Building Location Models for Visual Place Recognition

Elena Stumm, Christopher Mei, Simon Lacroix

Building Location Models for Visual Place Recognition

Elena Stumm[1,2], Christopher Mei[1], and Simon Lacroix[1]

[1]Robotics and Interactions Group, LAAS-CNRS, Toulouse, France
[2]Université de Toulouse III - Paul Sabatier, Toulouse, France

September 15, 2014

Abstract

This paper deals with the task of appearance-based mapping and place recognition. Previously, the scope of a location generally varied between either using discrete poses or loosely defined sequences of poses, facing problems related to perceptual aliasing and path invariance respectively. Here, we present a unified framework for defining, modelling and recognizing places in a way which is directly related to the underlying structure of features in the environment. A covisibility map of the environment is incrementally maintained over time, where visual landmarks represent nodes in a graph which are connected if seen together. When queried, relevant places are retrieved as clusters from this map, and a novel probabilistic observation model is used to evaluate place recognition. Place retrieval is able to adapt to a given query and also inherently cope with trajectory variations, due to the use of the landmark covisibility structure. In addition, the chosen generative model is developed in a way which is robust to observation errors, mapping errors, perceptual aliasing, and parameter sensitivity. Validation is provided through a variety of tests using real-world datasets, which compare the behaviour of the proposed approach to other representative state-of-the-art methods (namely FAB-MAP and SeqSLAM).

1 Introduction

Long-term autonomous navigation is becoming increasingly important for a variety of mobile robotic platforms and applications. For this purpose, the navigation platform must be robust to errors, with localization working even in the case of unexpected, dynamic, and possibly self-similar environments. One important tool for reliably maintaining error bounds on a robot's position is visual place recognition for performing loop-closure, due to its accessibility and ability to work in a wide range of environments [Cummins and Newman, 2011, Maddern et al., 2012]. In addition to loop-closure, place recognition can be an important function, as a stepping stone towards semantic mapping and scene understanding, as well as map fusion. Incorrect place recognition can cause large mapping errors in a SLAM or map fusion context, or totally erroneous reasoning based on incorrect scene interpretation, emphasizing the importance of avoiding false-positive associations.

The goal being addressed in this paper is that of continuously evaluating whether a mobile robot is revisiting a place in which it has already been, or is in a previously unknown location, by analysing the visual appearance of the current scene. Ultimately, this should be achieved with minimal assumptions about the behaviour of the environment and robot, in order to remain applicable in the general case. Additionally, this work relies exclusively on visual information captured through a camera. Some of the main challenges related to this task include making reliable data-association decisions, differentiating between repetitive and self-similar scenes (referred to as perceptual aliasing), as well as appearance changes due to dynamic elements, variations in view-point, and lighting changes.

This paper presents a Bayesian framework which works with feature-based location models built from sets of covisible scene elements for evaluating place recognition. Location models which are built up by grouping features based on visual connectivity allow for more context than single images, while maintaining invariance to motion changes. Additionally, working with feature-based models provides a built-in robustness towards view-point and lighting changes through the use of locally invariant feature descriptors. Incorporating such location models into a Bayesian framework can allow for an integrated and probabilistically sound way to handle perceptual aliasing, dynamic elements, and uncertainty about detections.

1.1 Related Work

Appearance-based loop-closure and mapping has been brought to the forefront since the introduction of FAB-MAP (F*ast* A*ppearance*-B*ased* MAP*ping*) [Cummins and Newman, 2011, 2008], an implementation based on visual bag-of-words techniques derived by Sivic and Zisserman [2003]. In the FAB-MAP framework, an image is represented by a set (or bag) of quantized local image descriptors (or words) belonging to a predefined dictionary. This representation is easy to work with and fairly robust in the presence of lighting changes, view-point changes, and dynamic environments. Relevant images can be retrieved quickly using an inverted index system [Sivic and Zisserman, 2003], and compared to a query using a variety of approaches borrowed from the field of text retrieval.

One common scoring method that has been used for visual place recognition is known as TF-IDF (T*erm* F*requency* - I*nverse* D*ocument* F*requency*, [Manning et al., 2008]), which creates vectors for each location where each element is the ratio between how common a word is within that location and how common the word is within the entire set of locations [Sivic and Zisserman, 2003]. Locations can then be compared by finding the distance between their corresponding TF-IDF vectors. TF-IDF scoring has been successfully used in work such as Angeli et al. [2008], Mei et al. [2010], and Botterill et al. [2011]. This technique is easy to implement and computationally efficient. However, it requires threshold tuning, and (as in text retrieval), has been shown to be sensitive to location (document) characteristics such as repetitive structures [Schneider, 2004, Jegou et al., 2009] and size [Mei et al., 2010]. Alternative weighting schemes also exist, which try to compensate bursts in word frequencies [Jegou et al., 2009, Torii et al., 2013].

Other scoring methods rely on probabilistic models, such as the work of Cummins and Newman [2008], and Cadena et al. [2012]. The formulation of a generative model for place recognition in FAB-MAP [Cummins and Newman, 2008] is fundamental work on the topic. This allows for a natural and principled way to incorporate dynamic environments and perceptual aliasing, and only requires two parameters representing feature detection probabilities. Additionally, decision thresholds for matching locations become clear probabilities. Discriminative models used in [Cadena et al., 2012] have also been shown to produce good results without knowledge of hidden variables. However discriminative models risk overfitting and lack the ability to generalize well, due to a heavy reliance on representative training data.

On the other end of the spectrum, successful results have also been obtained by using global, rather than local features. The work of SeqSLAM [Milford, 2013] compares sequences of down-sampled images to detect loop closures. Working without feature-based techniques has the advantage of functioning even with poor quality images (e.g., low resolution, low image depth, and/or blur). In addition, this method has been proven to work even in the case of extreme lighting and weather changes. However, the algorithm relies on the assumptions of trajectory invariance, and the authors have demonstrated sensitivity to sequence length.

Inclusion of such place recognition algorithms has enabled significant improvements in long-term localization and mapping, employed in datasets up to distances of 1000 km, containing drastic lighting changes and many self-similar locations which cause perceptual aliasing [Cummins and Newman, 2011, Milford, 2013]. Impressive as these systems are, there is still room for improvement in terms of how maps and locations are modelled. For instance, location

models built using specific poses in the robot's trajectory imply that the robot must visit the same arbitrary pose in order to recognize any relevant loop-closures. Abstraction from single image location models has been addressed in the works of Mei et al. [2010], Maddern et al. [2012], Gálvez-López and Tardós [2012], and Milford [2013]. CAT-SLAM [Maddern et al., 2012] moves towards a continuous representation, but requires local metric information. In both the works of Gálvez-López and Tardós [2012] and Milford [2013], comparisons are made with sequences based on time, under the assumption that the speed and frame-rate remain approximately constant. Instead of grouping locations based on timed sequences, the work of Mei et al. [2010] dynamically queries location models as cliques from a covisibility graph of landmarks which are connected if seen together. These location models are then based on the underlying environmental features, rather than the discretization of the robot's trajectory in the form of individual images, or sequences of images in time. This methodology is therefore further explored in this work.

In addition, when it comes to comparing locations, methods based on TF-IDF such as [Angeli et al., 2008] and [Mei et al., 2010] require careful threshold tuning, while probabilistic methods such as [Cummins and Newman, 2008] implicitly require consistency in the map as probabilities are normalized across known locations. Alternatively, a normalization technique which remains independent of the current map by using sample images to model the unknown world was proposed in [Stumm et al., 2013]. The effects of the assumptions underlying these various models is investigated in this work, and the final probabilistic model is updated for further robustness to shortcomings of previous methods.

1.2 Overview

This paper focuses on visual place recognition for mobile robots, building on the works of [Stumm et al., 2013, Mei et al., 2010, Cummins and Newman, 2008] by establishing generative location models using covisibility maps. This section provides a brief overview of the structure of the paper and the place recognition framework which is presented.

Ideas inspired from the text-document retrieval field (see [Manning et al., 2008] for an overview) are used to find previously seen locations which match a query observation. Locations are represented by a bag-of-words model, with words provided by quantized visual features and places used analogously to documents, as is common in recent literature [Cummins and Newman, 2008, Sivic and Zisserman, 2003, Angeli et al., 2008, Botterill et al., 2011]. However, unlike typical pose-based implementations which rely on single-image location models, here relevant *"virtual locations"* are retrieved as subgraphs from a covisibility graph at query time, therefore dubbed a *dynamic bag-of-words* approach [Mei et al., 2010]. This covisibility graph is constructed as the robot explores the environment, by noting which landmarks are observed

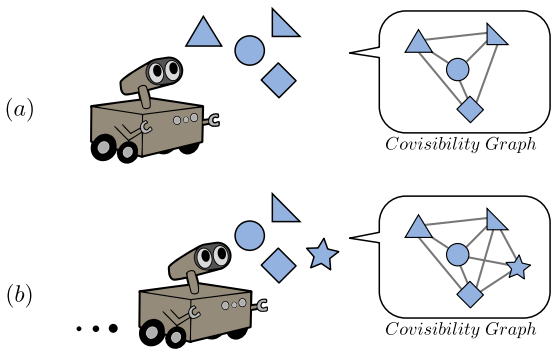


Figure 1: As the robot moves, it makes observations, detects landmarks, and notes which ones were seen together in a graph structure. Here you can see a simple example of two steps as a robot moves forward through the environment, and the resulting covisibility graph.

together in a graph structure. The basic mapping concept is depicted in Figure 1. By working with this covisibility graph, a more truthful, continuous representation of the environment is used, rather than a discrete selection of arbitrary poses from the robot's trajectory, which generally lack both context and consistency. Places are now defined using direct properties of the environment (landmarks), and become less dependent on variations in trajectory while eliminating the problem of pose selection.

Once the virtual locations are retrieved, a probabilistic framework is used to identify any potential matches between the query and previously seen locations. Development of a proper generative model is a key factor for providing useful results, especially in challenging environments, and is therefore the main focus of this paper. For instance, a rigorous probabilistic method allows for inherent confidence thresholds, and can handle problematic situations such as perceptual aliasing by understanding the likelihood of scene elements. This relies on a careful treatment of probability normalization, done here using a set of sample locations which are used to model the unknown world. The method presented in this paper allows the system to search for all matching locations, rather than the one most probable, giving the potential to cope with erroneous maps which may contain more than one instance of the same location (for example where loop-closures were missed). The developed model also improves the stability, in comparison to previous work. The resulting posterior probabilities represent an intuitive measure of place similarity, with values varying smoothly as a queried location is traversed.

Covisibility graphs and location retrieval is discussed next in Section 2, followed by the development of location models in Section 3. Then, Section 4 provides details about the implementation and a discussion about the behaviour of the system, and Section 5 summarizes the paper with concluding remarks.

1.3 Preface to the Evaluation

Since the theoretical developments of this paper are largely backed by experimental evaluation, many test results will be presented throughout the paper for clarity. This requires a brief introduction to test metrics and image datasets. The framework is evaluated on a total of six different sequences of images which are labeled with ground-truth position estimates. The six sequences which are used are referred to throughout the paper as the Begbroke dataset, Begbroke Multispeed dataset, City Centre dataset, New College dataset, the KITTI dataset, and the Ruelle dataset. Details and example images from each dataset can be found in Appendix A. To summarize, the Begbroke sequences consist of images from a forward-facing camera, with three loops around a path which is mostly surrounded by trees and fields, totalling approximately 1 km. The City Centre and New College datasets are taken from the work of Cummins and Newman [2008], and consist of images from two cameras angled slightly left and right, as the robot traverses a university campus with buildings, roads, gardens, cars and people. Each of these datasets make up approximately 2 km in total. The KITTI dataset is provided by [Geiger et al., 2013], and consists of roughly 1.6 km of forward-facing images from a car travelling through the city of Karlsruhe. Lastly, the Ruelle dataset is a much shorter sequence of roughly 200 m of forward-facing images from a hand-held camera. This dataset only covers one street, but traverses it multiple times from different view points and with varying speeds. The overall performance of the place-recognition algorithms are primarily evaluated by analysing the precision-recall characteristics. Precision is defined as the number of true-positive detections divided by the sum of all returned detections (true-positives plus false-positives), while recall is defined as the number of true-positive detections divided by the sum of all detections which should be found (true-positives plus false-negatives). Therefore, one important test metric is the maximum recall which can be obtained while maintaining full precision (no false-positives). Ground truth is provided by GPS tags on the images, where each image taken within a given radius of the query location is considered to be a true match. More details regarding the precision-recall metrics can be found in Section 4 and Appendix B.

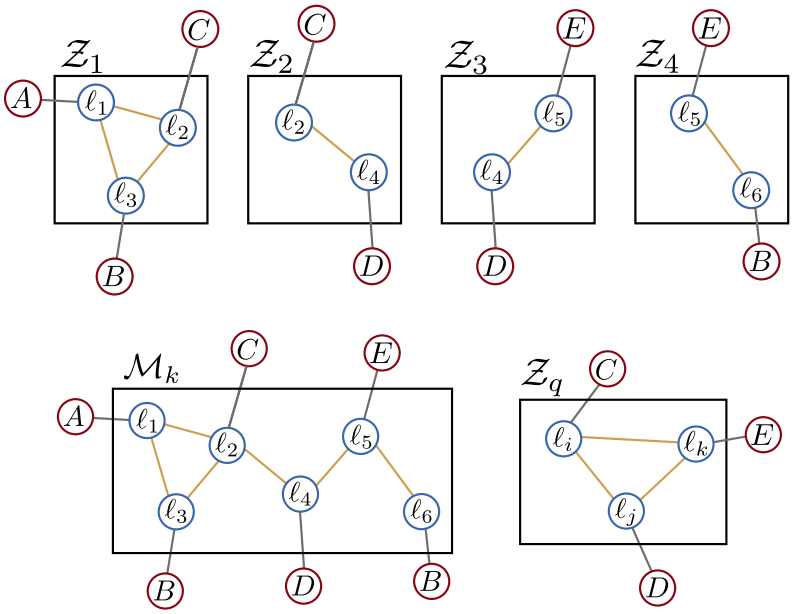


Figure 2: A sequence of simplified example observations is shown in the top row ($\mathcal{Z}_1$, $\mathcal{Z}_2$, $\mathcal{Z}_3$, $\mathcal{Z}_4$), along with the corresponding covisibility map, $\mathcal{M}_k$, on the bottom-left, and the current query observation, $\mathcal{Z}_q$, on the bottom-right. The figure also depicts which word (represented as A, B, C, D or E) is associated with each landmark, ℓ_i.

2 Covisibility Graphs

Given a query location (usually corresponding to the current position of the robot), the system needs to be able to evaluate if and where the same location was seen before. The primary approach used in this paper is a probabilistic bag-of-words method using sets of visual words [Sivic and Zisserman, 2003], which is able to compare the query to a set of candidate locations retrieved from the current map. This section outlines how the environment is represented as a graph of visual features, where covisibility defines connectivity [Mei et al., 2010]. Probabilistic location models are built using sets of observed visual words, as will be described in detail in Section 3. The visual words represent quantized feature descriptors provided by a set of landmarks (distinct visual features in the image). These sets of landmarks are given by subgraphs of the robot's covisibility map, $\mathcal{M}_k$, an undirected graph, with nodes representing landmarks, and edges representing the information that the connected landmarks were seen together. At query time, the graph is searched for clusters of landmarks which share strong similarity with the query. These clusters are then expanded based on covisibility, to extract subgraphs which represent candidate virtual locations for further comparative evaluation. Section 2.1 explains how the covisibility graph is built and maintained over time, and Sections 2.2 and 2.3 discuss how to retrieve the relevant virtual locations.

2.1 Building the Covisibility Graph

As each image is processed, a set of visual features, ℓ_i, are detected and represented by a vector-based descriptor such as SIFT [Lowe, 1999] or SURF [Bay et al., 2008]. Each landmark is furthermore associated with a quantized visual word, which is taken by the closest match in a pre-trained visual dictionary, $\mathcal{V}$ [Sivic and Zisserman, 2003]. Thus, each image provides a set of words, which represent an observation $\mathcal{Z}_k$, which is able to maintain some invariance to view-point and lighting changes. Tracking is performed between features in subsequent image frames by comparing descriptors, and optionally refined using RANSAC. Tracked features are then represented as the same landmark, ℓ_i. A simple example of some observations, the resulting

covisibility map, and a given query observation can be seen in Fig. 2.

The current map, $\mathcal{M}_k$, is updated as information from each new image is processed. The map is implemented as a sparse clique matrix, C_k, with each column representing an observation $\mathcal{Z}_k$, and each row representing a particular landmark, ℓ_i. Therefore the value in row r, and column c indicates whether or not landmark ℓ_r was seen in observation $\mathcal{Z}_c$. An adjacency matrix, A_k, for the covisibililty graph can simply be found by taking $A_k = H(C_k C_k^T)$ (with $H(\cdot)$ being the element-wise unit step function), but is not needed explicitly in this work.

In addition to these matrices, an inverted index between visual words and observations is maintained, for efficient look-up during the creation of virtual locations [Sivic and Zisserman, 2003]. In the simple example of Fig. 2, at time k there are 4 observations ($\mathcal{Z}_1 = \{\ell_1, \ell_2, \ell_3\}$, $\mathcal{Z}_2 = \{\ell_2, \ell_4\}$, $\mathcal{Z}_3 = \{\ell_4, \ell_5\}$, and $\mathcal{Z}_4 = \{\ell_5, \ell_6\}$), then the clique matrix, adjacency matrix, and inverted index are given by:

$$C_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_4 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \quad \begin{array}{lcl} A & : & \{\mathcal{Z}_1\} \\ B & : & \{\mathcal{Z}_1, \mathcal{Z}_4\} \\ C & : & \{\mathcal{Z}_1, \mathcal{Z}_2\} \\ D & : & \{\mathcal{Z}_2, \mathcal{Z}_3\} \\ E & : & \{\mathcal{Z}_3, \mathcal{Z}_4\} \end{array}$$

2.2 Identification and Retrieval of Virtual Locations

At query time, virtual locations similar to the query image need to be retrieved from the covisibility graph, in order to be compared as a potential match. The idea is to find any clusters of landmarks in the map, which may have generated the given query. Because new virtual locations are drawn from the graph for each specific query, they are more closely linked to the actual arrangement of landmarks in the environment than individual images would be. This provides a more adaptable solution to place recognition, compared to methods which rely on pose-based location models. Defining places using covisibility avoids the need for time-based image groupings which rely on prior motion knowledge [Gálvez-López and Tardós, 2012] or more exhaustive key frame detection [Ranganathan and Dellaert, 2009].

The process of finding relevant virtual locations will now be described, with the aid of Figures 2 and 3, using the simple example introduced in Section 2.1:

- Using the inverted index, a list of observation cliques (columns in C_k), containing words from the current query observation, $\mathcal{Z}_q$, can be found. *In the example, $\mathcal{Z}_q = \{C, D, E\}$, and so the relevant observation cliques are $\{\mathcal{Z}_1, \mathcal{Z}_2, \mathcal{Z}_3, \mathcal{Z}_4\}$.*

- Then, these clusters are extended to strongly connected cliques (sharing a certain percentage of covisible landmarks). This covisibility parameter represents the probability of re-observing landmarks between images. Refer to [Mei et al., 2010] for a discussion on the influence of this parameter, and Section 2.3 of this paper for alternative clustering methods. *This will extend clique $\mathcal{Z}_2$ to $\mathcal{Z}_3$, clique $\mathcal{Z}_3$ to $\mathcal{Z}_2$ & $\mathcal{Z}_4$, clique $\mathcal{Z}_4$ to $\mathcal{Z}_3$ (which all co-observe 50% of their landmarks), and $\mathcal{Z}_1$ with nothing (because it doesn't share enough landmarks with any other cliques).*

- The result is sets of landmarks/words, which in turn, provide models for a set of virtual locations, $\{\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_M\}$. *Four virtual locations are produced for the given example, and are shown in Fig. 3.*

Note that the set of virtual locations in Figure 3 provides a direct match to the query shown in Figure 2, despite the fact that those landmarks were never directly covisible in any

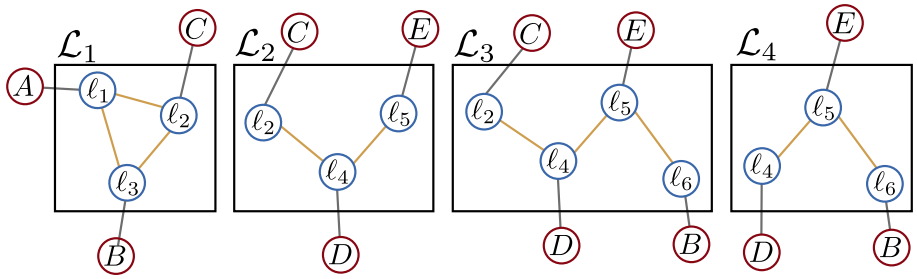


Figure 3: Given the query $\mathcal{Z}_q = \{C, D, E\}$ and covisibility map $\mathcal{M}_k$ (both shown in Fig. 2), will produce four virtual locations, $\mathcal{L}_1$, $\mathcal{L}_2$, $\mathcal{L}_3$ & $\mathcal{L}_4$.

one observation. This emphasizes the adaptive nature of virtual locations, and the continuity which is achieved through covisibility. Perhaps one limiting factor of this approach is the use of a parameter to determine the extent of clustering when forming the virtual locations. The next section discusses an alternative method of clustering, which eliminates the need for a covisibility parameter without changing the chosen probabilistic model.

2.3 Building Virtual Locations by Graph Clustering

The process of retrieving virtual locations requires some method of grouping relevant landmarks together into clusters. In Section 2.2 this was described by the use of a covisibility parameter, which extends clusters to strongly connected cliques sharing a certain percentage of covisible landmarks. This technique allows for very efficient clustering, but suffers in the sense that it relies on setting the covisibility parameter. In order to avoid using parameters for the expansion of virtual location clusters, more traditional graph clustering approaches can be used. A review of the theory surrounding graph clustering techniques is given by Schaeffer [2007]. Since clustering aims at grouping together nodes of a graph with similar properties and strong connectivity, it is inherently application dependent, and no single method is universally accepted. In addition, working with real-word image data means that the graphs tend to contain many different sources of error and noise, adding difficulty which the clustering algorithm needs to cope with.

One common metric used in clustering is the ratio of internal edges (those which connect two nodes in the cluster) to external edges (those which connect one node inside the cluster to another node outside the cluster). Therefore, landmarks are added to the cluster in a way which maximizes this parameter. In this application of forming virtual locations, landmarks are allowed to be a member of more than one location (overlap between locations is admissible), because of the normalization method discussed later in Section 3.4, simplifying the clustering process to a local (rather than global) optimization. The advantage of one is efficiency, whereas the advantage of the other is a parameter-free technique. Tests were carried out to investigate how the different clustering techniques affect the results, and the outcome (seen in Table 1 of Section 4.5) shows that results are similar for both methods of clustering.

3 Location Models

Section 2 explained how a set of candidate locations can be identified and retrieved from the covisibility graph, while the current section aims at developing a probabilistic observation model of places, in order to evaluate if any locations match. The clusters of detected visual words which were introduced in Section 2 represent observations of places, and here, a probabilistic bag-of-words approach is used to compare the query to a the set of candidate virtual locations.

The probability of a location generating the given query observation can be found using Bayes' theorem:

$$P(\mathcal{L}_i|\mathcal{Z}_q) = \frac{P(\mathcal{Z}_q|\mathcal{L}_i)P(\mathcal{L}_i)}{P(\mathcal{Z}_q)} \tag{1}$$

where $\mathcal{L}_i$ is a particular virtual location, and $\mathcal{Z}_q$ is the query observation given by a set of visual words $\{z_1, z_2, \ldots, z_N\}$. The development of each term in Equation 1 is given throughout this section. Section 3.1 explains how the existence and observation of visual elements in the scene are modeled, followed by the introduction of a novel model for the observation likelihood given a location in Section 3.2, a discussion of normalization techniques in Section 3.4, an overview of the sampling procedure in Section 3.5, and a description of how location priors are estimated in Section 3.6.

3.1 Modelling Scene Elements

An observation of the query location, $\mathcal{Z}_q$, is represented as a binary word-observation-vector of length equal to the number of words in a visual dictionary, $\mathcal{V}$:

$$\langle z_1^q, z_2^q, \ldots, z_{|\mathcal{V}|}^q \rangle$$

And the observation of a virtual location, $\mathcal{Z}_\mathcal{L}$, is represented analogously as:

$$\langle z_1^\mathcal{L}, z_2^\mathcal{L}, \ldots, z_{|\mathcal{V}|}^\mathcal{L} \rangle$$

where each z_n is set to one if the n^{th} word in the dictionary was present in the observation and zero otherwise. The visual dictionary, $\mathcal{V}$, is pre-trained with sample features, using a clustering algorithm to define a set of visual words that span the relevant feature space [Sivic and Zisserman, 2003].

Note that the negative information (lack of a word) is explicitly considered; however frequency information (word count) is removed from the observations. The reason for this is twofold. Firstly, ignoring word frequencies is justified by the fact that features tend to appear in bursts, where most of the information is provided by the presence (or lack of presence) of the word, rather than the number of occurrences of the word [Schneider, 2004]. As an example, objects such as bricks or leaves tend to be present in large multiples which would have an overwhelming effect on the outcome of comparison methods between locations, whereas seeing one leaf or brick will already give a good indication of what is present in the scene.

Secondly, these feature vectors are binary and of fixed length, allowing for a simple representation which does not vary depending on the number of words. In the context of text classification, this corresponds to the use of a Bernoulli model, as opposed to a Multinomial model which requires an assumption of fixed document length (i.e. all locations contain the same number of words), which does not hold in this context [Eyheramendy et al., 2003, Schneider, 2004].

Observations are modelled under a generative scheme, where locations are assumed to be composed of a set of existing visual elements which are observed using an imperfect sensor, resulting in our observation vectors $\mathcal{Z}$, as is also done in [Cummins and Newman, 2008] (see Figure 4 for reference). In this case, e_n is introduced as a hidden layer, which represents the true existence of scene elements generated by $\mathcal{L}_i$. The observations z_n represent (possibly imperfect) measurements of these underlying scene elements. The choice for a generative model is driven by its capability to incorporate aspects such as location priors, various types of measurements, and sensor models in a principled way which generalises well to unknown environments [Bishop, 2007].

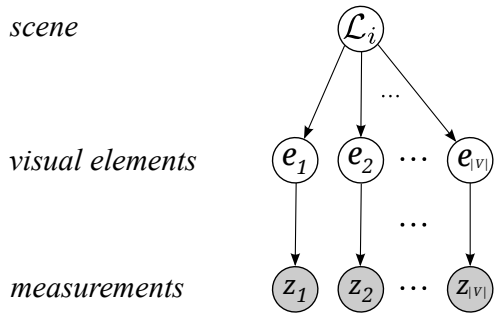


Figure 4: Graph of the observation model, with observed variables shaded in gray. A location consists of a set of visual elements e_n, which are then observed by an imperfect sensor, giving measurements z_n. The possible set of visual elements is defined by a visual dictionary of size $|\mathcal{V}|$, and visual features e_n and z_n take on boolean values based on existence or observation respectively.

3.2 Estimating Observation Likelihoods

For computational reasons, a conditional independence (Naive-Bayes) assumption is made about individual word observations, simplifying the observation model shown in Figure 4, such that when conditioned on the location, the likelihood of one word does not depend on any other words. Although this assumption is generally false, it has been shown to provide meaningful results when compared to more complex models [Cummins and Newman, 2008]. Due to this conditional independence assumption, the observation likelihood, $P(\mathcal{Z}_q|\mathcal{L}_i)$, can be reduced to a product of individual word likelihoods, as given in Equation 2a.

The observation likelihood can therefore be written using the sum rule of probability as Equation 2b, which simplifies to Equation 2c under the assumption that detection is independent of location: $P(z_n^q|e_n{=}\alpha, \mathcal{L}_i) = P(z_n^q|e_n{=}\alpha)$ (see Figure 4), and by estimating the existence of an element using the prior observations of a location $P(e_n{=}\alpha|\mathcal{L}_i) \approx P(e_n{=}\alpha|z_n^{\mathcal{L}})$ [Cummins and Newman, 2008, 2011].

$$P(\mathcal{Z}_q|\mathcal{L}_i) \approx \prod_{n=1}^{|\mathcal{V}|} P(z_n^q|\mathcal{L}_i) \tag{2a}$$

$$\approx \prod_{n=1}^{|\mathcal{V}|} \sum_{\alpha\in\{0,1\}} P(z_n^q|e_n{=}\alpha, \mathcal{L}_i) P(e_n{=}\alpha|\mathcal{L}_i) \tag{2b}$$

$$\approx \prod_{n=1}^{|\mathcal{V}|} \sum_{\alpha\in\{0,1\}} P(z_n^q|e_n{=}\alpha) P(e_n{=}\alpha|z_n^{\mathcal{L}}) \tag{2c}$$

Although the complexity grows with the number of words in the vocabulary, the sparse nature of observations can be used to greatly reduce computation in most cases. Note that the model could be further extended to remove the conditional independence assumption between words, for instance by using a Chow-Liu tree as done in [Cummins and Newman, 2008]. Doing so tends to improve results slightly in the presence of minor scene changes, but for simplicity has not been implemented here. See [Cummins and Newman, 2008] for a thorough analysis of such observation models.

The final terms in Equation 2c are defined by the underlying nature of how visual observations are made and modelled. There are a number of ways to estimate these values, and the details are discussed in the following Section 3.3.

3.3 Modelling Visual Observations

In previous work [Stumm et al., 2013, Cummins and Newman, 2011], the term $P(z_n|e_n{=}\alpha)$ represents the sensor detection probabilities, which are set as parameters by the user. For example, the true positive probability of observing an element which exists, $P(z_n{=}1|e_n{=}1)$; and the false positive probability of observing an element which doesn't exist, $P(z_n{=}1|e_n{=}0)$, are generally pre-calibrated. This leaves the likelihood of a particular element existing in the location, $P(e_n{=}\alpha|\mathcal{L}_i)$, to be estimated from the observation we have of the virtual location, $\mathcal{Z}_{\mathcal{L}_i}$, using the sensor model and prior knowledge about how common the element is [Glover et al., 2012]:

$$P(e_n{=}\alpha|\mathcal{L}_i) = P(e_n{=}\alpha|z_n^{\mathcal{L}}) \tag{3a}$$

$$= \frac{P(z_n^{\mathcal{L}}|e_n{=}\alpha)P(e_n{=}\alpha)}{P(z_n^{\mathcal{L}})} \tag{3b}$$

$$= \frac{P(z_n^{\mathcal{L}}|e_n{=}\alpha)P(e_n{=}\alpha)}{\sum_{\beta\in\{0,1\}} P(z_n^{\mathcal{L}}|e_n{=}\beta)P(e_n{=}\beta)} \tag{3c}$$

However, implementing this requires estimating $P(e_n)$, which is not possible in practise as true existence is not known. In effect, other works such as [Cummins, 2009, Glover et al., 2012, Stumm et al., 2013], substitute $P(z_n)$ for $P(e_n)$, undermining the observation model they define. One possible way around this problem, would be to redefine the observation model in terms of $P(e_n|z_n)$, rather than $P(z_n|e_n)$. This means that the problem is flipped, and now $P(e_n{=}\alpha|z_n^{\mathcal{L}})$ in Equation 2c is predefined, and $P(z_n^q|e_n{=}\alpha)$ is calculated as follows:

$$P(z_n^q|e_n{=}\alpha) = \frac{P(e_n{=}\alpha|z_n^q)P(z_n^q)}{P(e_n{=}\alpha)} \tag{4a}$$

$$= \frac{P(e_n{=}\alpha|z_n^q)P(z_n^q)}{\sum_{\beta\in\{0,1\}} P(e_n{=}\alpha|z_n{=}\beta)P(z_n{=}\beta)} \tag{4b}$$

This removes an explicit inclusion of $P(e_n)$, as well as having a number of other consequences. Now, the required parameters are all defined in the sense of having access to z_n, the *observed* variable, rather than e_n, the *hidden* variable. This includes the probability of existence *given* the observation, $P(e_n|z_n)$.

In order to investigate the empirical effect of these changes, the FAB-MAP code [OxfordMRG, 2013] was augmented to incorporate the new model, and both versions were tested across a range of parameters (with parameter settings $P(z_n{=}1|e_n{=}1) > P(z_n{=}1|e_n{=}0)$ and $P(e_n{=}1|z_n{=}1) > P(e_n{=}1|z_n{=}0)$ in each case). The results are seen in Figure 5, where the value of maximum recall at 100% precision was plotted for varying parameter settings, for each model, for four different datasets (Begbroke, City Centre, New College, and KITTI). The plots in the left column show the recall results for fixed values of $P(z_n|e_n)$ (corresponding to Equation 3), and the plots in the right column show the recall results for the novel model where $P(e_n|z_n)$ is fixed (corresponding to Equation 4). From these plots, one can see that recall results remain stable across a wider selection of parameters, reducing the sensitivity of results to parameter settings while maintaining recall performance. Based on these experiments, the parameters $P(e_n|z_n) \approx 0.8$ with $P(e_n|\bar{z_n}) \approx 0.3$ are of interest, with finer scale tests providing $P(e_n|z_n) = 0.78$ with $P(e_n|\bar{z_n}) = 0.32$ as the selected parameters. All results shown in Figure 5

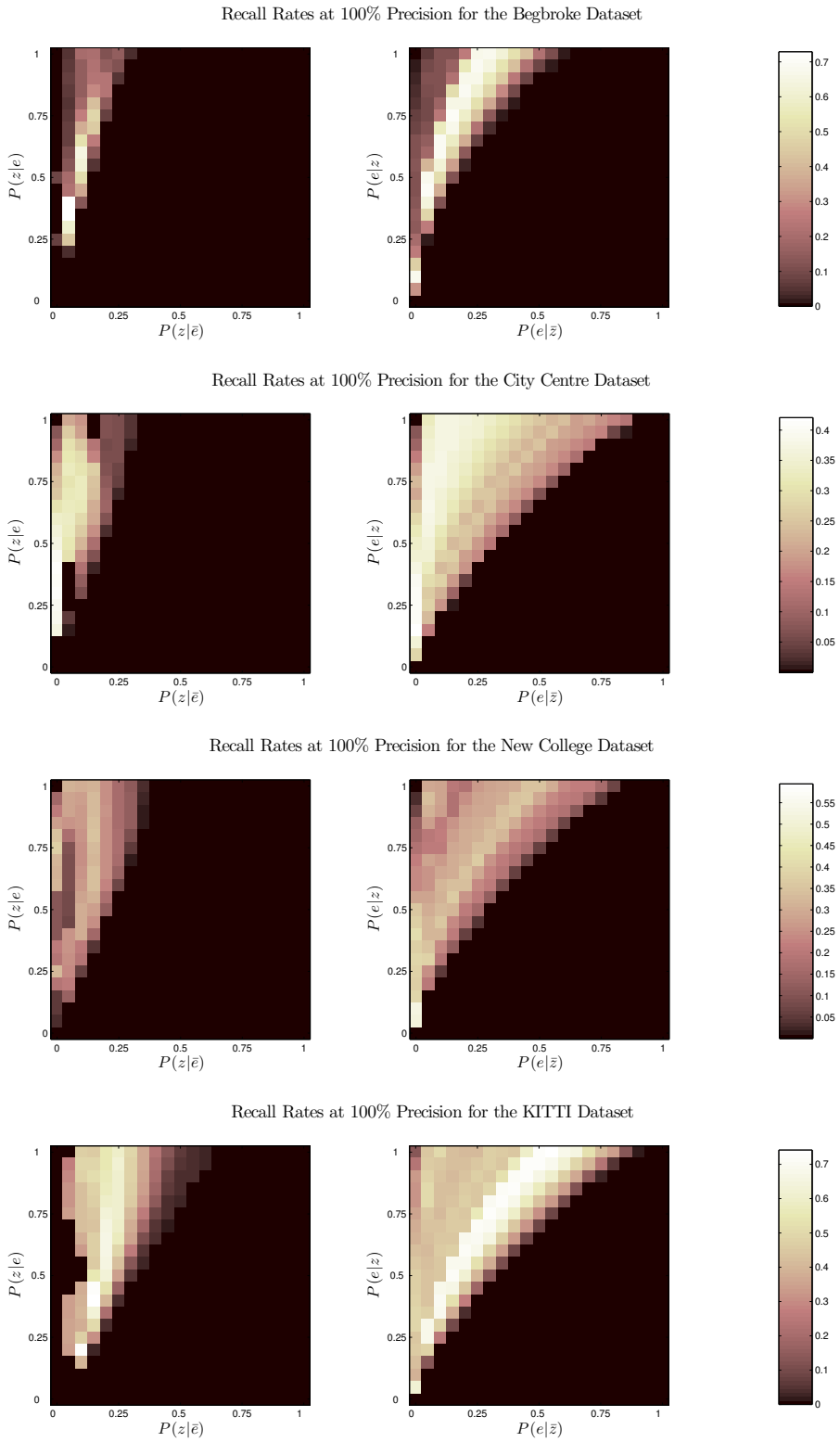


Figure 5: A comparison of maximum recall results at perfect precision for a variety of possible parameter settings, computed for four different datasets. The left column shows the results of the model implied by Equation 3, whereas the right column shows the results of the model implied by Equation 4. Colour bars are also given to indicate recall values for each dataset.

were obtained using the Naive-Bayes version of FAB-MAP, with the default motion model and default sample data. In addition, the ground truth was given by accepting all loop-closures within a generous radius of 10 m (to compensate for some significant errors in the GPS data), and the 10 most recent images were masked from consideration.

3.4 Normalization using Sample Locations

Working with true (normalized) probabilities is essential for the decision making process in the context of loop-closure for mobile robots, as false loop-closures result in fundamental mapping and localization errors. However, accurate normalization in the presence of such complex and high-dimensional observations of scenes requires careful treatment using previously obtained sample observations [Bishop, 2007] which will be explained throughout this section.

The formulation presented here differs from the typical treatment of classification problems, where only the best (maximizing) class is assigned to an observation (as in Equation 5, for example), and therefore normalization is not required and rarely calculated in practice.

$$c_{MAP} = \arg\max_{c \in \mathbb{C}} P(c|z) = \arg\max_{c \in \mathbb{C}} \frac{P(z|c)P(c)}{P(z)} = \arg\max_{c \in \mathbb{C}} P(z|c)P(c) \tag{5}$$

where $c \in \mathbb{C}$ is a set of classes, z is a given observation, and c_{MAP} represents the maximum aposteriori estimate for the class [Manning et al., 2008].

However, within the application of place recognition and loop-closure, the number of locations which match the query observation is unknown, and there may even be no matches; meaning that it would be incorrect to always associate the maximizing location to the query. In addition, the severity of making any incorrect data associations can be further motivation for basing decisions on the results of posterior probabilities, only fusing locations if the probability lies above a certain confidence threshold. Therefore, the denominator in Equation 1, $P(\mathcal{Z}_q)$, is required. As previously mentioned, due to the high-dimensional nature of visual observations of places, estimating this term in practice requires the use of sample locations. These sample locations function as a representation of all other locations in the world, $\bar{\mathcal{L}}_i$. Using samples, the likelihood of the observation coming from any other place, $P(\mathcal{Z}_q|\bar{\mathcal{L}}_i)$, is calculated and then $P(\mathcal{Z}_q)$ is found through marginalization:

$$P(\mathcal{Z}_q) = P(\mathcal{Z}_q|\mathcal{L}_i)P(\mathcal{L}_i) + P(\mathcal{Z}_q|\bar{\mathcal{L}}_i)P(\bar{\mathcal{L}}_i) \tag{6}$$

Implicitly, the sample locations provide an indication as to how common or ambiguous an observation is, and help deal with a problem known as perceptual aliasing. As an example, in a city, something like a brick wall does not provide much information about where one might be since it is commonly seen throughout cities. On the other hand, a distinct fountain or sculpture will. In the same way, a representative group of sample locations should augment the posterior probability according to how often elements of the observation were seen throughout the samples. A more detailed intuition behind working with sample locations is given in Section 3.5.

Note that this method of normalization makes no reliance on the robot's current map, since it makes no assumption on the number of matching locations in the map, or the number of previously seen locations. This provides an advantage to other place recognition frameworks such as FAB-MAP [Cummins and Newman, 2008] which normalize over all locations in the map (plus an unknown location, $\mathcal{L}_u$). In these approaches, probabilities are summed across locations:

$$P(\mathcal{L}_1|\mathcal{Z}_q) + P(\mathcal{L}_2|\mathcal{Z}_q) + \ldots + P(\mathcal{L}_u|\mathcal{Z}_q) = 1 \tag{7}$$

whereas here in this work,

$$P(\mathcal{L}_i|\mathcal{Z}_q) + P(\bar{\mathcal{L}}_i|\mathcal{Z}_q) = 1. \tag{8}$$

Equation 7 is based on an underlying assumption that each location is only represented once, thereby assuming no loop closures will be missed, and that the map is accurate. As previously

mentioned, there may in fact be more than one match to the query. This can happen when a previous loop-closure is missed – leaving two or more representations of the location in the covisibility map. In addition, images immediately surrounding the query do not need to be masked (removed from consideration) as commonly done ([Cummins and Newman, 2011, Angeli et al., 2008]), since these local matches will not steal probability mass from others. Another benefit of this technique is that probabilities no longer need to be normalized over all locations in the map, leaving room for efficiency improvements over other techniques. It should be noted, however, that this method of normalization opens up the risk of obtaining more than one false-positive loop-closure per location, whereas in previous work, at most one loop-closure could be detected per location.

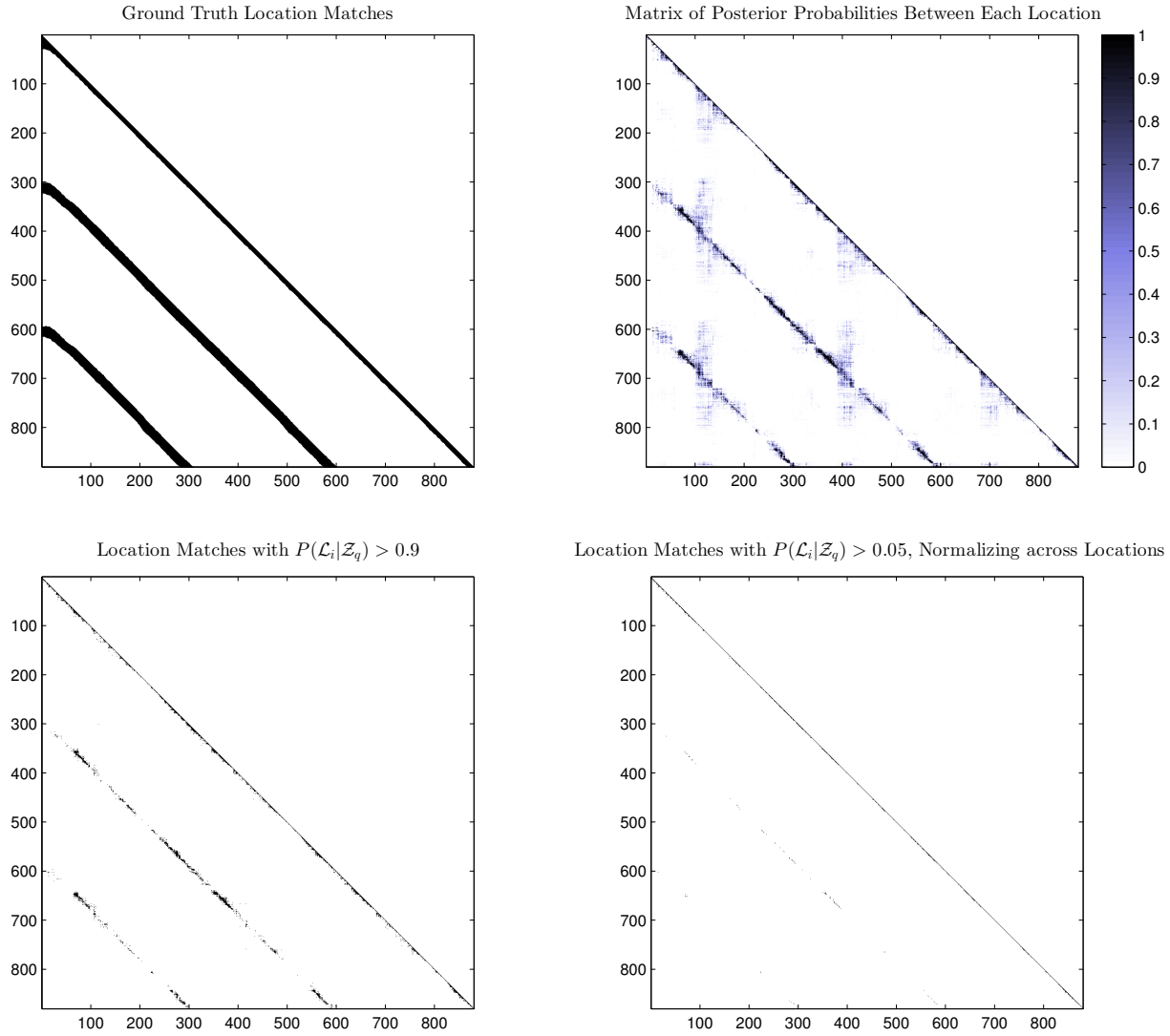


Figure 6: Depiction of place recognition results for the three-loop Begbroke dataset, using two different normalization methods. The first plot shows the ground truth location matches. The second and third plots show the results given by $P(\mathcal{L}_i|\mathcal{Z}_q)\Big|_{Eq.8}$. In the second plot, probability scores are indicated by the colour bar, whereas in the third plot scores are thresholded at 0.9. The fourth plot shows the results given by $P(\mathcal{L}_i|\mathcal{Z}_q)\Big|_{Eq.7}$, thresholded at 0.05. Note that when normalizing across all locations in the map (fourth plot), the probability mass is split accordingly, therefore generally resulting low values.

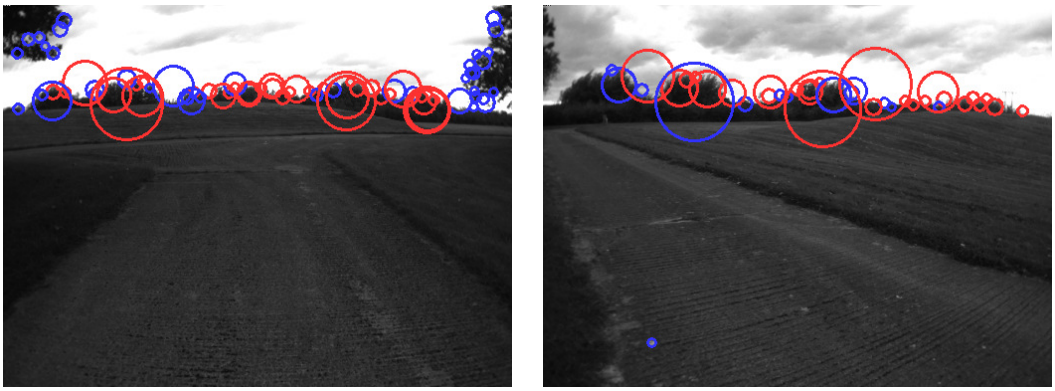
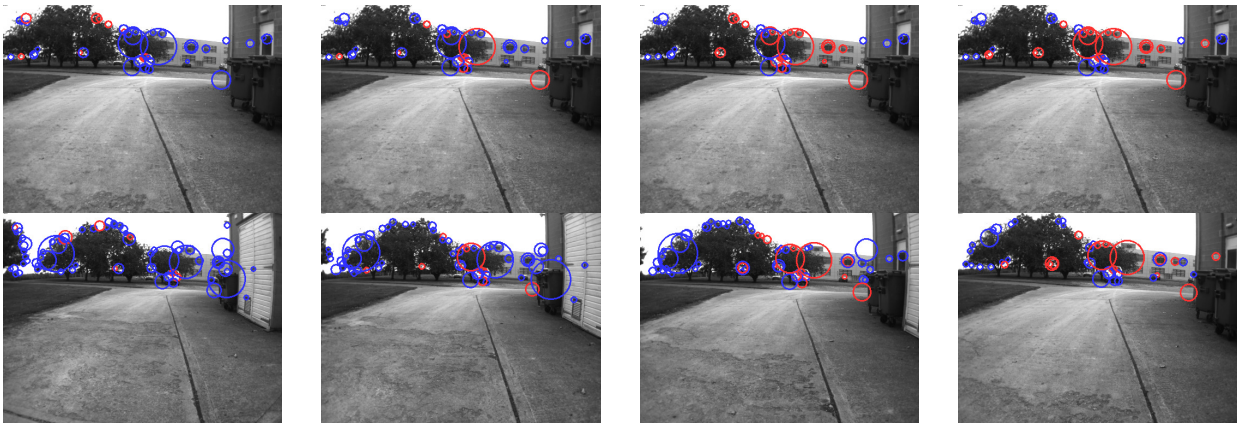


Figure 7: Examples of images from the Begbroke sequence which correspond to the regions of confusion in first matrix of Figure 6, due to their non-discriminative appearance. The images contain many matched visual words (shown in red), despite depicting different locations. However, these locations tend to produce low matching probabilities regardless, which indicates a low confidence due to common appearance.

In order to compare the effects of each normalization method, the FAB-MAP code [OxfordMRG, 2013] was augmented again, to incorporate the normalization method given by Equation 6. Figure 6 shows the results for the Begbroke dataset using the two different normalization methods. The first plot shows the expected ground truth based on GPS data, with location matches shown by black lines. The second plot shows the matrix of posterior probability values between each location under the marginalization scheme of Equations 8, with values shaded between 0 (white) and 1 (black). The third and fourth plots show the location matches resulting from thresholding the probabilities ($P(\mathcal{L}_i|\mathcal{Z}_q)\Big|_{Eq.8}$ and $P(\mathcal{L}_i|\mathcal{Z}_q)\Big|_{Eq.7}$) at 0.9 and 0.05 respectively, where the third plot is made using the normalization given by the marginalization of Equation 8 and the fourth plot is made using the marginalization of Equation 7. Results show that the chosen method allows for multiple location matches, whereas the other method has trouble dealing with many instances of the same location in the map. In these tests, the Naive-Bayes version of FAB-MAP was used, with the default sample data. However, no motion model was used, and there was no masking of recent images, in order to exclude effects of strong assumptions on motion.

Further investigation into the second plot of Figure 6, indicates that the areas where the diagonals are thinner correspond to locations where the robot turns corners (leaving fewer visual matches, given a forward-facing camera). In addition, the faint vertical streaks of confusion in the probability matrix correspond to locations which do not have a discriminative appearance. Figure 7, shows two examples of images from this region, where one can see many matched visual words (shown in red, rather than blue), despite showing different features. However, referring back to Figure 6, one can see that despite many matched words, the final match probability remains low, which can be seen as the streaks disappear completely after thresholding. This also serves as an illustration of how normalization can deal with the problem of perceptual aliasing.

Another implication of this normalization technique is shown in Figure 8. Here the matched visual words (shown in red rather than blue) and resulting posterior probabilities are shown for a series of images, as a location which matches the query is approached. Under the normalization scheme of Equation 7, at most one location in the map can have a significant probability mass, leaving the rest with low scores (in this scenario, the most recent location in the map has a score of $P(\mathcal{L}_i|\mathcal{Z}^q) = 0.6$). When probabilities are no longer normalized across all locations in the map, the behaviour of probabilities becomes more intuitive, since now locations with similar appearances can have similar scores and there is a natural progression of probabilities as a location is approached.



position - 3 m	position - 2 m	position - 1 m	position - 0 m
$P(\mathcal{L}_i\|\mathcal{Z}_q)\Big\|_{Eq.7} = 4.0e^{-5}$	$P(\mathcal{L}_i\|\mathcal{Z}_q)\Big\|_{Eq.7} = 5.0e^{-5}$	$P(\mathcal{L}_i\|\mathcal{Z}_q)\Big\|_{Eq.7} = 1.4e^{-4}$	$P(\mathcal{L}_i\|\mathcal{Z}_q)\Big\|_{Eq.7} = 3.6e^{-3}$
$P(\mathcal{L}_i\|\mathcal{Z}_q)\Big\|_{Eq.8} = 0.05$	$P(\mathcal{L}_i\|\mathcal{Z}_q)\Big\|_{Eq.8} = 0.20$	$P(\mathcal{L}_i\|\mathcal{Z}_q)\Big\|_{Eq.8} = 0.71$	$P(\mathcal{L}_i\|\mathcal{Z}_q)\Big\|_{Eq.8} = 0.99$

Figure 8: Demonstration of matched features and posterior probability values as a location which matches the query is approached. The query image is repeated in the top row, with four other locations preceding a match to the query shown in the second row. Matched visual words are shown in red, while unmatched visual words are shown in blue. The resulting match probabilities are shown below each image pair.

3.5 Sampling

As discussed in Section 3.4, the success of this framework relies on correct normalization of posterior probabilities, which is done here using sampling techniques. Sampling from such a high-dimensional space poses a variety of difficulties, including obtaining representative samples, modelling the sample locations properly, and working efficiently with the samples.

The normalization term in Equation 6 requires $P(\mathcal{Z}_q|\bar{\mathcal{L}}_i)$ which is calculated from the sample set as follows,

$$P(\mathcal{Z}_q|\bar{\mathcal{L}}_i) = \sum_{s=1}^{N_s} \frac{P(\mathcal{Z}_q|\mathcal{L}_s)}{N_s} \tag{9}$$

with N_s being the number of samples, $\mathcal{L}_s$ being the s^{th} sample location, and $P(\mathcal{Z}_q|\mathcal{L}_s)$ is subsequently given by Equations 2 and 3.

One straightforward way to generate sample locations is to collect images from the robot in the same manner as the typical use-case scenario. However, this may not always be possible, and it is often difficult to represent the full range of possible scenes. Therefore it may be necessary to combine sample images from a variety of experiments, as well as other resources such as online map repositories. This task benefits from prior knowledge about the types of environments that the robot will operate in, and should contain the scope of features which the robot is expected to encounter. These samples are what allows the system to understand which features are distinctive and avoid perceptual aliasing, by augmenting the impact of features based on how common they are in the data. Note that sample locations require no extra processing in comparison to the run-time virtual locations, since they require no supplementary labels or information.

For the experiments presented in this work, samples were taken from a variety of publicly available datasets, as well as images obtained from Google Street View[1]. The same samples were reused across each test scenario (with the exclusion of images coming from the corresponding test set).

[1] https://www.google.com/maps/views

One of the difficulties in using samples to estimate the likelihood of unknown locations is knowing how many samples are required in order to get a reasonable estimate. Unfortunately this remains an open question. There are no guarantees on the quality of the estimate, and the number of samples required will vary, depending on the extent of the world which the robot will operate in. In general, the more samples that are available, the better.

Experience from the work presented here, as well as that documented in [Cummins and Newman, 2008, 2011], shows that useful results can be obtained using a feasible number of sample locations, especially if some prior knowledge is known about the expected environment and therefore samples to use (*e.g.* urban, indoor, rural, etc.). For the results presented in this paper, approximately 3000-5000 samples were used (the amounts vary across experiments because images from the current test set were removed from the sample set in each case).

3.6 Location Priors

For the framework presented in this work, the location prior is estimated without the use of any motion prediction models. This is in part due to the fact that this work is ideally meant to remain robust to unpredictable movements and kidnapped robot situations. In practice, the effect of this prior is not especially strong, and it is therefore not a critical parameter. This is evident when comparing the order of magnitude of the observation likelihood (a product of probabilities over thousands of visual words) to that of a location prior. The weak influence of this term is also documented in Cummins and Newman [2008]. Therefore most of the prior probability is assigned to unobserved locations, conservatively favoring unobserved locations (to avoid false positives). Other cues could be used to more accurately estimate the location prior; such as global visual features or additional sensory information.

4 Experiments and Discussion

In order to analyse the performance of the proposed approach (hereby referred to as CovisMap), it was tested on each of the datasets described in Appendix A. The system is also compared to two widely known place recognition methods, FAB-MAP [Cummins and Newman, 2008] and SeqSLAM [Milford, 2013], to investigate the implications of the various assumptions and the relative behaviour of each system. To begin with, the implementation details are explained in Section 4.1. Next, precision-recall characteristics are discussed in Section 4.2, with direct comparisons to FAB-MAP and SeqSLAM. This is then followed by detailed discussions on normalization methods, trajectory invariance, and location retrieval in Sections 4.3-4.5.

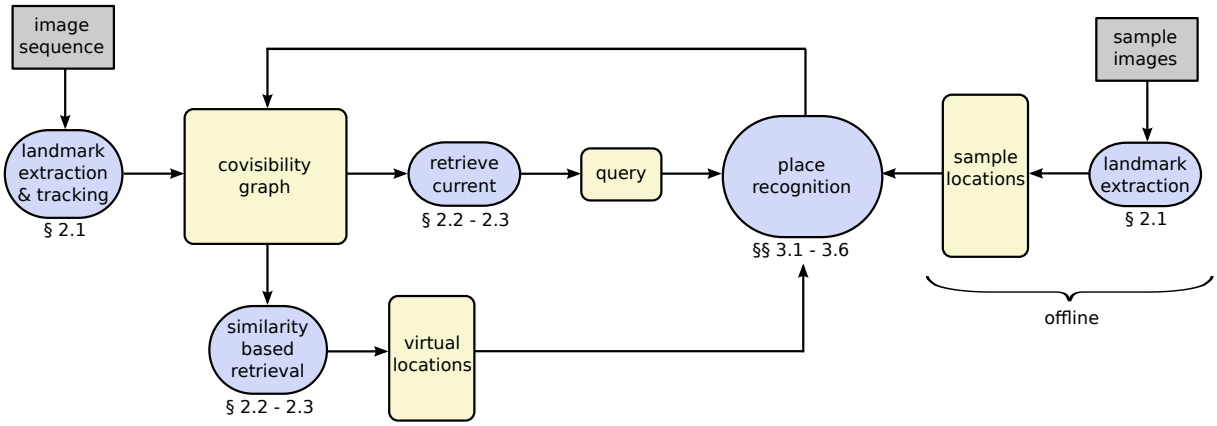


Figure 9: General scheme of the framework, with inputs shown by grey boxes, data structures shown by yellow boxes, and algorithmic blocks shown by blue ovals.

4.1 Implementation

The CovisMap framework presented in this paper essentially processes a stream of images, testing for place recognition using the current location, and updating the covisibility map at each time step. However, in order to better understand and evaluate the performance of each method, no data association from loop-closures is actually done during testing (similarly to the tests in [Cummins and Newman, 2011]). Figure 9 provides an overview of this process. The inputs to the system, shown by grey, square boxes, are the image sequence provided by the mobile robot, and a set of sample images as described in Section 3.5. Data structures are shown in yellow, rounded boxes, which include the covisibility graph, and all of the locations (query location, virtual locations, and sample locations). Then, the algorithmic blocks are shown by blue ovals, with relevant sections from the paper listed alongside. While the image stream from the robot must be processed during run-time, the processing of the sample locations (feature detection, extraction, etc.) can be done ahead of time. Also note that rather than using single-image queries, query locations are expanded analogously to the way in which virtual locations are formed, as described in Section 2.2 and Section 2.3. This query expansion process provides more context and suppresses false positives, similar to concepts used in text and image retrieval [Chum et al., 2007], but can take advantage of covisibility to greatly simplify the expansion process.

The bulk of the computation lies in the place recognition block, since location retrieval can be done efficiently using the inverted index. However, the computation does not grow directly with the number of locations (like most other systems), since normalization does not require all locations in the map, and in general, only a small subset of areas in the covisiblity map are retrieved as candidate virtual locations.

Even though each system is tested on several datasets, one important point is that the system parameters are consistent across all of the tests. The only difference in the test configuration for each dataset is the set of sample locations used (only for CovisMap and FAB-MAP since SeqSLAM does not use any samples). This is because, in each case, the samples include images from the other datasets, but images from the tested dataset are not included in the samples. However, since both CovisMap and FAB-MAP require the use of sample locations, the same sample set is always used for both methods, with a different sample set for each dataset.

In the case of FAB-MAP, the implementation by the original developers [OxfordMRG, 2013] was used for testing, while in the case of SeqSLAM, a modified (in order to match the descriptions by Milford [2013]) version of OpenSeqSLAM was used [Suenderhauf, 2013]. For both FAB-MAP and SeqSLAM, parameters settings are given by descriptions in [Cummins and Newman, 2008] and [Milford, 2013] respectively. Therefore in FAB-MAP, the detection probabilities are set as $P(z_n|e_n) = 0.39$ and $P(z_n|\bar{e_n}) = 0.0$. However, in order to maintain consistency across different methods, no motion model or image masking was used. In addition, the naive-bayes implementation was used, rather than that which uses the Chow-Liu tree. As in [Cummins and Newman, 2008], both FAB-MAP and CovisMap use SURF features.

For the SeqSLAM tests, the sequence length was set to 50 frames and the image resolution was always kept well above the documented threshold for performance degradation. In addition, the difference matrix was locally normalized using a radius of 20 frames, as documented. Finally, in order to be able to cope with the speed or frame-rate variations in the datasets, the slope for sub-route searches was varied between 0.25 and 4 for all tests.

Using the test results from Section 3.3, the detection probabilities for the CovisMap implementation were set to $P(e_n|z_n) = 0.78$ and $P(e_n|\bar{z_n}) = 0.32$. The covisibility parameter was set to 5%, and the percentage of observed words for candidate virtual location retrieval was set to 4% (although these settings are not very critical, relative to the detection probabilities).

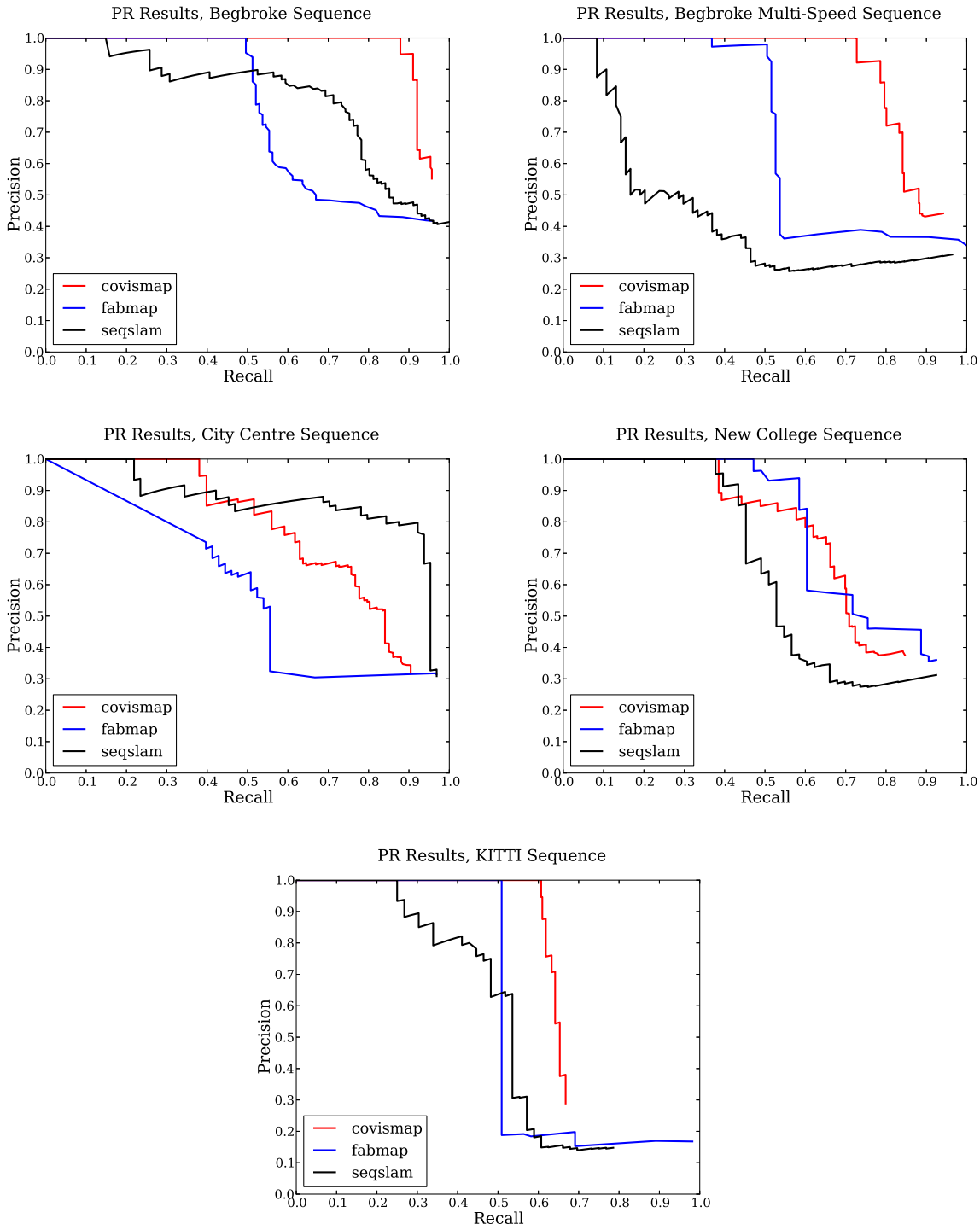


Figure 10: Precision-recall results for three methods (CovisMap, FAB-MAP, and SeqSLAM), tested on the Begbroke, Begbroke Multi-Speed, CityCentre, New College, and KITTI datasets.

4.2 Comparison with State-of-the-Art

This section presents precision-recall results on five different datasets, for CovisMap, FAB-MAP, and SeqSLAM. As mentioned in Section 1.2, for robotic SLAM applications, the primary goal is to increase the achievable recall rate while maintaining perfect precision. The details of the definition and calculation of precision-recall curves is given in Appendix B. Each result is discussed, and example images are given to provide insight about how each system performs.

Figure 11: A subset of images from a false positive sequence match example from SeqSLAM on the Begbroke dataset.

Figure 12: Example of a significant perspective changes in the Begbroke dataset. These perspective changes introduce difficulties for algorithms which rely on direct image comparisons, such as SeqSLAM.

Figure 10 shows the precision-recall curves for the Begbroke, Begbroke Multi-Speed, City-Centre, New College, and KITTI datasets as the loop-closure threshold is varied. The details of the precision-recall evaluation are described in Appendix B. One can see that in general, the CovisMap framework achieves a higher recall, especially while maintaining 100% precision.

When comparing the two Begbroke datasets, recall results drop for each system on the Multi-Speed sequence. The drop in recall is most severe for the SeqSLAM dataset, which relies on consistent sequences to produce good scores. In both Begbroke datasets, SeqSLAM and FAB-MAP tend to suffer from perceptual aliasing more than CovisMap. In the case of FAB-MAP, perceptual aliasing arises because of the use of single-image locations which often have very similar appearance (see Figure 7 for example). SeqSLAM, on the other hand, has difficulty because many incorrect sequences can look similar on the level of global image comparisons (see Figure 11), while correct matches often look different due to changes in perspective (see Figure 12).

The City Centre dataset is also challenging for all three algorithms because of perceptual aliasing. Figure 13 shows the kind of extreme examples of perceptual aliasing that arise when using the bag-of-words methods in FAB-MAP and CovisMap which ignore position information from the features. Other than a few high-scoring false-positives, the SeqSLAM is able to obtain a very high recall rate at almost 90% precision because it does not suffer from such feature-based aliasing. However, these kinds of false-positives can generally be suppressed by post-processing matched locations with a check for geometrical consistency [Sivic and Zisserman, 2003, Cummins and Newman, 2011]. This is not done here, in order to maintain a clear analysis of the underlying

Figure 13: Example of two scenes which cause false-positives due to perceptual aliasing in the City Centre dataset.

Figure 14: Due to the larger image spacing in the City Centre and New College datasets, there are often significant position offsets, creating difficulties for the SeqSLAM algorithm.

approaches.

For both the City Centre and New College datasets, the relatively large image spacing means that the direct image comparisons used in SeqSLAM may give low scores because of position offsets. Refer to Figure 14 for an example from the New College dataset, where the same location can have strong image difference scores.

Performance on the New College Dataset is pretty similar for all three methods. CovisMap has difficulty distinguishing between different positions when there are scenes with a wide visibility and repetitive features. Figure 15 shows this kind of false-positive response given by CovisMap. However, although during testing these are considered to be two different locations, there are indeed many common landmarks between the two scenes. This raises the question of how to best evaluate loop-closure, since some applications may require accurate position-based matches, while others only require matched landmarks. The loss of structure when using a bag-of-words model reduces the distinctiveness of each location, and increases the risk of false-positives. False-positives like that in Figure 15, or even more severe mismatches could possibly be avoided by including more structure during comparison. Ideas related to this are subject of current on-going work.

In the New College and the KITTI datasets, there are often relatively short sequences of loop-closures (unlike the Begbroke and City Centre datasets which mostly traverse the same sequence multiple times). This can result in recall problems for the SeqSLAM algorithm which relies on a fixed sequence length. Figure 16 shows an example of this from the KITTI dataset, where the middle portion of two sequences overlap, while the beginning and ends of the sequences vary, resulting in a relatively low score.

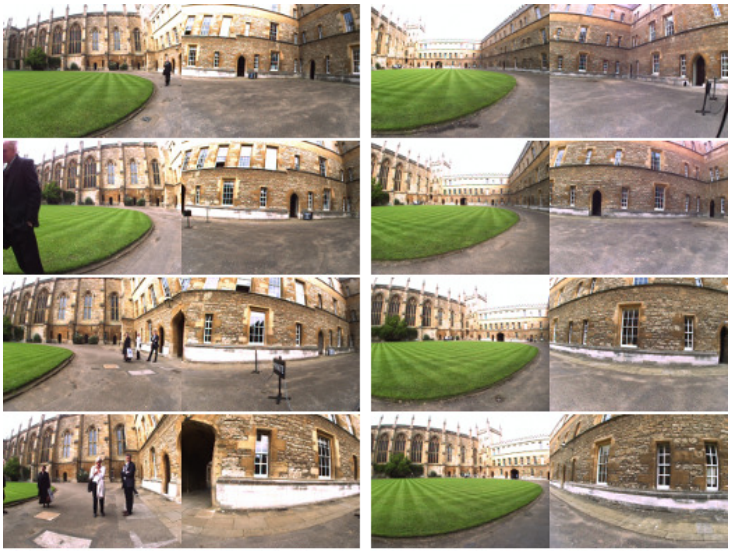


Figure 15: Locations with a wide visibility and repetitive features can cause false-negatives under the bag-of-words model, especially when considering larger sets of images. The loss of structure in the bag-of-words model reduces the distinctiveness of each location and causes perceptual aliasing. Here is a false-positive example given by the CovisMap algorithm on the New College dataset The query location is shown on the left, and the retrieved location on the right (only a subset of images from each location are shown).

4.3 Investigating Relative Scoring Methods

The benefit of rigorous probabilistic methods are especially clear when looking at the thresholds used to determine matches. Figure 17 shows the scores from several passes by a query location in the Begbroke sequence from each of the tested methods. In this example, FAB-MAP struggles to recognize all instances of the location because the posterior probabilities must be normalized across all locations (refer to Section 3.4 for details). Therefore, most of the probability mass is assigned to the most similar location just before the query location, resulting in low probabilities being assigned to other instances of the same location. In the case of SeqSLAM, the scores are still normalized between 0 and 1, but do not represent probabilities. In the example in Figure 17, each other instance of the query location receives scores between $0.2 - 0.8$, whereas the negative location example has a score of about 0.1. The generative model developed in Section 3 allows the CovisMap algorithm to retrieve each instance of the query location with posterior probabilities above 0.99, while assigning a near zero probability to the negative location.

4.4 Investigating Trajectory Invariance

In this section, the system is investigated, to see whether it can handle changes in the motion between different traverses of the same area; namely offsets in position and variations in speed (or image frame-rate). This is done by traversing a section of street in several different ways: once quickly along the right side of the street, once quickly along the left side of the street, once down the right side of the street with inconsistent speed, and once down the left side of the street with inconsistent speed (including backtracking). The chosen street is completely surrounded by houses on either side, in order to present significant perspective changes between traverses on the right and left sides of the street. The images were collected using a basic hand-held camera, and therefore the images only roughly point in the same direction in each traverse, adding even more variation. Example images of the environment can be seen in Figures 18 and 19. In this scenario, the system is shown to implicitly handle variations in speed (since locations are created based on covisibility and not a predefined number of images or time scale), as well as perspective

Figure 16: Example of a loop-closure sequence which is shorter than the sequence length used in the SeqSLAM algorithm, resulting in relatively low scores and therefore missed loop-closures. The left column shows the image sequence from one pass, and the right shows the image sequence from a second pass (only a subset of images from the sequence are shown).

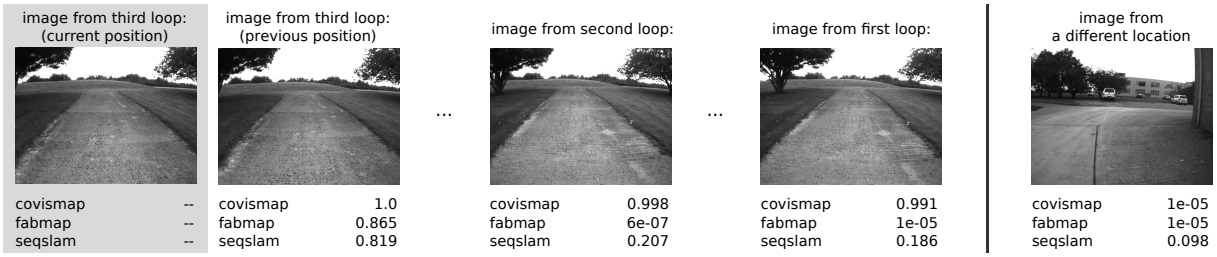


Figure 17: Example of a location from the Begbroke sequence which is passed several times, and the matching scores resulting from a query generated during the third pass. Scores are shown for the system described in this paper (CovisMap), FAB-MAP [Cummins and Newman, 2008], and SeqSLAM [Milford, 2013]. The query location is shown on the left in grey, followed by the image just before the query location, an image from the previous pass, an image from the first pass, and a negative location example.

changes (when relying on feature based models).

Figures 18 and 19 show two examples of queries, along with the virtual locations deemed most probable from each previous traverse of the street (only a few representative images from the start, middle, and end of each location are shown for clarity). In both cases, the query locations were collected from more erratic traverses, where speed and sometimes direction was varied (although the camera was always facing forward). Note that there were no data associations between any separate traverses of the street, and therefore each individual sequence is independent from the others.

One can see that even though each instance of the same location can contain varying amounts of images, the scope remains consistent in each case. Additionally, the corresponding probabilities reflect the degree of similarity, while producing no false positives above 99%. The system defines locations based on covisible features, an attribute which should theoretically be independent of direction and speed (given that there remains enough overlapping features between images), allowing the system to implicitly cope with such variations. The algorithms use of features also allows the system to deal (at least to some extent) with changes in view-point.

Figure 18: Given the query shown in green, the most probable virtual locations from two separate traverses of a street are shown (with only three images displayed per location for clarity). Both the query and the first pass are traversed on the right side of the street, while the second pass is traversed on the left side. In addition, the images collected from the query were generated from a much slower and erratic traverse, resulting in more images representing the same traverse yet poses no problems for the system.

Figure 19: Given the query shown in green, the most probable virtual locations from three separate traverses of a street are shown (with only four images displayed per location for clarity). Both the query and the second pass are traversed on the left side of the street, while the first and third passes are traversed on the right side. In addition, the images collected from the query were generated from a much slower and erratic traverse (including backtracking), resulting in more images representing the same traverse yet poses no problems for the system.

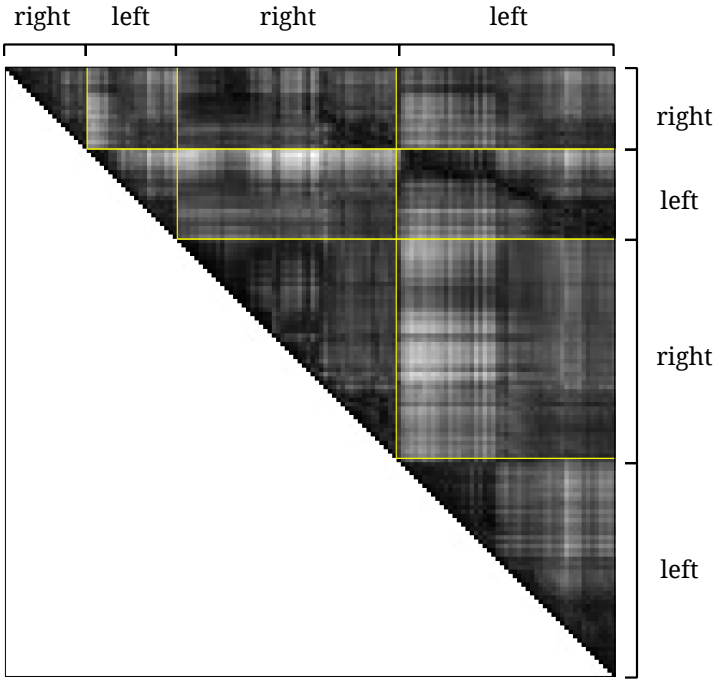


Figure 20: The confusion matrix when comparing differences (SAD) between all the images of the street sequence, with lighter pixels indicating larger differences. Yellow lines indicate the start of each new traverse down the street. Only half the matrix is shown because of symmetry. The SeqSLAM algorithm [Milford, 2013] searches for diagonals of low difference values in the matrix.

These points contrast algorithms such as SeqSLAM [Milford, 2013] which rely on temporal sequences of direct image difference values such as sum of absolute difference (SAD). Figure 20 illustrates the confusion matrix of the SAD values from each image in the four traverses of the same street. In the case of limited perspective change and consistent speed, repeated sequences should manifest themselves as diagonals of low difference scores in the matrix (dark pixels here). This image highlights where each traverse begins and ends with yellow lines and corresponding labels. It can be seen that very few of the blocks in the confusion matrix show strong diagonal features. The strongest diagonal appears when comparing the two traverses along the left side of the street, but even here, the diagonal does not have consistent slope, as a result of the speed variations in the second left pass. Although such algorithms suffer from these disadvantages, there are a number of alternative benefits; these include robustness to low quality and blurred images, reduced computational complexity, and no reliance on training samples.

Dataset Name	Fixed Covisibility	Clustering Approach
Begbroke	0.88	0.84
City Centre	0.38	0.32
New College	0.38	0.43
KITTI	0.61	0.62

Table 1: Maximum recall results at perfect precision for a version of the CovisMap algorithm which uses a fixed covisibility parameter and a version which uses a clustering algorithm rather than using parameters.

4.5 Investigating Graph Clustering

Section 2.3 discussed how clustering landmarks based on the relative number of internal and external edges can be done to form virtual locations. In order to compare this method of clustering with that of expanding locations using a set covisibility parameter, precision-recall results were calculated for the four main datasets, and Table 1 shows the maximum recall results at perfect precision for each case. Based on these results, one can see that both methods maintain similar performance, and that compared to the other frameworks, the minor changes in recall at perfect precision is not enough to change the relative ranking of any frameworks (see Figure 10).

5 Concluding Remarks

This paper has studied the task of appearance-based place recognition; primarily investigating various representations of places and the corresponding models based on visual observations. In order to group landmarks together in a relevant way, a covisibility graph is created, and clusters within the graph represent locations. This method is shown to inherently able to cope with variations in robot trajectories, including irregular changes in speed, direction, and viewpoint. A detailed analysis into probabilistic observation models was used to improve robustness to error and parameter sensitivity. The resulting generative model provides the posterior probability of being in a certain location, given a particular observation, in a way which does not require normalizing over the entire map and is able to find multiple instances of a location. The framework is useful for loop-closure detection, recovery from the kidnapped robot problem, map fusion, and topological mapping. Therefore the methodology is especially useful in applications where the robot travels in unconstrained environments, or even using unconventional modes of travel such as elevators or trains (equivalent to the kidnapped robot problem, where sensing egomotion becomes very difficult).

Future work includes a study of how to incorporate the covisibility structure into the observation model. For example by using ideas related to graph clustering and graph matching. This should help reduce remaining perceptual aliasing problems which can occur when large locations become ambiguous under the bag-of-words model. Incorporating semantic information could be used to further exploit the covisibility map, by learning to provide labels and relationships between different types of locations. In addition, because of the reliance on feature-based techniques, poor performance might occur when working with blurry or low quality images, and the framework could therefore benefit from improvements in feature detection, description and quantization.

Appendices:

A Datasets

A total of six different datasets of image sequences were used in order to analyse the behaviour of the systems. Table 2 provides a summary of each of these datasets, with a few representative example images shown in Figure 21.

The two Begbroke sequences listed in Table 2 actually contain images from the same dataset, but using different subsampling of a high-framerate image stream. Both of these sequences are good for investigating place recognition, as the robot made three loops around a path, passing each location three times, therefore giving three instances of each place. Some challenges involved in the Begbroke dataset include a high repetition of scene elements (trees, bushes, grass, paved

Dataset Name	Description	Sequence Length	Image Spacing	Image Specs
Begbroke	3 loops around a path surrounded by fields, trees, buildings, and cars.	approx. 1km, 1000 images	approx. 1m	forward facing, greyscale, 512×384px
Begbroke Multi-speed	3 loops around a path surrounded by fields, trees, buildings, and cars. Each loop is given by a different framerate.	approx. 1km, 1000 images	approx. 0.5m, 1.0m, 2.0m	forward facing, greyscale, 512×384px
City Centre	University campus with many buildings, cars, roads, gardens, and people.	approx. 2km, 1200 images	approx. 1.6m	left and right, colour, 640×480px each
New College	University campus with many buildings, cars, roads, gardens, and people.	approx. 2km, 1200 images	approx. 1.6m	left and right, colour, 640×480px each
KITTI	Urban dataset containing mostly roads, houses, trees, and cars.	approx. 2.3km, 1400 images	approx. 1.6m	forward facing, greyscale, 1226×370px
Ruelle	Urban dataset of a narrow alleyway surrounded by houses. Several traverses are made with variations in the trajectory.	approx. 200m, 242 images	varied	forward facing, colour, 3264×2448px

Table 2: Overview of datasets used for testing.

paths), blurred images in some locations, and 'speed' variations in the case of the Begbroke Multi-speed sequence which uses a different framerate for each loop.

The City Centre and New College datasets originate in [Cummins and Newman, 2008], and consist of two different parts of a university campus, each with fairly varied terrain (roads, gardens, paths). These two datasets are challenging due to many dynamic elements such as cars and pedestrians. In addition, these datasets are non-ideal for the covisibility framework presented in this paper, because images were collected in an attempt to be independent from eachother, providing relatively little connectivity between frames and therefore proving to be even more challenging.

The KITTI dataset is provided by Geiger et al. [2013]. More specifically, it is the fifth sequence from the odometry benchmark sequences. This particular sequence was chosen because it includes both interesting loop-closures and accurate ground truth coordinates. Some challenges involved with this dataset are dynamic objects, speed variations, and some relatively short loop-closure sequences.

Finally, the Ruelle dataset is provided by a handheld point-and-shoot camera, with images from a narrow alleyway. The street which is traversed is relatively short, but is traversed several times from different view-points and different speeds (namely image spacing). The difficulties in this dataset lie in the fact that loop-closure sequences are inconsistent in length and order, as well as view-point.

The first five datasets from Table 2 are used to compare precision-recall characteristics for different place recognition frameworks in Section 4.2, while the last, shorter dataset is only used for illustrative purposes in Section 4.4.

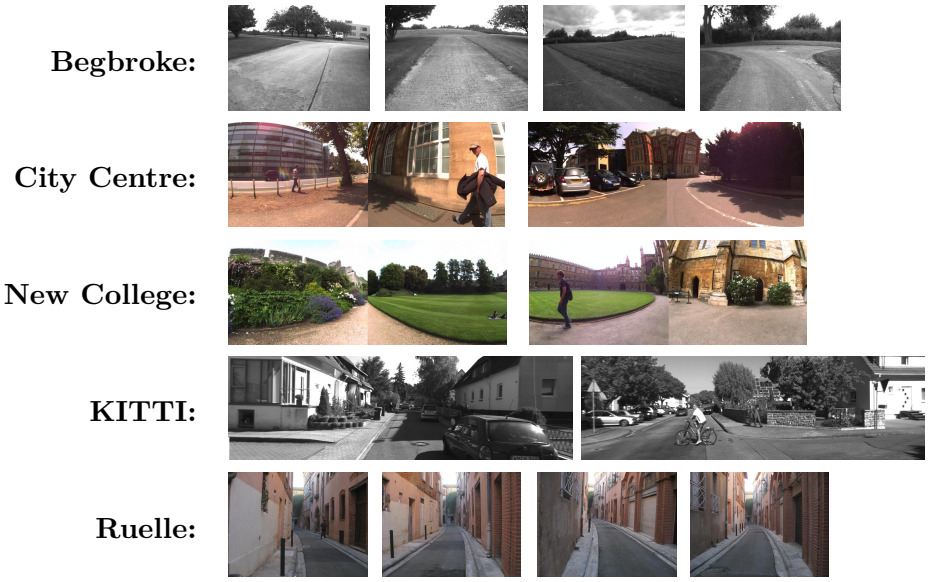


Figure 21: Example images from each of the datasets described in Table 2.

B Testing and Precision-Recall Metrics

Precision-recall characteristics are commonly used as a tool for system evaluation. Precision relates the number of correct matches to the number of false matches, whereas recall relates the number of correct matches to the number of missed matches. More formally, precision is the ratio of true positives over true positives plus false positives:

$$P = \frac{tp}{tp + fp}$$

And recall is the ratio of true positives over true positives plus false negatives:

$$R = \frac{tp}{tp + fn}$$

Curves can then be plotted, giving precision versus recall as the scores output by the algorithm are thresholded. A perfect system would return a result where both precision and recall have a value of one. When this is not achievable, the goal is to come as close to this as possible, possibly giving preference to maintaining certain levels of precision or recall, depending on the application.

For the tests in this work, a dataset of images is incrementally traversed, creating a map of locations over time and uses the most recent location as a query on the current map, with the goal of retrieving any previous instances of the location. A true positive is therefore defined as any returned location containing images that were taken within a certain radius of the query location. Similarly, a false positive is defined as a returned location which lies outside of the same radius, and a false negative is a location which lies within the radius but was not returned. When comparing different state-of-the-art methods, defining true and false matches in an unbiased way can be difficult. For example, defining all locations within the given radius as positive would be unfair towards algorithms such as FAB-MAP [Cummins and Newman, 2008] since it can only return one image per query. As a result, the test shown throughout this work use binary values of true positive, false positive, and false negative for each location. This additionally helps to avoid having to define the exact extent of a specific location which should be returned, as many cases can be ambiguous. The radius used for evaluation is set to 8 m unless otherwise stated. This choice is related to errors in the GPS ground truth data, which can reach up to several meters, and the fact that images can be spaced upwards of 2 m in some datasets.

References

Adrien Angeli, David Filliat, Stéphane Docieux, and Jean-Arcady Meyer. A fast and incremental method for loop-closure detection using bags of visual words. *IEEE Transactions on Robotics, Special Issue on Visual SLAM*, October 2008.

Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded Up Robust Features. *Computer Vision and Image Understanding (CVIU)*, 2008.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer Science, New York, NY, USA, 2007.

Tom Botterill, Steven Mills, and Richard Green. Bag-of-words-driven, single-camera simultaneous localization and mapping. *Journal of Field Robotics*, March/April 2011.

César Cadena, Dorian Gálvez-López, Juan D. Tardós, and José Neira. Robust place recognition with stereo sequences. *IEEE Transactions on Robotics*, August 2012.

Ondřej Chum, James Philbin, Josef Sivic, Michael Isard, and Andrew Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *IEEE International Conference on Computer Vision*, Rio de Janeiro, Brazil, 2007.

Mark Cummins. *Probabilistic localization and mapping in appearance space*. PhD thesis, University of Oxford, Balliol College, October 2009.

Mark Cummins and Paul Newman. FAB-MAP: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, June 2008.

Mark Cummins and Paul Newman. Appearance-only SLAM at large scale with FAB-MAP 2.0. *The International Journal of Robotics Research*, August 2011.

Susana Eyheramendy, David D. Lewis, and David Madigan. On the naive bayes model for text categorization, 2003.

Dorian Gálvez-López and Juan D. Tardós. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, October 2012.

Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research (IJRR)*, 2013.

Arren Glover, William Maddern, Michael Warren, Stephanie Reid, Michael Milford, and Gordon Wyeth. OpenFABMAP: An open source toolbox for appearance-based loop closure detection. In *IEEE International Conference on Robotics and Automation*, St. Paul, MN, USA, 2012.

Herve Jegou, Matthijs Douze, and Cordelia Schmid. On the burstiness of visual elements. In *IEEE Conference on Computer Vision and Pattern Recognition*, Miami Beach, United States, June 2009.

David G. Lowe. Object recognition from local scale-invariant features. *IEEE International Conference on Computer Vision*, 1999.

Will Maddern, Michael J. Milford, and Gordon F. Wyeth. CAT-SLAM: probabilistic localisation and mapping using a continuous appearance-based trajectory. *The International Journal of Robotics Research*, April 2012.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.

Christopher Mei, Gabe Sibley, and Paul Newman. Closing loops without places. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipai, Taiwan, 2010.

Michael J. Milford. Vision-based place recognition: how low can you go? *The International Journal of Robotics Research*, June 2013.

Oxford Mobile Robotics Group - OxfordMRG. Open source FabMap 2.0 code, 2013. URL `{http://www.robots.ox.ac.uk/~mjc/Software.htm}`. [Online; accessed 2013].

Ananth Ranganathan and Frank Dellaert. Bayesian surprise and landmark detection. In *IEEE International Conference on Robotics and Automation*, Kobe, Japan, 2009.

Satu Elisa Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, August 2007.

Karl-Michael Schneider. On word frequency information and negative evidence in naive Bayes text classification. In *Advances in Natural Language Processing*. Springer, 2004.

Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. *IEEE International Conference on Computer Vision*, 2003.

Elena Stumm, Christopher Mei, and Simon Lacroix. Probabilistic place recognition with covisibility maps. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Tokyo, Japan, 2013.

Niko Suenderhauf. OpenSeqSLAM code, 2013. URL `{https://openslam.org/openseqslam.html}`. [Online; accessed 2013].

Akihiko Torii, Josef Sivic, Tomas Pajdla, and Masatoshi Okutomi. Visual place recognition with repetitive structures. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, Portland, United States, June 2013.