouring landmark of x, $y \in Y_x$, given that x is present. If there are multiple candidate features that match to any y with both visual and spatial words, then the feature is taken whose visual and spatial words are the most likely to have been observed in a feature representing y:

$$p(E_{Y_x}|u \mapsto x) = \prod_{y \in Y_x} \max_v p(e_y|v \mapsto y, u \mapsto x) \tag{9}$$

For each neighbouring landmark y, we can now define that likelihood of observing its evidence. The likelihood of not observing any feature matches for y is equal to the probability that y does not co-occur with x in the scene. The likelihood of observing a feature that does match to y is equal to the product of the likelihood of the visual word assigned to v, and the likelihood of the spatial word assigned to w_{uv}, multiplied by the prior probability of co-occurrence of x and y. In summary:

$$\begin{aligned} &p(e_y|v \mapsto y, u \mapsto x) \\ &= \begin{cases} 1 - p(y|x) & \text{if } e_y = \emptyset \\ p(\pi_v|\Pi_y)p(\delta_{uv}|\Delta_{xy})p(y|x) & \text{otherwise} \end{cases} \end{aligned} \tag{10}$$

where $p(y|x)$ is the co-occurrence rate of x and y in images containing x, and $p(\pi_v|\Pi_y)$ and $p(\delta_{uv}|\Delta_{xy})$ are likelihoods of visual word π_v and spatial word δ_{uv} in the respective wordsets of landmark y and landmark co-occurrence z_{xy}.

4.4.2 Neighbourhood evidence for competing landmarks x'

To calculate the likelihood of the neighbourhood evidence for all competing landmarks, we consider the likelihood that these features have occurred randomly in the image and do not represent any landmarks. For each neighbouring landmark of competing landmark x', if the neighbouring landmark for x' has zero evidence ($e_{y'} = \emptyset$), then we model the likelihood of not observing the neighbouring landmark as 1. This is because, given the large size of both visual and spatial dictionaries, the probability of observing any combination of visual and spatial words is negligible, and so the probability of *not* observing them is approximately 1. However, if a feature matching the neighbouring landmark has been observed, we model the likelihood as the joint probability of randomly observing the respective visual word, $p(\pi_v)$, and spatial word, $p(\delta_{uv})$, combination. In summary:

$$p(e_{y'}|v \mapsto y, u \mapsto x) = \begin{cases} 1 & \text{if } e_{y'} = \emptyset \\ p(\pi_v)p(\delta_{uv}) & \text{otherwise} \end{cases} \tag{11}$$

4.4.3 Evidence for no true landmark matches

Finally, we now consider the evidence likelihood when u is, in fact, not a true match to any of the candidate landmarks, $p(E|u \mapsto \emptyset)$. This is the joint probability, over all features in E, of randomly observing the respective visual words and spatial words in any given image:

$$p(E|u \mapsto \emptyset) = p(\pi_u) \prod_{x \in X^C} \prod_{y \in Y_x} p(e_y) \tag{12}$$

Here, $p(e_y)$ is the probability of randomly observing the evidence for neighbouring landmark y, and is defined in a similar manner to Equation 11:

$$p(e_y) = \begin{cases} 1 & \text{if } e_y = \emptyset \\ p(\pi_v)p(\delta_{uv}) & \text{otherwise} \end{cases} \tag{13}$$

Finally, we calculate the prior likelihood that u is not a true match to any landmark, by computing the joint probability over all landmarks that each is not a true match to u:

$$p(u \mapsto \emptyset) = \prod_{s \in S} \prod_{x \in X_s} (1 - p(u \mapsto x)) \tag{14}$$

5 Simultaneous Place Recognition

In this section, we show how the geometric score in Equation 4 is calculated for each scene simultaneously, rather than computing a score for each scene independently before moving onto the next. Consider the neighbourhood evidence likelihood $p(E_{Y_x}|u \mapsto x)$ in Equation 9, where, for each candidate landmark x, a joint distribution is considered across landmark x's neighbouring landmarks $y \in Y_x$. Now, the contents of the neighbouring set could feasibly incorporate all landmarks in the scene other than landmark x. Whilst this would provide the most powerful geometric constraints and the maximum evidence from which to compute $p(u \mapsto x|E)$, it would be an unnecessary use of computational time if a confident place recognition can be achieved with a smaller set. As such, we design the recognition engine to proceed iteratively, with each iteration concluding with a score $Score_{geo}(q, s)$ computed for each scene as in Equation 4. In the first iteration, every landmark in every scene starts with only one neighbouring landmark in Y_x, and after each iteration, one further neighbouring landmark is added. The score for each scene is computed after each iteration, and the process stops when a scene has been recognised with sufficiently high confidence. In this way, scene scores are calculated simultaneously, with the iteration converging on the most likely scene and stopping much more quickly than if each scene was considered independently.

5.1 The Iterative Algorithm

At each iteration, we compute the confidence that the currently best matching scene is a true match. First, we extract peaks in the distribution of scene scores by use of non-maximal suppression. Scene s_i is retained if, and only if, its score is greater than those for scenes s_{i-1} *and* s_{i+1}. This is to reduce the effect of perceptual aliasing whereby adjacent scenes along a path appear similar. Eliminating non-maximal locations allows computation of a confidence level that the query image depicts scene s_i; without non-maximal suppression, this confidence may never converge to a sufficient level due to a wide distribution of high scores around s_i.

At each loop of the iteration, we take the highest and second-highest scores across all locations that are locally maximal, denoted s^{max_1} and s^{max_2} respectively. The level of confidence that the scene scoring s^{max_1} is the correct match, is then defined as:

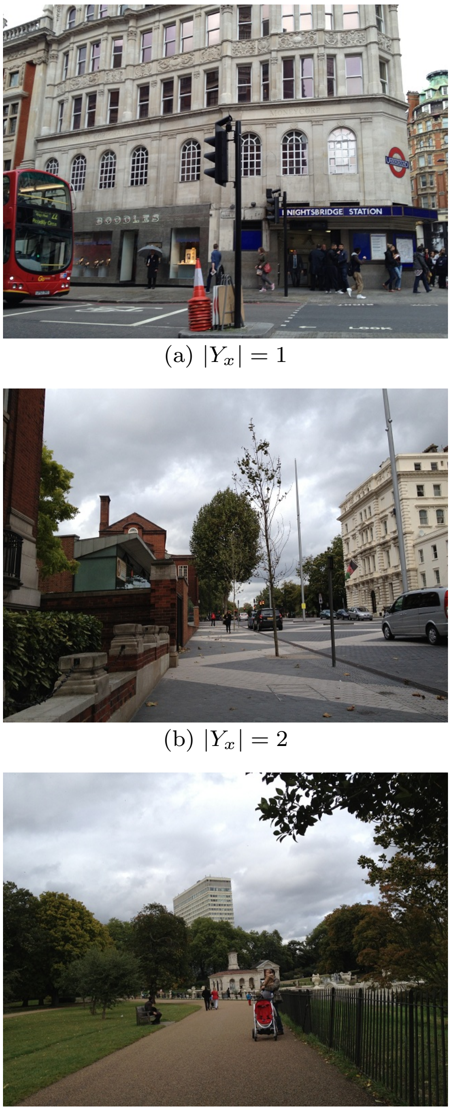
$$c = \frac{s^{max_1}}{s^{max_2}} \tag{15}$$

If this confidence is less than threshold c_{min}, then a further neighbouring landmark is added to each set Y_x for all landmarks, and the process repeats until a sufficient confidence is achieved. Determining the value of c_{min} is a compromise between efficiency and recognition accuracy. We chose a value of 25 for our experiments, which typically results in one to four neighbouring landmarks being included in each set Y_x, depending on the difficulty in recognising the query image. Increasing c_{min} any further yields little recognition improvement but increases computational time dramatically.

Figure 7 shows three query images, each requiring a different number of neighbouring landmarks before a confident scene match was achieved. The image in 7a is easy to recognise due to the abundance of stable, static features in the scene. However, the image in 7c has few stable features due to the dynamic appearance of the foliage, and so requires a more rigorous evaluation of the geometry of co-occurrences before the correct scene can be found.

5.2 Entropy-guided Neighbouring Landmark Choice

We now consider the order in which the neighbouring landmarks are added to Y_x, such that those which are most informative about the presence of x in the query image, are added first. A neighbouring landmark y could be informative for several reasons. First, it could have a high co-occurrence rate with x. Second, it could have a rigid and hence discriminative geometric relationship with x. Third, it could have a discriminative set of observed visual words. In fact, the most informative neighbouring landmarks will most likely exhibit all three properties.



(a) $|Y_x| = 1$

(b) $|Y_x| = 2$

(c) $|Y_x| = 4$

Fig. 7: The number of required neighbouring landmarks in Y_x increases as the query image becomes more difficult to recognise.

To evaluate the suitability of y for inclusion in Y_x, we consider to what extent making an observation about y reduces our uncertainty on the presence of x. First, we define $\mathbf{X}$ as a binary variable indicating whether x has been observed in a query image. Then, we define $\mathbf{E}_y$ as a binary variable indicating whether evidence has been observed for y, i.e. a query feature with the necessary visual word and spatial word combination. The reduction in uncertainty of $\mathbf{X}$ when attempting to find a feature matching y can now be represented by the conditional entropy of $\mathbf{X}$ on $\mathbf{E}_y$:

$$H(\mathbf{X}|\mathbf{E}_y) = \sum_{\mathbf{X} \in \{0,1\}} \sum_{\mathbf{E}_y \in \{0,1\}} p(\mathbf{X}, \mathbf{E}_y) \log \frac{p(\mathbf{E}_y)}{p(\mathbf{X}, \mathbf{E}_y)} \tag{16}$$

5.3 Calculating the Conditional Entropy

To calculate this conditional entropy, it is necessary to compute the probabilities of each of the four binary combinations of $\{\mathbf{X}, \mathbf{E}_y\}$ in Equation 16. This is achieved by summing the individual contributions from each scene in the database:

$$p(\mathbf{X}, \mathbf{E}_y) = \sum_{s \in S} p(\mathbf{X}, \mathbf{E}_y|s) p(s) \tag{17}$$

For any given scene, the probability of a combination of $\{\mathbf{X}, \mathbf{E}_y\}$ occurring is calculated by considering how likely it is that a feature in the scene causes $\mathbf{X}$'s given value, together with a neighbouring feature in the scene causing $\mathbf{E}_y$'s given value. By assuming that all stable features in a scene represent landmarks, we therefore marginalise over all landmarks X'_s in the scene, and the respective neighbouring landmarks $Y'_{x'}$:

$$p(\mathbf{X}, \mathbf{E}_y|s) = \sum_{x' \in X'_s} \sum_{y' \in Y'_{x'}} p(\mathbf{X}|x') p(\mathbf{E}_y|y') p(x'|s) p(y'|x') \tag{18}$$

For any given scene, $p(\mathbf{X} = 1|x')$ is only non-zero if s is the scene containing x, *and* x' is equivalent to x, in which case the value is 1. $p(\mathbf{E}_y = 1|y')$ is computed by considering the likelihood of observing the visual word and spatial word combination in y's wordsets given the combinations in y''s wordsets. $p(\mathbf{X} = 0|x')$ and $p(\mathbf{E}_y = 0|y')$ are then one minus their respective complements.

The calculation of Equation 18 is therefore carried out by simulating all pairs of features in all the training images that form landmarks, and calculating how likely it is that the pair's visual and spatial words match the co-occurrence of x and y. Similar calculations are performed to compute $p(\mathbf{E}_y)$ in Equation 16. Those neighbouring landmarks whose wordsets have more matches from the original feature tracks of y, and fewer matches from tracks of other landmarks, provide the most evidence for the presence of x, and hence result in a lower conditional entropy $H(\mathbf{X}|\mathbf{E}_y)$.

In this way, the entropy of x conditional on y is dependent on not only how likely x and y are to co-occur

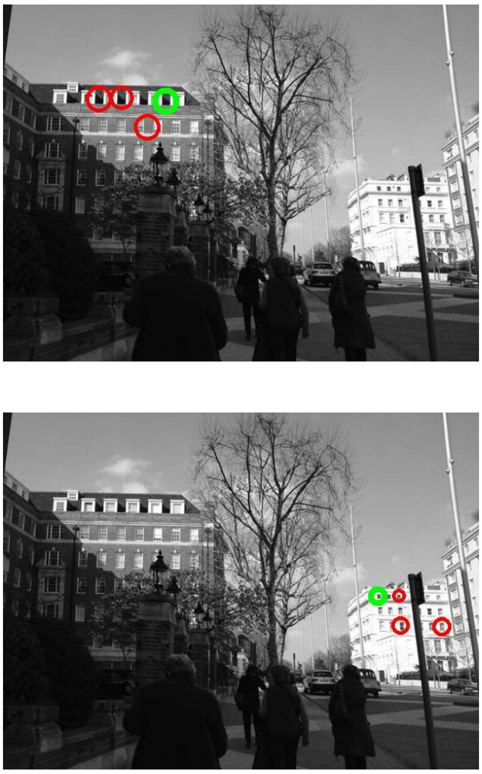


Fig. 8: Each image depicts a landmark (green) and the three neighbouring landmarks (red) with the lowest conditional entropy in Equation 16.

in the same image in a rigid manner, but also how likely it is that this same visual word and spatial word combination will appear when one of the other scenes in the map is observed. Landmarks are then added to x's neighbouring landmark set Y_x in order of lowest conditional entropy first, such that those that are most informative about the presence or absence of x are considered first. Figure 8 shows examples of landmarks and their neighbouring landmarks which resulted in the lowest values of this conditional entropy.

6 Parameter Learning

In this section, we explain how all necessary probabilistic parameters referenced in Sections 3, 4 and 5 are learned. This is done directly from the training images for a scene, with some smoothing applied to avoid unrealistic probabilities given that the training set for each scene is small. As further training images are then incorporated over time, the dynamic elements in the scene can be reflected by updating the relevant parameters.

We divide parameters into two types: static and dynamic. *Static scene parameters* are those probabilities whose true underlying values do not change over time, although their learned values may change as further training images improve the estimation. For example, the spatial relationship between two points on a building will not change from one day to the next, but the learned relationship between the associated landmarks may change as we make more observations of the geometry of the two landmarks. *Dynamic scene parameters* are those probabilities whose true underlying values may change with time. For example, the probability that a landmark is observed in an image will change if the landmark is removed from the scene due to building renovations.

6.1 Static Scene Parameters

6.1.1 Maximum Likelihood Estimation

For many of the probabilities to be learned, the Maximum Likelihood Estimation (MLE) is a suitable technique. The scene BOW vector $\boldsymbol{\pi}_s$ and covariance matrix $\boldsymbol{\Sigma}_s$ in Equation 3 are learned in this way, by averaging the individual BOW vectors of the training images and learning the variance for each visual word. Smoothing is not appropriate because a prior on the likelihood of a visual word occurrence will reduce the discriminative power of the vector, and poor estimation of one element in the BOW vector will not affect the BOW score significantly, due to the high-dimensionality of the vector.

Further parameters that can be learned with MLE are the visual word and spatial word likelihoods assigned to landmarks and co-occurrences respectively, such as $p(\pi_u|\mathit{\Pi}_x)$ in Equation 8 and $p(\delta_{uv}|\mathit{\Delta}_{xy})$ in Equation 10. Priors on these likelihoods could be smoothed by soft assignment of visual words (Philbin et al, 2008). However, as long as the dictionaries are not too fine, the quantisation of both feature space and image space offers a natural smoothing that allows for small variations in these spaces, and so further smoothing was not considered necessary.

Finally, the probability of a landmark being observed in a scene, $p(x|s)$ as in Equation 6, is estimated with MLE by dividing the number of observations of x by the number of training images for s. Given that Equation 4 sums the contributions of each landmark individually rather than computing a joint distribution, unrealistic values of $p(x|s)$ will not affect the geometric score significantly and so smoothing was not called upon.

6.1.2 Smoothing With Priors

However, when the probabilistic model is evaluated further, it becomes apparent that smoothing with priors is necessary for estimation of the landmark co-occurrence probabilities, $p(y|x)$, as in Equation 10. Due to the probabilistic nature of Equation 9, any of these parameters being equal to 0 or 1 could cause an unrealistic score of zero when the product is computed across the joint distribution. For example, suppose that landmark x and its neighbouring landmark y always co-occur in the training images. Then if, in a query image, x is observed but y is not, for example due to an occlusion, the score in Equation 9 will be zero because it will include a factor of $p(y|x) = 0$. It is therefore necessary to smooth the co-occurrence probabilities to ensure that this does not happen.

Smoothing techniques for parameter estimation range from simple linear interpolation between two distributions to multivariate Dirichlet priors (Zhai and Lafferty, 2001). Linear interpolation was used in learning visual word likelihoods for place recognition in (Cummins and Newman, 2008) by combining the explicitly-learned likelihood with a prior likelihood on the visual word (Cummins and Newman, 2009). However, this prior is a point estimate that does not consider the full range of possible probability distributions in a true Bayesian manner. We adopt a more sophisticated method to estimate the co-occurrence probabilities by considering a prior over distributions, rather than a fixed-point prior, and evaluating the likelihood of each distribution given the observations.

Let θ denote the parameter we are estimating (in this case, $p(y|x)$), and let D denote the observed data (the co-occurrence statistics of landmarks x and y). The probability distribution over all possible values of θ, conditional on these observations, is:

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)} \tag{19}$$

Now, consider that the observations D are represented by k, the number of observations of landmark x in a training image, and n, the number of observations of both landmarks x and y in the same image. For a given value of θ, the probability that n co-occurrences are observed out of a total possible k can be computed using the binomial distribution:

$$p(D|\theta) = \binom{n}{k}\theta^k(1-\theta)^{n-k} \tag{20}$$

Figure 9 illustrates example distributions for $p(D|\theta)$ over different values of n and k.

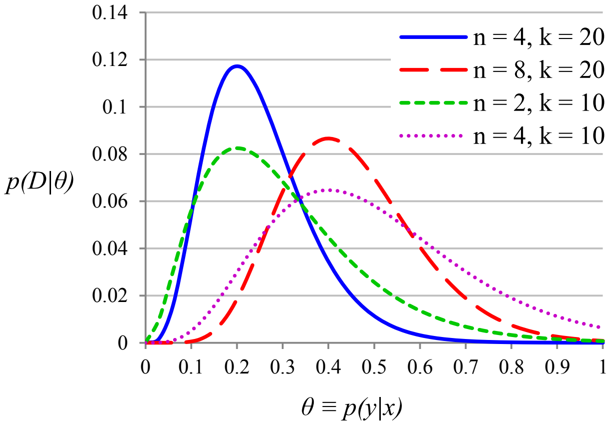


Fig. 9: The likelihood of observing co-occurrence data D given a prior on the co-occurrence probability θ, where D describes n co-occurrence observations of both x and y, out of k total observations of x.

We now consider this prior on θ, which determines the level of smoothing applied in the parameter estimation. Given that θ represents a one-dimensional distribution between 0 and 1, we choose the beta distribution as a suitable representation of the prior, which itself is parameterised by α and β:

$$p(\theta) \sim \mathrm{Beta}(\alpha, \beta) := \frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{\mathrm{B}(\alpha, \beta)} \tag{21}$$

The distribution can be learned by computing mean and variance statistics and using the method of moments (Bowman and Shenton, 2007) to determine α and β. We achieved this by computing the relevant statistics across a separate training database of manually-labelled feature correspondences, through which the presence or absence of landmark co-occurrences was recorded. Figure 10 illustrates this learned prior on θ as a continuous distribution.

Finally, we compute $p(y|x)^*$, our estimation of the co-occurrence probability, as the expected value of θ by integrating over the full range of possible values of θ:

$$p(y|x)^* = \mathrm{E}[p(\theta|D)] \tag{22}$$

Due to the smooth and simple nature of the function $p(\theta|D)$, this integration was achieved computationally by sampling the function at regular intervals between $\theta = 0$ and $\theta = 1$ and computing a summation over linear interpolating functions at each sample point. The sampling density was iteratively increased until the difference in the summation between iterations was less than 0.001. Figure 11 shows the posterior probability

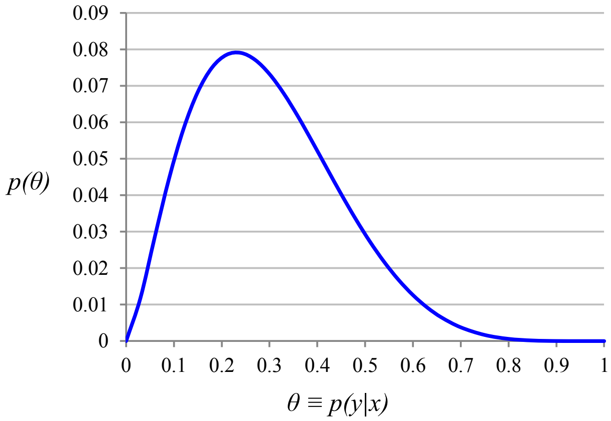


Fig. 10: The prior distribution over co-occurrence probabilities, modelled as a beta distribution, and computed by considering a separate training set and counting landmark co-occurrence rates.

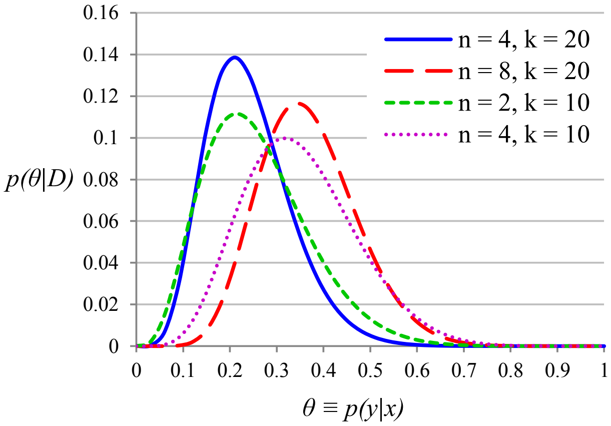


Fig. 11: The posterior co-occurrence probability θ given the observation D, where D describes n co-occurrence observations of both x and y, out of k total observations of x.

of θ given different values of evidence D. The expected value of θ is then the integration of the relevant curve.

6.2 Dynamic Scene Parameters

As time passes and the appearance of each scene changes, the scene models begin to reflect these appearances less accurately. For example, the true probability of observing a landmark drops to zero if that landmark is removed from the scene, for example due to building renovations or seasonal changes to trees. (We do not consider cyclic dynamics with features that continually appear and disappear, such as from doors opening and closing.) As a consequence, the probability of observing the co-occurrences associated with that landmark will also drop to zero. Furthermore, landmarks that were not present in the original set of training images can enter the scene at a later date.

In order to account for these dynamic effects, more recent training images need to be acquired to update the respective scene models in an incremental fashion. Given a new training image, feature correspondences with all existing training images are detected as in Section 2.4. Then, the static scene parameters for landmarks and landmark co-occurrences are updated based on the observations of landmarks in the new image. Finally, any new landmarks that have recently appeared in the scene are added to the scene model.

Given a set of T training images that have been acquired in chronological order, we now wish to compute $p(x|s)_T$, the landmark occurrence probability at time T (the time of the most recent training image). Computing this is not trivial however, because landmarks are naturally imperfectly observable, and the absence of a landmark observation could arise both from failure of the feature detector and from the elimination of the landmark from the scene.

As such, we decompose the observability of a landmark into two components: the *landmark presence* α_T, the probability that the real-world point representing the landmark is still present in the scene at time T, and the *landmark stability* β, the probability that the landmark will be observed as a feature, given that the landmark is actually present in the scene. Whilst α may drop to zero if the landmark is removed from the scene, we assume β to be constant and not dependent on when the landmark is observed. Multiplying these together yields the overall observation probability at time T:

$$p(x|s)_T = \alpha_T \times \beta \tag{23}$$

The landmark stability β is computed by dividing the number of observations of the landmark by the number of training images, for the sequence of images over which the landmark is present in the environment (but not necessarily observed). If the landmark was first observed at time t_0, and observed n times between t_0 and t_T, the landmark stability is calculated as:

$$\beta = \frac{n}{T} \tag{24}$$

The landmark presence α is now addressed. We impose a Markov blanket on α such that α_t is only dependent on α_{t-1} and the observations at $t-1$. Let us define the binary output function $f(x,t)$ indicating whether or not landmark x was detected in the training image at time t. If the landmark was detected in a training image at time t, then we assume that it is still present in the scene at time $t+1$, and so set the presence at $t+1$ to 1. However, if the landmark was not detected at time t, then there are two explanations. First, the landmark has a stability of less than 1 and does not always cause a feature to be observed. Second, the landmark has been eliminated from the environment and so will no longer be observed. Now, the probability that the landmark still remains in the environment, but has not been observed for n adjacent images, is equal to $(1-\beta)^n$. Therefore, the landmark presence can be summarised as:

$$\alpha_T = \begin{cases} 1 & \text{if } f(x, t-1) = 1 \\ (1-\beta)^n & \text{otherwise} \end{cases} \tag{25}$$

7 Experiments

In this section, we describe our experimental procedure and evaluation of the Spatio-Temporal Landmarks framework on our new long-term dataset, together with comparisons to state-of-the-art place recognition using other image retrieval methods.

7.1 Dataset

Our new dataset consists of GPS-tagged images captured from a standard camera whilst walking along a 7km outdoor path. Each tour of the path contains around 2300 images, each roughly 3 metres apart, and 6 tours were completed over a period of 5 months (March to August). Half the path traverses a park containing trees, foliage and grassland that undergoes significant appearance variations over the seasons, and the other half follows a road through a busy urban area undergoing structural changes due to building renovations and roadworks. As such, the dataset contains significant long-term dynamic elements, together with short-term dynamics from moving bodies, illumination variations, and lateral deviations along the path. Figures 1 and 2 highlight the challenges within our dataset over the two halves of the path.

7.2 Dictionaries

Each visual word represents a discretised portion of 128-dimension feature space (SIFT features from Lowe (2004)). The visual dictionary was generated with k-means clustering (Sivic and Zisserman, 2003) using a separate set of training images to those used for training the scene models. Building and searching the visual dictionary was achieved by approximate nearest-neighbours using random forests (Philbin et al, 2007).

The size of the visual dictionary is a compromise between precision and recall of landmarks. Given a dictionary with a very fine structure, the same landmark may be assigned to different visual words when observed under slightly different illumination or viewpoint conditions. However, if the dictionary has a coarse division, then many landmarks will all be assigned to the same visual word, which can cause confusion. Choosing an appropriate dictionary size is therefore a compromise between precision and recall, and requires consideration of the likely variation of landmark appearance over the expected range of conditions. In object classification tasks, the large intra-class variation requires small dictionaries of hundreds or thousands of visual words (Winn et al, 2005) to enable a feature representing the same part of an object to be assigned to the same visual word. For recognition of near-duplicate planar images that vary little in appearance, such as book covers or logos, larger dictionaries with millions of visual words are often used (Nister and Stewenius, 2006) because features representing the same part of the object are almost identical. We use a visual dictionary of medium size 100K in our experiments to allow for small shifts in image appearance from viewpoint and illumination effects, whilst still maintaining discriminative visual words.

Similarly, the spatial dictionary is again a compromise between precision and recall. By observing from our dataset that most inter-feature relationships do not vary by much more than 10 pixels in location, a factor of 1.1 in scale ratio, and 30 degrees in orientation, we used a spatial dictionary of 64 divisions in x- image space, 48 divisions in y- image space, 100 divisions in scale ratio, and 10 divisions in orientation, to give a dictionary of $64 \times 48 \times 100 \times 10 = 3$ million spatial words in our experiments, for 640×480 pixel images.

7.3 Experimental Procedure

For each tour of our dataset, we used three adjacent images to describe one location. This was in order to seed our system with feature tracks and sufficient statistics to compute the necessary parameters in the probabilistic model. Each tour was therefore divided into 770 locations. The initial set of locations was defined by the images in the first tour (March). For testing the recognition performance at the i^{th} month, all of the images in tour i were used as queries, and tours 0 to $i-1$

were used as training images. A recognition was considered correct if the returned dataset image location was within a distance of 5 scenes ($\sim$ 15 metres) of the query image location.

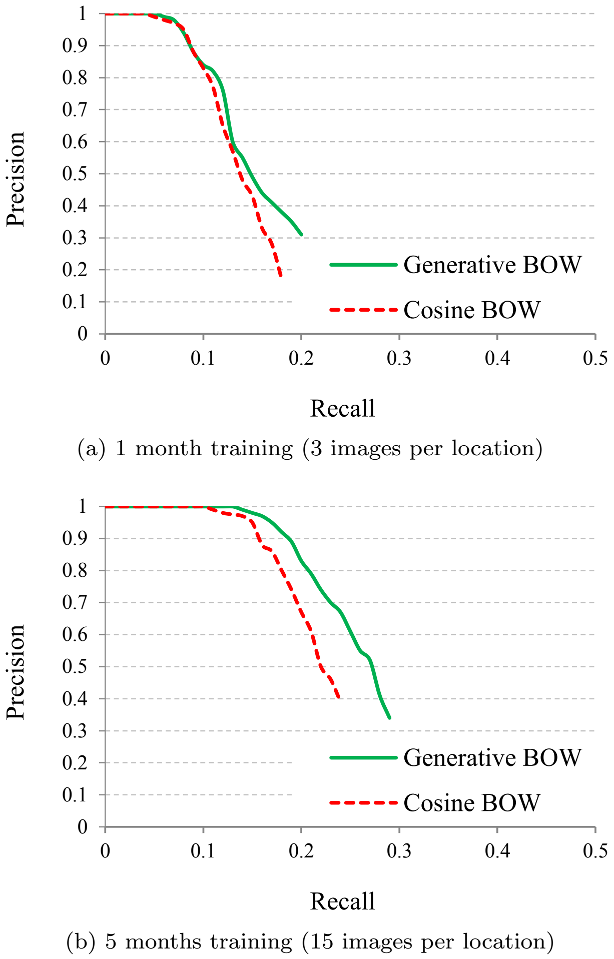
We evaluated our method against state-of-the art approaches to image retrieval. For the BOW filtering stage, we compare against the cosine similarity of *tf-idf*-weighted BOW vectors (Sivic and Zisserman, 2003) using soft quantisation (Philbin et al, 2008). For the geometric verification stage, we compare against the Hough-pyramid matching scheme (Tolias and Avrithis, 2011). Here, the Hough-pyramid proposes likely feature correspondences based on consistent geometric shifts across the two images, and these are then narrowed down with epipolar constraints following a RANSAC-based estimation of the fundamental matrix. We denote this comparison method as the baseline method. For the competing methods, when more than one training tour was available, all training images for each location formed candidates to which the query image was compared to. This is the image retrieval equivalent of incremental learning; in our system however, individual landmark properties are updated, whereas in the competing method, the entire image has to be updated. Whilst non-maximal suppression is used in our method to speed up the simultaneous place recognition algorithm, the baseline method does not employ this component because it adds no benefit when places are considered independently.

For all the methods, we use a value of $k = 50$ to define the number of images or scenes returned from the BOW similarity measure that are passed on for geometric verification. Precision-recall curves were generated by finding the location with the top score, and varying the threshold on this score as to whether it is considered a match. Experiments on both ours and the baseline geometric verification stage were carried out on the same 50 locations returned from the generative BOW filtering stage.

7.4 Quantitative Results

7.4.1 Precision-Recall

Figure 12 compares the precision-recall performance of our generative BOW method with the cosine similarity measure. In 12a, only the first tour was used to train each location, and the sixth tour was used for query images. In 12b, all of the first five tours were used to train each location. The precision-recall performance is similar when only one tour was used for training, but the performance of our system is significantly greater when several training tours are incorporated. Whilst the cosine similarity measure considers point estimates of the BOW vector for each image, our generative model is able to capitalise on the variance in visual words and determine a match in a more probabilistic manner, so long as sufficient training data is available.

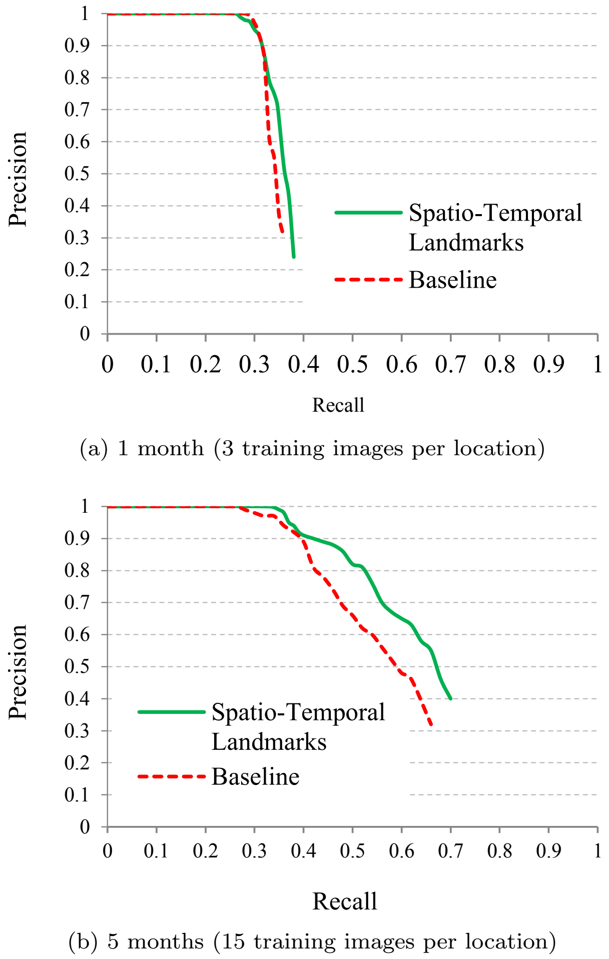


(a) 1 month training (3 images per location)

(b) 5 months training (15 images per location)

Fig. 12: Precision-recall curves for the BOW filtering stage.

Figure 13 then shows the precision-recall performance of our probabilistic voting scheme compared to the baseline geometric verification method, again over two different training periods. As with the BOW filtering stage, performance is similar when training data is limited, but dramatically improves as training data is introduced, for two reasons. First, the probabilistic model is enriched with data from which to learn its parameters more accurately. Second, the adaptation to dynamic scenes allows for landmarks that have disappeared from, or appeared in, the scene, to be accounted for. Whilst the baseline method also updates its database with the newer images, these images themselves introduce new unstable features too, whereas our



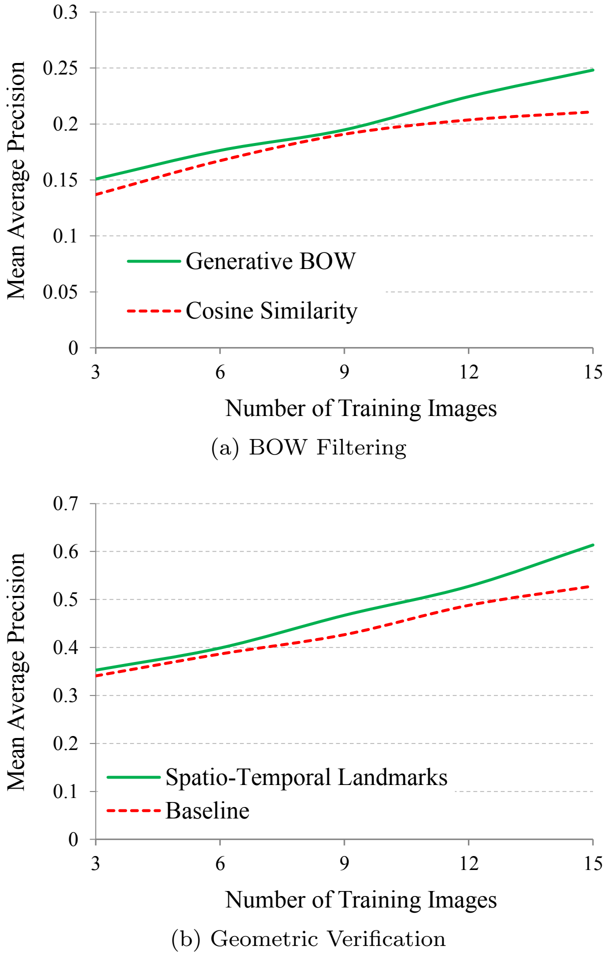
(a) 1 month (3 training images per location)

(b) 5 months (15 training images per location)

Fig. 13: Precision-recall curves for the geometric verification stage.

model only updates itself with those landmarks which we know are stable and have been tracked across more than one training image. We achieve around 35% recall at 100% precision which is very promising considering the challenges in the dataset, and would be suitable for an appearance-based SLAM system.

It should be noted that neither method is able to achieve 100% recall. This is because the database locations are ranked and the best score is returned as the recognised place, and only a fraction of queries have the correct place assigned to the best score. Adjusting the minimum score (as was done to create the precision-recall curves) defines how much confidence we need in our belief that the scene with the highest score is a correct match.



(a) BOW Filtering

(b) Geometric Verification

Fig. 14: Mean Average Precision for both stages of the two methods, as a function of the number of training images.

7.4.2 Training Data Size and Mean Average Precision

Figure 14 demonstrates the Mean Average Precision (MAP) for the BOW filtering and geometric verification stages as a function of the number of training images available. Whilst similar performance is observed with only a small number of training images, our system generally improves more rapidly than the baseline. The accumulation of newer images does provide the image retrieval engine with an updated appearance of the scene, however, there is no overall model of the scene that draws these images together, and hence recognition will only be as effective as the best image in the training set. With our Spatio-Temporal Landmarks method, conversely, each image will always improve the model because parameters are refined with each further image acquired, hence the greater rate of increase in MAP performance with further training data.

7.4.3 Computational Speed

Table 1 summarises the MAP performance and recognition time (excluding feature extraction) for both our method and the baseline method. Results are shown for each method's BOW filtering stage, and the geometric verification stage. We also show results for training on only the first month (3 images per location) and for the first five months (15 images per location).

The speed of the BOW filtering stage in our method is significantly faster than with the standard cosine similarity measure. Only those visual words that represent landmarks in the scene are considered in the BOW score, rather than comparing counts for all visual words in the vector. As further training images are acquired, the recognition time increases for our overall method due to both the incorporation of further landmarks into the scene, and the updating of parameters to include larger sets of visual and spatial words for each landmark and landmark co-occurrence. However, this increase is far less dramatic than that for the baseline method, which increases roughly linearly with the number of training images, as each must be considered independently. Even when compared to the baseline method with only 1 month of training, our method with 5 months of training still has a faster recognition, with a significantly greater MAP.

Method	Training	MAP	Recognition Time (ms)
Gen BOW	1 month	0.151	83
STL Geo	1 month	0.353	514
Gen BOW	5 months	0.248	134
STL Geo	5 months	0.614	765
Cosine BOW	1 month	0.137	276
Baseline Geo	1 month	0.341	827
Cosine BOW	5 months	0.211	1365
Baseline Geo	5 months	0.528	1879

Table 1: Summary of MAP performance and recognition time (excluding feature extraction) for our method (Gen BOW and STL (Spatio-Temporal Landmarks) Geo), and the competing image retrieval method (Cosine BOW and Baseline Geo). Results were generated on a 2.67 GHz Intel Xeon CPU.

7.5 Qualitative Results

In this section, we show some qualitative results illustrating examples when our method outperforms the image retrieval method, but also when our method fails.

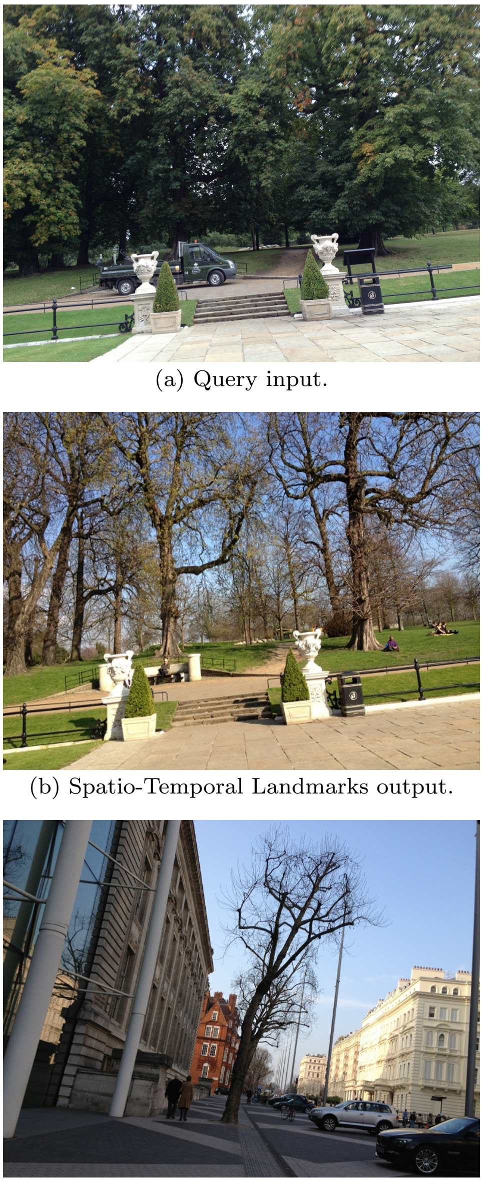
Figure 15 shows an example of where our Spatio-Temporal Landmarks system correctly recognises an image that was captured 5 months previously, but the competing image retrieval system fails. The query image exhibits significant dynamic behaviour from the trees which change in appearance over this time period. Our method is able to filter out the dynamic features, and focus on the static scene elements, such as the statues. In the baseline image retrieval system, however, it is not known which features are static or dynamic, and hence matches are attempted with dynamic features from the trees, which no longer appear in the scene 5 months later.

Whilst filtering out dynamic scene elements is one of the strengths of our method, the use of local features in describing images is largely ineffective when the scene is entirely dynamic, and there are no stable features detected between different sets of training images. Figure 16 shows an example of such a case, where the scene is covered almost entirely by foliage, which itself changes dramatically over the course of the training period. The only features that exist in the images are representative of foliage, and these are largely filtered out, leaving the scene with very few landmarks to which feature matches can be made. This problem also exists in the image retrieval case, and solutions must involve divergence away from local features towards holistic features or semantic labelling of the scene.

8 Conclusions

In this paper, we have presented a new framework for place recognition and incremental learning of dynamic changes to scenes. Modelling a place as a set of real-world landmarks enables a more robust understanding of the expected distribution of local features in an image, both in terms of descriptors and spatial relationships. A generative BOW filtering stage was introduced which learns the expected variance in visual word counts, enabling better filtering when compared to the standard cosine similarity measure. By learning which landmarks co-occur most frequently, each landmark can then be efficiently yet discriminatively verified by using the geometric relationships with only a small set of co-occurring landmarks. Furthermore, dynamic elements in a scene can be incorporated incrementally by introducing new landmarks into a scene and filtering out old landmarks. Results have shown improvements in the long-term recognition performance and efficiency over standard image-retrieval techniques.

The localisation system proposed in this paper is one that can be incorporated within a broader robot navigation framework that requires loop closure or global localisation as part of a SLAM framework. An appropriate application would be a system such as

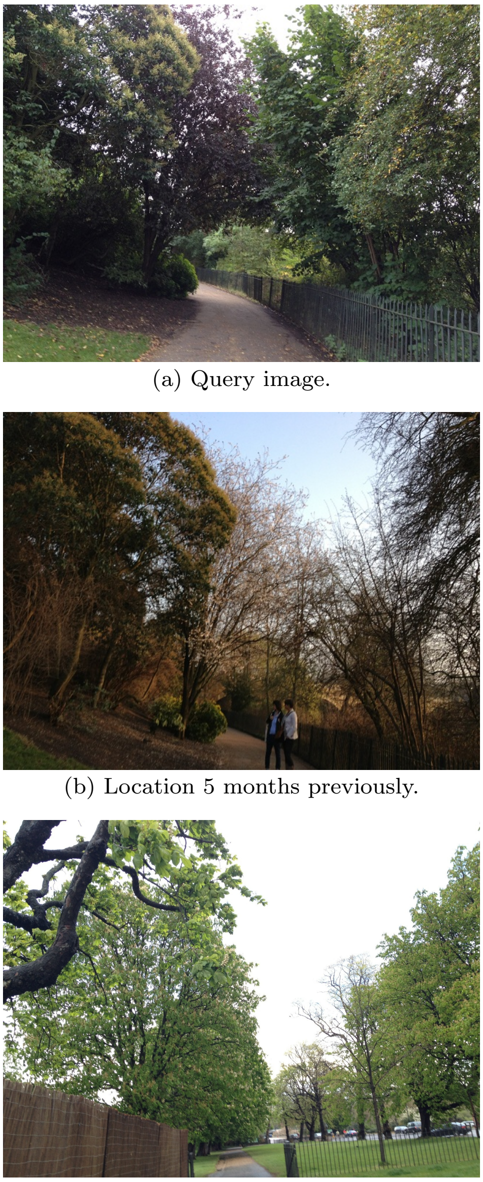


(a) Query input.

(b) Spatio-Temporal Landmarks output.

(c) Image Retrieval output.

Fig. 15: Example case when our method correctly recognises a query image, but the baseline image retrieval method fails. Our method is able to filter out the long-term dynamic features on the trees, but these can confuse the baseline method.



(a) Query image.

(b) Location 5 months previously.

(c) Returned location.

Fig. 16: Example case when our method fails due to the entire scene being dynamic, with few stable local features detected across the training images. The baseline image retrieval method also fails with this query.

an autonomous vehicles network (Cummins and Newman, 2009; Johns and Yang, 2013b), where GPS-tagged training images sequences from one vehicle can be used to train the scene models, for use by all other vehicles, which do not need the GPS data themselves. Suitable further work would be to address the issue of map building, self-intersecting maps, more precise quantitative localisation, and issues with memory efficiency for long-term applications. As it stands, it is necessary to train the scene models with GPS-tagged images, but a SLAM extension is fitting given the probabilistic nature of the scene similarity score and the ability to update individual landmark properties incrementally. However, for applications where it is easy to collect GPS-tagged images over time, such as on roads for autonomous car navigation, this system is readily applicable in its current form.

References

Agarwal S, Snavely N, Simon I, Seitz SM, Szeliski R (2009) Building rome in a day. In: Proc. ICCV 1

Arandjelovic R, Zisserman A (2012) Three things everyone should know to improve object retrieval. In: Proc. CVPR 1, 5

Bowman KO, Shenton LR (2007) The beta distribution, moment method. Far East Journal of Theoretical Statistics 23 12

Cao Y, Wang C, Li Z, Zhang L (2010) Spatial bag-of-features. In: Proc. CVPR 2

Chatfield K, Lempitsky V, Vedaldi A, Zisserman A (2011) The devil is in the details: An evaluation of recent feature encoding methods. In: Proc. BMVC 2

Chum O, Philbin J, Zisserman A (2008) Near duplicate image detection: min-hash and tf-idf weighting. In: Proc. BMVC 2

Chum O, Mikulík A, Perdoch M, Matas J (2011) Total recall ii: Query expansion revisited. In: Proc. CVPR, pp 889–896 3

Csurka G, Dance CR, Fan L, Willamowski J, Bray C (2004) Visual categorization with bags of keypoints. In: Proc. Workshop on Statistical Learning in Computer Vision, ECCV 5

Cummins M, Newman P (2008) Fab-map: Probabilistic localization and mapping in the space of appearance. IJRR 27:647–665 12

Cummins M, Newman P (2009) Highly scalable appearance-only slam - fab-map 2.0. In: Robotics: Science and Systems 1, 12, 18

E Arnaud FO E Delponte, Verri A (2006) Trains of keypoints for 3d object recognition. In: Proc. ICPR 3

F Orabona LJ, Caputo B (2010) Online-batch strongly convex multi kernel learning. In: Proc. CVPR 3

Hartley RI, Zisserman A (2004) Multiple View Geometry in Computer Vision. Cambridge University Press 3, 5

Jegou H, Chum O (2012) Negative evidences and co-occurrences in image retrieval: the benefit of pca and whitening. In: Proc. ECCV 2

Jégou H, Douze M, Schmid C (2010) Improving bag-of-features for large scale image search. IJCV 87(3):316–336 2, 3

Johns E, Yang GZ (2011a) From images to scenes: Compressing an image cluster into a single scene model for place recognition. In: Proc. ICCV, pp 874–881 1, 3

Johns E, Yang GZ (2011b) Global localization in a dense continuous topological map 4

Johns E, Yang GZ (2011c) Place recognition and online learning in dynamic scenes with spatio-temporal landmarks. In: Proc. BMVC, pp 10.1 – 10.12 2, 4, 5

Johns E, Yang GZ (2013a) Dynamic scene models for incremental, long-term, appearance-based localisation. In: Proc. ICRA 1

Johns E, Yang GZ (2013b) Feature co-occurrence maps: Appearance-based localisation throughout the day. In: Proc. ICRA 18

Leordeanu M, Hebert M (2005) A spectral technique for correspondence problems using pairwise constraints. In: Proc. ICCV, pp 1482 – 1489 5

Li Y, Snavely N, Huttenlocher DP (2010) Location recognition using prioritized feature matching. In: Proc. ECCV, pp 791 – 804 3, 5

Lik F, Kosecka J (2006) Probabilistic location recognition using reduced feature set. In: Proc. ICRA 3

Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Trans IJCV 60:91 – 110 3, 14

Luo J, Pronobis A, Caputo B, Jensfelt P (2007) Incremental learning for place recognition in dynamic environments. In: Proc. IROS 3

Marszalek M, Schmid C (2006) Spatial weighting for bag-of-features. In: Proc. CVPR 2

Mikulk A, Perdoch M (2010) Learning a fine vocabulary. In: Proc. ECCV 2

Ni K, Kannan A, Criminis A, Winn J (2009) Epitomic location recognition. Trans PAMI 3

Nister D, Stewenius H (2006) Scalable recognition with a vocabulary tree. In: Proc. CVPR, pp 1222–1229 1, 14

Philbin J, Chum O, Isard M, Sivic J, Zisserman A (2007) Object retrieval with large vocabularies and fast spatial matching. In: Proc. CVPR, pp 1 – 8 3, 14

Philbin J, Chum O, Isard M, Sivic J, Zisserman A (2008) Lost in quantization: Improving particular object retrieval in large scale image databases. In: Proc. CVPR 2, 5, 11, 15

Pronobis A, Caputo B (2007) Confidence-based cue integration for visual place recognition. In: Proc. IROS 3

Raguram R, Wu C, Frahm JM, Lazebnik S (2011) Modeling and recognition of landmark image collections using iconic scene graphs. Trans IJCV 95(3):213–239 3

Schindler G, Brown M, Szeliski R (2007) City-scale location recognition. In: Proc. CVPR 1

Se S, Lowe D, Little J (2001) Vision-based mobile robot localization and mapping using scale-invariant features. In: Proc. ICRA 3

Sivic J, Zisserman A (2003) Video google: a text retrieval approach to object matching in videos. In: Proc. ICCV, pp 1470–1477 1, 2, 3, 5, 6, 14, 15

Tolias G, Avrithis Y (2011) Speeded-up, relaxed spatial matching. In: Proc. ICCV 3, 5, 15

Winn J, Criminisi A, Minka T (2005) Object categorization by learned universal visual dictionary. In: Proc. ICCV, pp 1800 – 1807 14

Zhai C, Lafferty J (2001) A study of smoothing methods for language models applied to ad hoc information retrieval. In: Proc. ACM SIGIR, pp 334 – 342 12

Zhang Y, Jia Z, Chen T (2011) Image retrieval with geometry-preserving visual phrases. In: Proc. CVPR, pp 809 – 816 3

Zheng YT, Zhao M, Song Y, Adam H, Buddemeier U, Bissacco A, Brucher F, Chua TS, Neven H (2009) Tour the world: building a web-scale landmark recognition engine. In: Proc. CVPR 1