Appearance-only SLAM at Large Scale with FAB-MAP 2.0

Mark Cummins and Paul Newman, Mobile Robotics Group, University of Oxford

***Abstract*— We describe a new formulation of appearance-only SLAM suitable for very large scale place recognition. The system navigates in the space of appearance, assigning each new observation to either a new or previously visited location, without reference to metric position. The system is demonstrated performing reliable online appearance mapping and loop closure detection over a 1,000 km trajectory, with mean filter update times of 14 ms. The scalability of the system is achieved by defining a sparse approximation to the FAB-MAP model suitable for implementation using an inverted index. Our formulation of the problem is fully probabilistic and naturally incorporates robustness against perceptual aliasing. We also demonstrate that the approach substantially outperforms the standard tf-idf ranking measure. The 1,000 km data set comprising almost a terabyte of omni-directional and stereo imagery is available for use, and we hope that it will serve as a benchmark for future systems.**

Fig. 1: Segments of the 1,000 km evaluation trajectory (ground truth).

I. Introduction

This paper is concerned with the problem of appearance-based place recognition at very large scales. We refer to the problem as "appearance-only SLAM" because we aim to address more than localization. Our approach can also determine when an observation comes from a location that has not previously been seen. Thus the system can incrementally construct a map, and so is a SLAM technique. However, our formulation of the problem is quite different to typical SLAM algorithms. We make no attempt to keep track of the vehicle or of landmarks in metric co-ordinates. Instead we parameterize the world as a set of discrete locations, and estimate their positions in an appearance space. Because distinctive places can be recognized even after unknown vehicle motion, appearance-only SLAM techniques provide a natural solution to the problems of loop-closure detection, multi-session mapping and kidnapped robot problems. The approach is thus complementary to metric SLAM methods that are typically challenged by these scenarios.

In prior work we have considered systems suitable for appearance-only SLAM at the scale of a few kilometers [12], and approximate inference techniques which extend applicability to a few tens of kilometers [11]. This paper builds on the probabilistic framework introduced in those papers, but modifies the structure of the model to support efficient inference over maps several orders of magnitude larger than previously considered. In seeking such a model, there are some compromises to be made between the fully Bayesian approach of our prior work, and a system which meets the efficiency needs of large scale applications. We describe a formulation which preserves almost all the key features of our earlier model, but allows for the exploitation of the sparsity of visual word data to achieve large reductions in computation and memory requirements. We validate the work on a 1,000 km data set; to date this is the largest experiment conducted with systems of this kind by a considerable margin. The data set, including omni-directional imagery, 20Hz stereo imagery and 5Hz GPS, is available for use by other researchers and is intended to serve as a benchmark for future systems. The paper concludes with an extensive performance evaluation for the new system, including an analysis of a modified visual vocabulary learning stage which is shown to increase performance, and a comparison to the commonly used tf-idf ranking measure, which is considerably out-performed by our new approach. The material was first presented in [13]; here we expand the presentation with a more detailed treatment and additional results.

II. Related Work

While appearance-based navigation has a long history within robotics [17], [40], there has been considerable development in the field in the last five years. Appearance-based navigation and loop closure detection systems operating on trajectories on the order of a few kilometers in length are now commonplace [1], [20], [7], [42], [25]. Indeed, place recognition systems similar in character to the one

described here are now used even in single-camera SLAM systems designed for small-scale applications [18], [41].

Use of these systems on the scale of tens of kilometers or more has also begun to be feasible. In the largest appearance-based navigation experiment we are aware of [28], a set of biologically inspired approaches is employed. The system achieved successful loop closure detection and mapping in a collection of more than 12,000 images from a 66 km trajectory, with processing time of less than 100 ms per image. The appearance-recognition component of the system is based on direct template matching, so scales linearly with the size of the environment. Operating at a similar scale, Bosse and Zlot describe a place recognition system based on distinctive keypoints extracted from 2D lidar data [4], and demonstrate good precision-recall performance over an 18 km suburban data set. Related results, though based on a less scalable correlation-based submap matching method, were also described in [5].

Another recent research direction is the development of integrated systems which combine appearance and metric information. Olson described an approach to increasing the robustness of general loop closure detection systems by using both appearance and relative metric information to select a single consistent set of loop closures from a larger number of candidates [32]. The method was evaluated over several kilometers of urban data and shown to recover high-precision loop closures even with the use of artificially poor image features. Blanco *et al.* described a system where metric and topological position information is considered jointly in the estimator [3]. More loosely coupled systems were also described in [24], [30].

Considerable relevant work also exists on the more restricted problem of global localization. For example, Schindler *et al.* describe a city-scale location recognition system [37] based on the vocabulary tree approach of [31]. The system was demonstrated on a 30,000 image data set from 20 km of urban streets, with retrieval times below 200 ms. Also of direct relevance is the research on content-based image retrieval systems in the computer vision community, where systems have been described that deal with more than a million images [34], [9], [31], [21] . However, the problem of retrieval from a fixed index is considerably easier than the full loop-closure problem, because it is possible to tune the system directly on the images to be recognized, and the difficult issue of new place detection does not arise. We believe the results presented in this paper represent the largest scale system that fully addresses these issues of incrementality and perceptual aliasing.

III. Probabilistic Model

The probabilistic model employed in this paper is based directly on the scheme outlined in [12], [14]. For completeness, we recap it briefly here.

The basic data representation used is the bag-of-words approach developed in the computer vision community [39]. Features are detected in raw sensory data, and these features are then quantized with respect to a *vocabulary*, yielding *visual words*. The vocabulary is learned by clustering all feature vectors from a set of training data. The Voronoi regions of the cluster centres then define the set of feature vectors that correspond to a particular visual word. The continuous space of feature vectors is thus mapped into the discrete space of visual words, which enables the use of efficient inference and retrieval techniques. In this paper, the raw sensor data of interest is imagery, processed with the SURF feature detector [2], though in principle the approach is applicable to any sensor or combination of sensors, and we have explored multi-sensory applications elsewhere [35].

FAB-MAP, our appearance-only SLAM system, defines a probabilistic model over the bag-of-words representation. An observation of local scene appearance captured at time k is denoted $Z_k = \{z_1, \ldots, z_{|v|}\}$, where $|v|$ is the number of words in the visual vocabulary. The binary variable z_q, which we refer to as an observation component, takes value 1 when the q^{th} word of the vocabulary is present in the observation. $\mathcal{Z}^k$ is used to denote the set of all observations up to time k.

At time k, our map of the environment is a collection of n_k discrete and disjoint locations $\mathcal{L}^k = \{L_1, \ldots, L_{n_k}\}$. Each of these locations has an associated appearance model, which we parameterize in terms of unobservable "scene elements", e_q. A detector yields visual word observations z_q, which are noisy measurements of the existence of the underlying scene element e_q. The appearance model of a location in the map is our belief about the existence of each scene element at that location:

$$L_i : \{p(e_1 = 1 \mid L_i), \ldots, p(e_{|v|} = 1 \mid L_i)\} \tag{1}$$

where each of the scene elements e_q are generated independently by the location. A detector model relates scene elements e_q to feature detection z_q. The detector is specified by

$$\mathcal{D} : \quad \begin{cases} p(z_q = 1 \mid e_q = 0), & \text{false positive probability.} \\ p(z_q = 0 \mid e_q = 1), & \text{false negative probability.} \end{cases} \tag{2}$$

A further salient aspect of the data is that visual words do not occur independently – indeed, word occurrence tends to be highly correlated. For example, words associated with car wheels and car doors are likely to be observed simultaneously. We capture these dependencies by learning a tree-structured Bayesian network using the Chow Liu algorithm [8], which yields the optimal approximation to the joint distribution over word occurrence within the space of tree-structured networks. Importantly, tree-structured networks also permit efficient learning and inference even for very large visual vocabulary sizes. The graphical model of the system is shown in Figure 2.

Given our probabilistic appearance model, localization and mapping can be cast as a recursive Bayes estimation problem, closely analogous to metric SLAM. A pdf over location given the set of observations up to time k is given by:

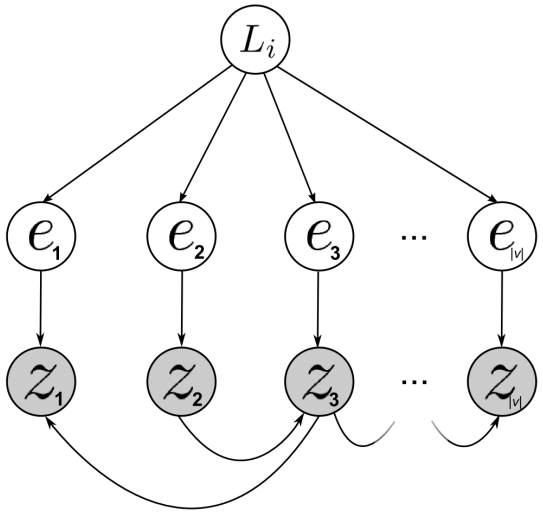


Fig. 2: Graphical model of the system. Locations L_i independently generate scene elements e_q. Visual word detections z_q are conditioned on scene elements e_q via the detector model, and on each other via the Chow Liu tree.

$$p(L_i \mid \mathcal{Z}^k) = \frac{p(Z_k \mid L_i, \mathcal{Z}^{k-1}) p(L_i \mid \mathcal{Z}^{k-1})}{p(Z_k \mid \mathcal{Z}^{k-1})} \tag{3}$$

Here $p(L_i \mid \mathcal{Z}^{k-1})$ is our prior belief about our location, $p(Z_k \mid L_i, \mathcal{Z}^{k-1})$ is the observation likelihood, and $p(Z_k \mid \mathcal{Z}^{k-1})$ is a normalizing term. We briefly discuss the evaluation of each of these terms below. For a more detailed treatment we refer readers to [12], [14].

Observation Likelihood: To evaluate the observation likelihood, we assume independence between the current and past observations conditioned on the location, and make use of the Chow Liu model of the joint distribution, yielding:

$$p(Z_k \mid L_i) = p(z_r \mid L_i) \prod_{q=2}^{|v|} p(z_q \mid z_{p_q}, L_i) \tag{4}$$

where z_r is the root of the Chow Liu tree and z_{p_q} is the parent of z_q in the tree. After further manipulation (see [12]), each term in the product can be further expanded as:

$$p(z_q \mid z_{p_q}, L_i) = \sum_{s_{e_q} \in \{0,1\}} p(z_q \mid e_q = s_{e_q}, z_{p_q}) p(e_q = s_{e_q} \mid L_i) \tag{5}$$

which can be evaluated explicitly.

In some configurations of the system we find that these likelihoods can be too peaked, so we introduce an optional smoothing step:

$$p(Z_k \mid L_i) \longrightarrow \sigma p(Z_k \mid L_i) + \frac{(1-\sigma)}{n_k} \tag{6}$$

where n_k is the number of places in the map and σ is the smoothing parameter, which we typically set to be slightly less than 1. This smoothing is helpful because our model inevitably captures only some of the dependencies between visual words – as some dependencies are not captured, individual visual words seem more informative than they actually are, and so loop closure probabilities have a tendency to be over-confident. See [14] for further discussion.

Location Prior: The location prior $p(L_i \mid \mathcal{Z}^{k-1})$ is obtained by transforming the previous position estimate via a simple motion model. The model assumes that if the vehicle is at location i at time $k-1$, it is likely to be at one of the topologically adjacent locations at time k.

Normalization: In contrast to a localization system, a SLAM system requires an explicit evaluation of the normalizing term $p(Z_k \mid \mathcal{Z}^{k-1})$. The normalizing term converts the appearance likelihood into a probability of loop closure, by accounting for the possibility that the current observation comes from a location not currently in the robot's map. Intuitively $p(Z_k \mid \mathcal{Z}^{k-1})$ is a measure of the distinctiveness of an observation, and thus directly related to the problem of perceptual aliasing.

To calculate $p(Z_k \mid \mathcal{Z}^{k-1})$, we divide the world into the set of places in our current map, $\mathcal{L}^k$, and the set of unmapped places $\overline{\mathcal{L}^k}$, so that

$$\begin{aligned} p(Z_k \mid \mathcal{Z}^{k-1}) &= \sum_{m \in \mathcal{L}^k} p(Z_k \mid L_m) p(L_m \mid \mathcal{Z}^{k-1}) \quad (7) \\ &\quad + \sum_{u \in \overline{\mathcal{L}^k}} p(Z_k \mid L_u) p(L_u \mid \mathcal{Z}^{k-1}) \quad (8) \end{aligned}$$

The second summation cannot be evaluated directly because it involves all possible unknown locations. However, if we have a large set of randomly collected location models L_u, (readily available from previous runs of the robot or other suitable data sources such as, for our application, Google Street View), we can approximate the summation by Monte Carlo sampling. Assuming a uniform prior over the samples, this yields:

$$\begin{aligned} p(Z_k \mid \mathcal{Z}^{k-1}) &\approx \sum_{m \in \mathcal{L}^k} p(Z_k \mid L_m) p(L_m \mid \mathcal{Z}^{k-1}) \quad (9) \\ &\quad + p(L_{new} \mid \mathcal{Z}^{k-1}) \sum_{u=1}^{n_s} \frac{p(Z_k \mid L_u)}{n_s} \quad (10) \end{aligned}$$

where n_s is the number of samples used, and $p(L_{new} \mid \mathcal{Z}^{k-1})$ is our prior probability of being at a new location.

Data Association: Once the pdf over locations is computed, a data association decision is made. The observation Z_k is used either to initialize a new location, or update the appearance model of an existing location. Recall that an appearance model consists of a set of beliefs about the existence of scene elements at the location, $\{p(e_1 = 1 \mid L_i), \ldots, p(e_{|v|} = 1 \mid L_i)\}$. Each component of the appearance model can be updated according to:

$$p(e_i = 1 \mid L_j, \mathcal{Z}^k) = \frac{p(Z_k \mid e_i = 1, L_j) p(e_i = 1 \mid L_j, \mathcal{Z}^{k-1})}{p(Z_k \mid L_j)} \tag{11}$$

In the case of new locations, the values $p(e_i = 1 \mid L)$ are first initialized to the marginal probability $p(e_i = 1)$ derived from training data, and then the update is applied.

IV. A Model for Scalable Navigation

We now discuss the development of a system suitable for appearance-based navigation in environments where the map may contain hundreds of thousands of places or more. The probabilistic model employed in the new system builds directly on the one outlined in Section III. For a highly scalable system, we modify the model so that it is suitable for implementation using an inverted index architecture. We begin by introducing the inverted index.

A. Inverted Indices

An inverted index is a simple data structure used throughout information retrieval, which enables efficient search of large document collections [26]. If a document is considered as a list of word identifiers, then an inverted index maintains the inverse mapping from words to documents. That is, for each word in the vocabulary, a list of the documents in which that word appears is maintained. Finding all documents that contain a word or set of words is then a very cheap operation. In computational terms, the inverted index approach scales to document collections that are arbitrarily large [6].

B. FAB-MAP 2.0 - An Approximation to the FAB-MAP Model

We would like to find a probabilistic model that can take advantage of the scalability of inverted index techniques. Our FAB-MAP model is not directly implementable using an inverted index, because the appearance likelihood $p(Z_k \mid L_i)$ requires evaluation of Equation 4, $\prod_{q=2}^{|v|} p(z_q \mid z_{p_q}, L_i)$. Every observation component contributes to the appearance likelihood, including *negative* observations – those where $z_q = 0$ (words not detected in the current image). As such, it does not have the sparsity structure that enables inverted index approaches to scale. The computation pattern is illustrated in Figure 3. Perhaps surprisingly, we have found that simply ignoring the negative observations has a detrimental impact on place recognition performance. Thus we seek a formulation that will enable efficient implementation, but preserve the information inherent in the negative observations.

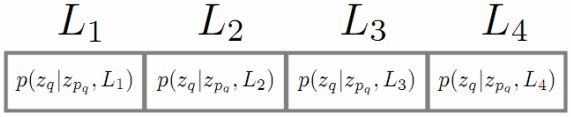
To enable an inverted index implementation, we modify the probabilistic model in two ways. Firstly, we place some restrictions on the probabilities in the location models. Recalling Equation 1, location models are parametrized as $\{p(e_1 = 1 \mid L_i), \ldots, p(e_{|v|} = 1 \mid L_i)\}$, that is, by a set of beliefs about the existence of scene elements that give rise to observations of the words in the vocabulary. Let $p(e_q \mid L_i)|_{\{0\}}$ denote one of these beliefs, where the subscript $\{0\}$ indicates the history of observations that have been associated with the location. Thus $\{0\}$ denotes one associated observation with $z_q = 0$, and $\{0, 0, 1\}$ denotes three associated observations, with $z_q = 1$ in one of those observations. Further, let $p(e_q \mid L_i)|_0$ indicate that in all observations associated with the location, $z_q = 0$.

In the FAB-MAP 1.0 model described in Section III, $p(e_q \mid L_i)|_0$ can take on a range of values – for example, $p(e_q \mid L_i)|_{\{0\}} \neq p(e_q \mid L_i)|_{\{0,0\}}$, as the belief in the non-existence of the scene elements increases as more supporting observations become available. While this is in some sense the correct model, a consequence is that the appearance likelihood due to a negative observation is just as expensive to calculate as that due a positive observation. Negative observations greatly outnumber positive ones, and are also generally less informative. In FAB-MAP 2.0, we restrict the model in such a way that the negative observations can be evaluated much more efficiently, at the cost of a partial loss of information content. The model is a compromise between a correct Bayesian approach as in FAB-MAP 1.0, and a system suitable for large scale applications. Concretely, in FAB-MAP 2.0, the model is restricted so that $p(e_q \mid L_i)|_0$ must take the same value for all locations; it is clamped at the value $p(e_q \mid L_i)|_{\{0\}}$. This restriction enables an efficient likelihood calculation, illustrated in Figure 4. Note that when a location in the map has been observed only once, this new model is identical to FAB-MAP 1.0. It is only when we have multiple observations of a location that some descriptive power is lost (because terms of the form $p(e_q \mid L_i)|_{\{0,0\}}$ remain clamped at the value $p(e_q \mid L_i)|_{\{0\}}$) . However, the effect of this change is negligible in practice because the location model built from a single observation is typically already sufficient to enable the location to be recognized.

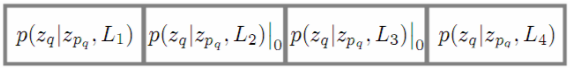
To understand why the restricted model enables an efficient implementation, consider the calculation of one term of the observation likelihood, as per Equation 5, across all locations in the map. That is, we wish to compute the term $p(z_q \mid z_{p_q}, L_i)$ for some visual word q, for all L_i in the map. Recalling Section III, the term is given by

$$p(z_q \mid z_{p_q}, L_i) = \sum_{s_{e_q} \in \{0,1\}} p(z_q \mid e_q = s_{e_q}, z_{p_q}) p(e_q = s_{e_q} \mid L_i)$$

In the unrestricted model, this will involve computing one term for each location, as illustrated in Figure 4a.



In the restricted model, Figure 4b, the term takes a single common value for all locations where word q was not previous observed (in those locations, the word exists with probability $p(e_q \mid L_i)|_0$, which determines the value of Equation 5. We denote this value by $p(z_q \mid z_{p_q}, L)\big|_0$) .



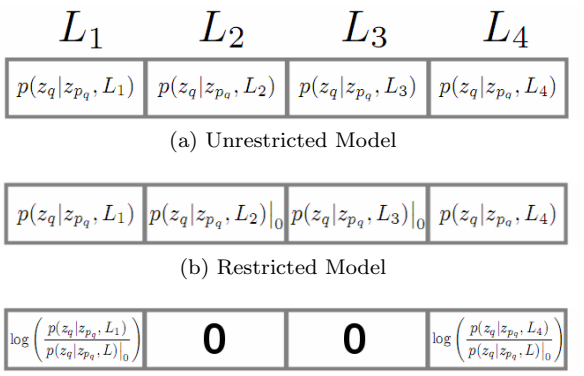
Working with log-likelihoods, and given that the distribution will later be normalized, the calculation can be reorganized so that it has a sparse structure (Figure 4c).

(a) FAB-MAP 1.0, [12].

(b) FAB-MAP 1.0 with bail out strategy, [11].

(c) FAB-MAP 2.0 (this paper)

Fig. 3: Illustration of the amount of computation performed by the different FAB-MAP models. The shaded region of each block represents the appearance likelihood terms $p(z_q \mid z_{p_q}, L_i)$ which must be evaluated to compute $p(Z_k \mid L_i)$. In (a), FAB-MAP 1.0, each time a new observation is collected the likelihood must be computed for all words in all locations in the map. In (b), the pattern is shown from the approximate inference procedure defined in [11], where a "bail out" strategy discards locations during the course of the computation. Many locations are quickly excluded, so the number of appearance likelihood terms which must be calculated is greatly reduced. In (c), the fully sparse evaluation in FAB-MAP 2.0 is shown, which further reduces computation requirements.



(a) Unrestricted Model

(b) Restricted Model

(c) Sparse likelihood update in the restricted model.

Fig. 4: Illustration of the calculation of one term of Equation 4, i.e. the observation likelihood due to a particular word in the vocabulary. This example shows a map with four locations. The total observation likelihood for a given location, $p(Z \mid L)$, is calculated by evaluating the illustrated terms for all words in the vocabulary. In (a), the model is unrestricted, and the likelihood term can take a different value for each location. In (b), the restricted model, the likelihood term in all locations where the currently considered word was not previously observed is constrained to take the same value. The calculation can now be organized so that it has a sparse structure, (c).

This allows for efficient implementation using an inverted index. The terms shown in Figure 4c can be thought of as the weights of the votes a word casts for a location.

We emphasize the fact that the restriction we have placed on the model is slight, and most of the power of the original model is retained. During the exploration phase, when only one observation is associated with each location, the two schemes are identical[1]. The "fixed" terms $p(e_q \mid L_i)|_0$ can (and do) vary with q (word ID), and in principle also with time. Treatment of correlations between words, of perceptual aliasing, and of the detector model remains unaffected.

The second change we make to the model concerns data association. Previously, data association was carried out via Equation 11, updating the beliefs $p(e_q \mid L_i)$. Effectively this amounts to capturing the average appearance of a location. For example, if a location has a multi-modal distribution over word occurrence, such as a door that may be either open or shut, then the location appearance model will approach the mean of this distribution. In FAB-MAP 1.0, when computation increased swiftly with the number of appearance models to be evaluated, this was a reasonable design choice. For FAB-MAP 2.0 we switch to representing locations in a sample-based fashion, which better handles these multi-modal appearance effects. Locations now consist of a set of appearance models as defined in Equation 1, with each new observation associated with the location defining a new such model.

Concretely, a location L_i now consists of a set of samples $\{l_1, l_2, ..., l_{\eta_k}\}_i$ where η_k is the number of samples associated with the location at time k. Each sample l is a FAB-MAP 1.0 "location model" as defined in Section III. Previously, each location L_i consisted of a single one of these models whose mean appearance was updated via Equation 11. In the sample based representation we simply associate multiple such models with a location, one for each observation collected from the location. The "samples" are initialized via Equation 11, but never subsequently updated. When evaluating the observation likelihood during inference, we compute the expectation over the samples:

$$p(Z_k \mid L_i) = \frac{1}{\eta_k} \sum_{r=1}^{\eta_k} p(Z_k \mid l_r \in L_i) \tag{12}$$

This change to a sample-based representation of the

[1] Assuming the detector model does not change with time.

locations is actually required because of the restrictions placed on scene element existence probabilities in our new architecture, but we expect it also to be largely beneficial. In addition to improving our ability to deal with multi-modal location appearance, it also makes data association a reversible operation, and would make the implementation of a scheme that maintains multiple hypotheses over data association decisions (e.g. [36]) very simple and efficient. While it means that inference time now increases with every observation collected, the system is sufficiently scalable that this is not of immediate relevance, and the greater ability to deal with variable location appearance is preferred.

C. Implementation

Pseudocode for the main likelihood calculation required in FAB-MAP is given in Algorithm 1. The complexity of this implementation is $O(\#vocab)$, the number of words in the visual vocabulary. This straight-forward implementation is in fact fast enough for practical use, however it can be improved by a caching scheme which yields an algorithm with complexity effectively independent of vocabulary size. The key observation is that for a given word and a given location, the log-likelihood increments (or "votes") to be calculated in Algorithm 1, $\log(\frac{p(z_q=s_q|z_{p_q}=s_{p_q},L_i)}{p(z_q=s_q|z_{p_q}=s_{p_q},L)|_0})$, have four possible states:

$$\begin{aligned} &\text{Case 1}: && (s_q=1, s_{p_q}=1) \\ &\text{Case 2}: && (s_q=1, s_{p_q}=0) \\ &\text{Case 3}: && (s_q=0, s_{p_q}=1) \\ &\text{Case 4}: && (s_q=0, s_{p_q}=0) \end{aligned} \tag{13}$$

which depend on whether or not the word q and its parent word p_q are present in the current observation Z_k. As observations are typically sparse, case four, $(s_q=0, s_{p_q}=0)$, will be by far the most common. We can exploit this fact by pre-calculating the likelihood of a location L_i as if all votes were from case four. This likelihood is calculated once, when the location is added to the map, and cached. We refer to this as the location's "default likelihood". When processing a new observation, we need only adjust the default likelihood of a location to take account of those words which actually lie in cases one to three for the current observation. Calculating this adjustment involves only those words that are present in the current observation (cases 1,2), or that are children of these words in the Chow Liu tree (case 3). The number of words present in a given observation is typically a small constant independent of vocabulary size. The number of words that relate to case three depends on the structure of the Chow Liu tree, and in pathological cases could still be $O(\#vocab)$. However, in practice we observe it to be a small multiple of the number of observed words. Pseudocode for the algorithm which exploits this sparsity is given in Algorithm 2. In our experiments using a 100,000 word vocabulary, we observed an order of magnitude speed increase with this approach.

V. Maintaining System Performance at Scale

This section discusses some issues relevant to maintaining system performance when the map is very large. A geometric verification stage is introduced which we found to be almost essential in preserving precision on our largest data sets. We also discuss scalable approaches to visual vocabulary and Chow Liu tree learning.

A. Geometric Verification

While a navigation system based entirely on the bag-of-words likelihood is possible (e.g. [12]), we have found in common with others [34] that a post-verification stage, which checks that the matched images satisfy geometric constraints, considerably improves performance. The impact is particularly noticeable as data set size increases - it is helpful on our 70 km data set but almost essential on the 1,000 km set.

We apply the geometric verification to a "shortlist" of the 100 most likely locations (those which maximize $p(Z_k \mid L_i, \mathcal{Z}^{k-1})p(L_i \mid \mathcal{Z}^{k-1})$) and to the 100 most likely samples (the location models used to evaluate the normalizing term $p(Z_k \mid \mathcal{Z}^{k-1})$). For each of these locations we check geometric consistency with the current observation using RANSAC [19]. Candidate interest point correspondences are derived from the bag-of-words assignment already computed. Because our aim is only to verify approximate geometric consistency rather than recover exact pose to pose transformations, we assume a highly simplified model where the transformation between poses is constrained to be a pure rotation about the vertical axis. A single point correspondence then defines a transformation. Due to this simplified model, and also because our point correspondences typically have few outliers, the geometric verification is very rapid. Only a few RANSAC iterations are required – we assume 65% inliers and so only 13 RANSAC iterations are needed to recover a model with an expected 10^{-6} error rate. The pure-rotation model is a gross approximation, but given the constrained motion of our vehicle mounted camera, it is good enough to give a substantial boost to recognition performance, while imposing very little computational overhead. We accommodate some translation between poses by allowing large inlier regions for point correspondences (up to 50 pixels in x and y, and a factor of 4 in scale). Typical "consistent" correspondences are shown in Figure 5. Having recovered a set of inliers using RANSAC we recompute the location's likelihood by setting $z_q = 0$ for all those visual words not part of the inlier set. A likelihood of zero is assigned to all locations not subject to geometric verification. For the 1,000 km experiment, the mean time taken to geometrically verify and re-rank all 200 shortlisted locations was only 10 ms and the maximum time was 145 ms.

The post-verification step considerably boosts recognition performance, however as a method of incorporating geometric information it is not entirely satisfying. An interesting alternative to post-verification would be to build the geometric information directly into the core

Algorithm 1 Calculation of $p(Z_k \mid L_i)$ using the inverted index.

```
for q in vocabulary do:
  //Get all locations where word q
  //was observed
  locations = inverted_index[q]
  for L_i in locations do:
     //Update the log-likelihood
     //of each of these locations
     loglikelihood[L_i] += log( p(z_q=s_q | z_pq=s_pq, L_i) / p(z_q=s_q | z_pq=s_pq, L) |_0 )
```

Algorithm 2 Log-likelihood update using the inverted index and exploiting observation sparsity.

```
Update, Part A (Default Likelihood):
//Each location's likelihood is initialized
//to appropriate "default likelihood"
//which assumes an "null" observation with z_q = 0, ∀q
//This can be thought of as the sum of "default votes" D_q
//for each observed word at the location
//D_q = log( p(z_q=0 | z_pq=0, L_i) / p(z_q=0 | z_pq=0, L) |_0 )
//Note that the default likelihood will be
//different for each location.
Initialize_Locations_To_Default_Likelihood()
Update, Part B (Observations such that z_q = 1):
//Now, adjust the votes based on the content of the current observation.
for z_q in Z, such that z_q = 1 do:
  //Get all locations where word q
  //was observed
  locations = inverted_index[q]
  for L_i in locations do:
     //Update the log-likelihood
     //of each of these locations
     //by removing the default vote D_q
     //and adding the appropriate vote.
     loglikelihood[L_i] += log( p(z_q=1 | z_pq=s_pq, L_i) / p(z_q=1 | z_pq=s_pq, L) |_0 ) - D_q
Update, Part C (Observations such that z_q = 0 and z_pq = 1):
//Same as Part B, but for unobserved words that are
//children of observed words in the CL tree.
for z_q in Z, such that z_q = 0 and z_pq = 1 do:
  locations = inverted_index[q]
  for L_i in locations do:
     loglikelihood[L_i] += log( p(z_q=0 | z_pq=1, L_i) / p(z_q=0 | z_pq=1, L) |_0 ) - D_q
```

probabilistic model which ranks locations. Some related work by colleagues in our lab has recently explored this approach [33], though not yet in a formulation which can be evaluated rapidly enough for the scales considered in this paper.

B. Visual Vocabulary Learning At Large Scale

Clustering: A number of challenges arise in learning visual vocabularies at large scale. The number of SURF features extracted from training images is typically very large; our relatively small training set of 1,921 images produces 2.5 million 128-dimensional SURF descriptors occupying 3.2 GB. Even the most scalable clustering algorithms such as k-means are too slow to be practical. Instead we apply the fast approximate k-means algorithm discussed in [34], where, at the beginning of each k-means iteration, a randomized forest of kd-trees [38], [29] is constructed over the cluster centres, which is then used for fast (approximate) distance calculations. This procedure has been shown to outperform alternatives such as hierarchical k-means [31] in terms of visual vocabulary retrieval performance.

As k-means clustering typically converges only to a

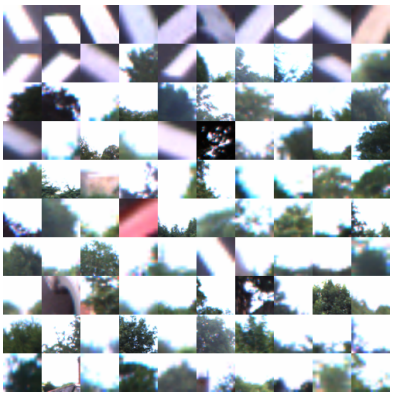
Fig. 5: A example of geometric verification, showing inliers identified by RANSAC. Note that we are not recovering an exact pose-to-pose transformation; the correspondences are only approximately geometrically consistent.

local minimum of its error metric, the quality of the visual vocabulary is sensitive to the initial cluster locations supplied to k-means. Nevertheless, random initial locations are commonly used. We have found that this leads to poor visual vocabularies, because there are very large density variations in the feature space. In these conditions, randomly chosen cluster centres tend to lie largely within the densest region of the feature space, and the final clustering over-segments the dense region, with poor clustering elsewhere. For example, in our vehicle-collected data, huge numbers of very similar features are generated by road markings, whereas rarer objects (more useful for place recognition) may only have a few instances in the training set. Randomly initialized k-means yields a visual vocabulary where a large fraction of the words correspond to road markings, with tiny variations between words. Similar effects were observed by Jurie and Triggs [23]. Examples are shown in Figure 6.

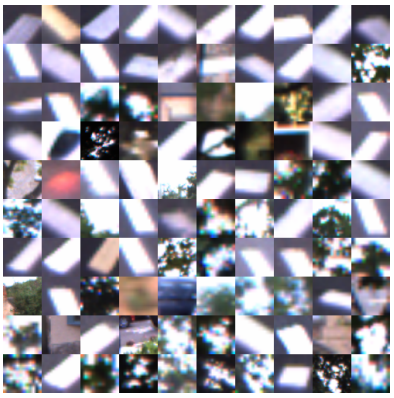
To avoid these effects, we choose the initial cluster centres for k-means using a fixed-radius incremental pre-clustering, where the data points are inspected sequentially, and a new cluster centre is initialized for every data point that lies further than a fixed threshold from all existing clusters. This is similar to the furthest-first initialization technique [15], but more computationally tractable for large data sets. We also modify k-means by adding a cluster merging heuristic. After each k-means iteration, if any two cluster centres are closer than a fixed threshold, one of the two cluster centres is reinitialized to a random location.

The modified clustering gives a robust boost to system performance (Figure 7). We have observed the effect on multiple data sets and under various different system configurations (with and without geometric verification, etc.). Curiously, however, we do not see the effect when using tf-idf ranking. We have no intuitive explanation for why tf-idf does not benefit in a similar way to FAB-MAP.

Chow Liu Tree Learning: Chow Liu tree learning is also challenging at large scale. The standard algorithm for learning the Chow Liu tree involves computing a (temporary) mutual information graph of size $|v|^2$, so the computation time is quadratic in the vocabulary size. For the 100,000 word vocabulary discussed in Section VII, the relevant graph would require 80 GB of storage. Happily,



(a) K-means with radius-based initialization and merging step.



(b) K-means with random initialization.

Fig. 6: The 100 most common visual words in the vocabularies used for the car-based experiments, showing one exemplar per word. With random initialization (Subfigure (b)) k-means tends to over-segment the densest regions of feature space, leading to a visual vocabulary with many highly similar visual words (in this case, many words corresponding to near-identical views of road markings). Using radius-based initialization and cluster merging (Subfigure (a)) produces a visual vocabulary with words that are better separated in feature space.

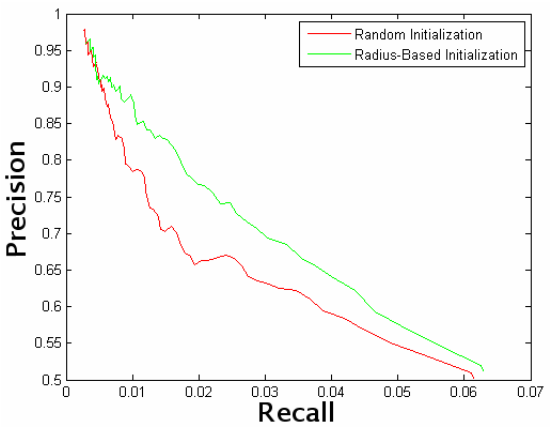


Fig. 7: Precision-recall curves showing the effect of k-means initialization on overall performance. Performance shown is for a subset of the 1,000 km data set, ranked according to the baseline FAB-MAP 2.0 model (i.e. not including motion model or geometric check). Detector terms were tuned separately for each vocabulary, so as to make the comparison fair. The performance difference persists after geometric re-ranking is added to the model. Curiously, we do not observe the effect when using tf-idf ranking.

there is an efficient algorithm for learning Chow Liu trees when the data of interest is sparse [27]. Meilă's algorithm has complexity $O(s^2 \log s)$, where s is a sparsity measure, equal to the maximum number of visual words present in any training image. Visual word data is typically very sparse, with only a small fraction of the vocabulary present in any given image. This allows efficient Chow Liu tree learning even for large vocabulary sizes. For example, the tree of the 100,000 word vocabulary used in Section VII was learned in 31 minutes on a 3GHZ Pentium IV.

For both the clustering and Chow Liu learning, we use external memory techniques to deal with the large quantities of data involved [16].

VI. Data Set

For a truly large scale evaluation of the system, the experiments in this paper make use of a 1,000 km data set. The data was collected by a car-mounted sensor array (see Figure 8), and consists of omni-directional imagery from a Point Grey Ladybug2, 20Hz stereo imagery from a Point Grey Bumblebee[2], and 5Hz GPS data. Omni-directional image capture was triggered every 4 meters on the basis of GPS. The omni-directional images were captured at 1920x512 resolution, and the stereo images at 512x384.

The data set was collected over six days in December, with a total length of slightly less than 21 hours, and includes a mixture of urban, rural and motorway environments. The total set comprises 803 GB of imagery (including stereo) and 177 GB of extracted features. There are 103,256 omni-directional images, of which 49,493 are loop closures. The median distance between image captures

[2]Not used in these results.

Fig. 8: Vehicle and sensor rig used to capture the 70 km and 1,000 km data sets. The rig consists of a Ladybug2 omni-directional camera and Bumblebee stereo camera, both from Point Grey Research. The cameras were mounted approximately three meters above the road surface. GPS data was collected with a Seres unit from CSI Wireless, mounted on the roof of the car.

is 8.7 m – this is larger than the targeted 4 m because the Ladybug2 could not provide the necessary frame rate during faster portions of the route. The median time between image captures is 0.48 seconds, which provides our benchmark for real-time image retrieval performance.

Two supplemental data sets were also collected. A set of 1,921 omni-directional images collected 30 m apart was used to train the visual vocabulary and Chow Liu tree, and also served as the sampling set for the Monte Carlo integration required in Equation 9. The area where this training set was collected did not overlap with that of the test data sets. A second data set of 70 km was also collected in August, four months prior to the main 1,000 km data set. This serves as a smaller-scale test of the system. The data sets are summarized in Table I.

The 1,000 km data set, collected in mid-December, provides an extremely challenging benchmark for place recognition systems. Due to the time of year, the sun was low on the horizon, so that scenes typically have high dynamic range and quickly varying lighting conditions. We developed custom auto-exposure controllers for the cameras that largely ensured good image quality, however, there is unavoidable information loss in such conditions. Additionally, large sections of the route feature self-similar

motorway environments, which provide a challenging test of the system's ability to deal with perceptual aliasing. The smaller data set collected during August features more benign imaging conditions and will demonstrate the performance that can be typically expected from the system.

Finally, collecting a data set of this magnitude highlights some practical challenges for any truly robust field robotics deployment. We encountered significant difficulty in keeping the camera lenses clean – in winter from accumulating moisture and particulate matter, in summer from fly impacts. For this experiment we periodically cleaned the cameras manually – a more robust solution seems a worthy research topic.

The 70 km data set is available at `http://www.robots.ox.ac.uk/~mobile/EynshamDataset.html` (Extension 2), and the 1,000 km dataset (Extension 3) is available upon request.

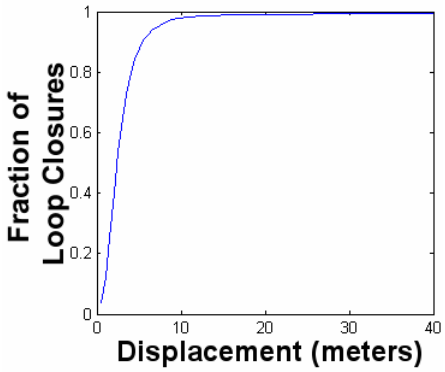


Fig. 9: The typical displacement in meters between two images identified as loop closures. While any pair separated by less than 40 m is accepted as a true positive, because the ground truth separation can occasionally be this large, 89% of detected loop closures are separated by less than 5m, and 98% by less than 10 m.

VII. Results

We now present the system performance evaluation. Overall performance is outlined, the impact of the Chow Liu tree is examined, and the system is benchmarked against the common tf-idf weighting function.

A. Test Conditions

The system was tested on the two data sets, respectively 70 km and 1,000 km. As input to the system, we used 128D non-rotationally invariant SURF descriptors. These features were quantized to visual words using a randomized forest of eight kd-trees. The visual vocabulary and Chow Liu tree were trained using the system described in Section V-B and the 1,921 image training set described in Section VI. In order to ensure an unbiased Chow Liu tree, the images in the training set were collected 30 m apart, so that as far as possible they do not overlap in viewpoint, and thus approximate independent samples from the distribution over images.

We investigate two different visual vocabularies, of 10,000 and 100,000 words respectively. The detector model (Equation 2), the main user-configurable parameter of our system, was determined by a grid search to maximize performance on a set of training loop closures. The detector model primarily captures the effects of variability in SURF interest point detection and feature quantization error. For the 10,000 word vocabulary we set $p(z = 1 \mid e = 1) = 0.39$ and $p(z = 1 \mid e = 0) = 0.005$. For the 100,000 word vocabulary, the values were $p(z = 1 \mid e = 1) = 0.2$ and $p(z = 1 \mid e = 0) = 0.005$. The likelihood smoothing term σ introduced in Section III was set to 0.99, except in the case where the geometric check was used, where we found it to be unnecessary. This means that when the geometric check was applied, the system could accept a loop closure on the basis of a single image. Finally, we also investigate the importance of learning the Chow Liu tree by comparing against a Naive Bayes formulation which neglects the correlations between words. We refer to these different system configurations as "100k, CL" and "100k, NB", and similarly for the 10k word vocabulary.

Performance of the system was measured against ground truth loop closures determined from the GPS data. GPS errors and dropouts were corrected manually. Any pair of matched images that were separated by less than 40 m on the basis of GPS was accepted as a correct correspondence. Note that while 40 m may seem too distant for a correct correspondence, on divided highways the minimum distance between correct loop closing poses was sometimes as large as this. Almost all loop closures detected by the system are well below the 40 m limit: 89% were separated by less than 5 m, and 98% by less than 10 m (See Figure 9).

We report precision-recall metrics for the system. Precision is defined as the ratio of true positive loop closure detections to total detections. Recall is the ratio of true positive loop closure detections to the number of ground truth loop closures. Note that images for which no loop closure exists cannot contribute to the true positive rate, however they can generate false positives. Likewise true loop closures which are incorrectly assigned to a "new place" depress recall but do not impact our precision metric. These metrics provide a good indication of how useful the system would be for loop closure detection as part of a metric SLAM system – recall at 100% precision indicates the percentage of loop closures that can be detected without any false positives that would cause filter divergence. Finally, note that a typical loop closure consists of a sequence of several images, so even a recall rate of 20% or 30% is sufficient to detect most loop closure events, provided that the detections have uniform spatial distribution.

B. Overall Performance

Overall, we found the system to have excellent performance on the 70 km data set, while the 1,000 km data set was more challenging. Precision recall curves for the two

TABLE I: Data set summary.

Data Set	No. of Images	No. of Loop Closures	Median distance between images	Extracted Features	Environment
1,000 km	103,256	48,493	8.7 m	177 GB	Motorways, Urban, Rural
70 km	9,575	4,757	6.7 m	16 GB	Urban, Rural

data sets are shown in Figure 10, and given numerically in Table II. A results video is available online in Extension 1. Loop closing performance is also visualized in the maps shown in Figures 15 and 16. Loop closures are often detected even in the presence of large changes in appearance, typical examples are shown in Figures 17 and 18. It is worth noting also that there are many examples of loop closures correctly detected by FAB-MAP but not by GPS, particularly under foliage and in city centres.

The performance contributions of the motion model and the geometric verification step are analysed in Figure 10 and presented numerically in Table II. The geometric check in particular is useful in maintaining recall at higher levels of precision. The motion model is largely unnecessary on the 70 km set. On this set we detect 44% of all pose-to-pose correspondences at 100% precision, *without using any temporal information*. These loop closures are detected on the basis of a single image. At 99% precision, the recall rises to 69.9%. On the 1,000 km set, the motion model makes a more noticeable contribution. Note, however, that the motion model we use is very weak. Stronger motion constraints, for example from a visual odometry system, would be expected to have a much larger impact. In combination with such motion information, it seems that it should be possible to achieve close to 100% recall on the 70 km set.

The effect of vocabulary size and the Chow Liu tree on performance is shown in Figure 11 and Table III. In common with other authors [31], [34], we find that performance increases strongly with vocabulary size. The Chow Liu tree also boosts performance on all data sets and at all vocabulary sizes. The effect is weaker at the very highest levels of precision. We discuss this in more detail in the next section.

The recall rate for the 70 km data set is 48.4% at 100% precision, rising to 73.2% at 99% precision. The spatial distribution of these loop closures is uniform over the trajectory – thus essentially every pose will be either detected as a loop closure, or a lie within a few meters of a loop closure. There are two short segments of the trajectory where this is not the case, one in a forest with poor lighting conditions, another in open fields with few visual landmarks. For practical purposes this data set can be considered "solved".

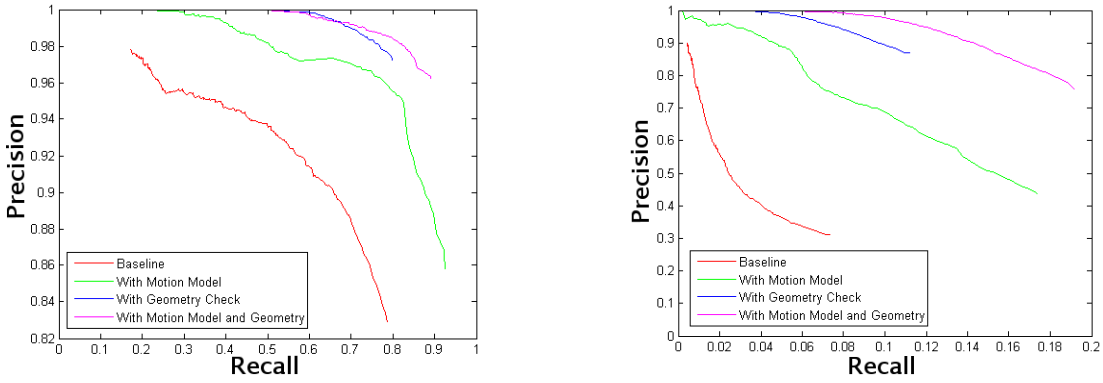
By contrast, the recall for the 1,000 km data set at 100% precision is only 3.1%. However, this figure requires careful interpretation – the data set contains hundreds of kilometers of motorways, where the environment is essentially devoid of distinctive visual features (see Figure 21). It is perhaps not reasonable to expect appearance-based loop closure detection in such conditions. To examine performance more closely, we considered separately the results for portions of the trajectory where the vehicle is travelling below 50 km/h (mainly urban areas). We refer to this evaluation as "1,000 km Urban" in Tables II and III. For these images (31% of the data set) the recall is 6.5% at 100% precision, rising to 18.5% at 99% precision. Note that the retrieval here is performed against the complete 1,000 km data set, the only salient difference being the distinctiveness of the query images. Given that the loop closures have an even distribution over the trajectory (Figure 15), even a recall rate of 6.5% is likely sufficient to support a good metric SLAM system.

Both data sets exhibit a sharp drop in recall between 99% and 100% precision. This drop is caused by particularly challenging cases of perceptual aliasing, such as encountering rare-but-repetitive objects in environment. Figure 19 shows the two highest confidence false positives from the 1,000 km set, typical of these difficult cases. The scenes have high similarity in both a bag-of-words and geometric sense, however the primary reason that they are difficult to identify as false positives is that the repetitive objects they contain are relatively uncommon in the environment, and so are not easily captured by the sampling set. By contrast, the scenes shown in Figure 21 do not cause such problems, despite high similarity, because the content of the image is common in the environment. Given that the system's sampling set consists of less than 2,000 images, these effects are perhaps not surprising. If a second navigation experiment were conducted using all of the 1,000km data for training, we may begin to develop robustness even to occasional repeated features such as those in Figure 19. Other natural methods to deal with these cases include relying more heavily on temporal support, or perhaps some level of semantic verification such as rejecting the loop closure in Figure 19b because the matched object is a vehicle.

C. The effect of the Chow Liu tree near 100% precision

It is a notable feature of Figure 11 that while the Chow Liu tree clearly improves the precision-recall curve up to the 99% precision point, it does not seem to give a consistent improvement at 100% precision. Performance at 100% precision is of most relevance to a metric SLAM system, which typically cannot recover from a false data association decision. This raises the question of whether the Chow Liu tree is actually of practical benefit in a loop closure detection system.

Firstly, we note that for some SLAM systems it may be possible to make use of loop closure signals which have

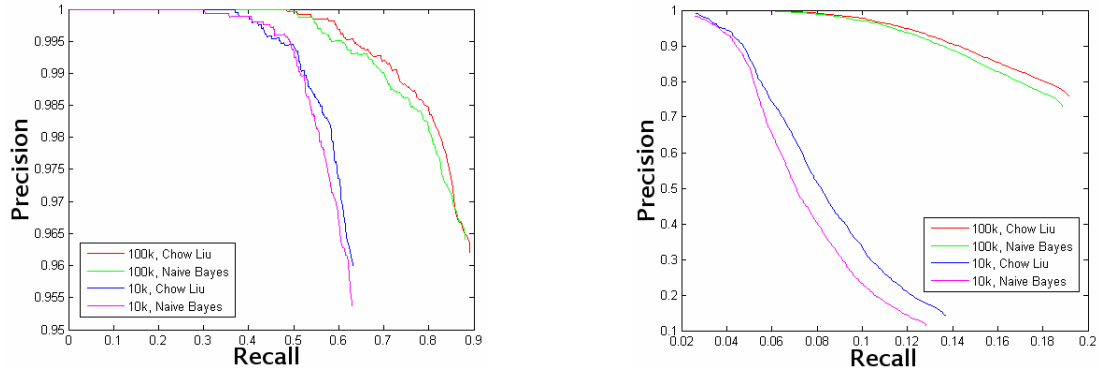


(a) 70 km set (b) 1,000 km set

Fig. 10: Precision-recall curves showing the effect of the different system components on performance. Note the scaling on the axes. Results shown are for the 100k vocabulary with Chow Liu tree. Relative performance in other configurations is similar. "Baseline" refers to the system without the geometric check and with a uniform position prior at each timestep. "Motion model" includes the position prior $p(L_i|\mathcal{Z}^{k-1})$, allowing loop closures to benefit from temporal support. "Geometric Check" re-ranks the top 100 most likely locations by considering the geometric consistency of matched image interest points.

Data Set	70 km		1,000 km			1,000 km Urban	
Precision	*100%*	*99%*	*100%*	*99%*	*90%*	*100%*	*99%*
Recall							
Motion Model and Geometric Check	48.5	73.2	3.1	8.3	14.3	6.5	18.5
Geometric Check Only	44.0	69.9	2.6	5.1	9.7	6.9	12.2
Motion Model Only	23.1	41.8	0.2	0.2	4.7	0.6	0.7
Baseline	-	-	-	-	0.5	-	-

TABLE II: Recall figures at specified precision showing the effect of different system components. A dash indicates that there is no threshold that produces the specified precision level. The same information is presented as a precision-recall curve in Figure 10. "Baseline" refers to the system without the geometric check and with a uniform position prior at each timestep. "Motion model" includes the position prior $p(L_i|\mathcal{Z}^{k-1})$, allowing loop closures to benefit from temporal support. "Geometric Check" re-ranks the top 100 most likely locations by considering the geometric consistency of matched image interest points.



(a) 70 km set (b) 1,000 km set

Fig. 11: Precision-recall curves showing the effect of the vocabulary size and the Chow Liu tree on performance. Note the scaling on the axes. Performance shown includes motion model and geometric check. Performance increases strongly with vocabulary size. The Chow Liu tree also increases performance, for all vocabulary sizes.

Data Set	70 km		1,000 km			1,000 km Urban	
Precision	*100%*	*99%*	*100%*	*99%*	*90%*	*100%*	*99%*
Recall							
100k CL	48.5	73.2	3.1	8.3	14.3	6.5	18.5
100k NB	49.1	70.0	3.7	7.9	13.5	7.5	17.9
10k CL	37.0	52.3	-	2.7	4.7	-	5.2
10k NB	30.1	51.5	-	-	4.4	-	-

TABLE III: Recall figures at specified precision for varying vocabulary size and with/without the Chow Liu tree. A dash indicates that there is no threshold that produces the specified precision level. The same information is presented as a precision-recall curve in Figure 11. Recall improves with increasing vocabulary size at all levels of precision. The Chow Liu tree also improves recall in all cases with the exception of the 100k vocabulary at 100% precision. The 100% precision figure is sensitive to the probability assigned to all possible false positives, so can be skewed by a single outlier with a high likelihood. So whereas the Chow Liu tree yields better probability estimates in general, the effect is more robustly observable at lower levels of precision, where it cannot masked by a small number of outliers.

less than 100% precision, if some secondary step can be used to increase the precision to 100%. For example, in the system of Willams et al. [41], when a putative loop closure is identified, the system attempts to track in the relevant section of the map. If the tracking fails, the loop closure is not accepted. In combination with a secondary step of this kind, the recall boost provided by the Chow Liu tree will be benificial.

However, it would obviously be preferable if we could determine why the Chow Liu tree does not naturally lead to higher recall at 100% precision. As noted in the previous section, the main difficulty in moving from 99% to 100% precision is overcoming a few very challenging examples of perceptual aliasing, such as those illustrated in Figure 19. The Chow Liu tree does not particularly help in dealing with false positives due to perceptual aliasing; its main purpose is to improve the similarity measure between images, allowing more difficult matches to be correctly identified (as evidenced by higher recall along most of the precision curve). Rejection of false matches due to perceptual aliasing is mainly achieved by the Monte Carlo integration of the partition function described at the end of Section III. This becomes the performance limiting factor at the top end of the precision recall curve, particularly in large data sets such as those considered here. No matter how good a similarity metric we learn (via the Chow Liu tree), recall at 100% precision cannot improve until we have a way to reject the (very visually similar) false positive matches that arise.

We conclude that the Chow Liu is indeed performing well, however it's impact is masked near 100% precision. We would expect the tree to have a bigger impact if (A) the perceptual aliasing is less severe (e.g. smaller environments, c.f. our earlier results in [12]), (B) the handling of perceptual aliasing was improved, perhaps via Monte Carlo integration over a larger sampling set, or via some other technique developed subsequent to this paper, or (C) the data was such that the performance-limiting factor was detecting difficult matches rather than rejecting perceptual aliasing.

A secondary factor which may be relevant is that while the Chow Liu tree will on average improve the likelihood estimates assigned, some individual likelihoods may get worse. The recall at 100% precision is determined by the likelihood assigned to the very last false positive to be eliminated. While on average we expect the Chow Liu tree to improve this likelihood estimate, the opposite may be observed in some fraction of data sets. Below 100% precision the results are sensitive to the likelihood estimates for a larger number of false positives, and so the improvement due to the Chow Liu tree is more robustly observable. However, we do not think that this is the dominant effect.

D. Comparison to tf-idf

Term-frequency inverse-document-frequency (tf-idf) is a standard ranking metric used in most existing visual search engines [39], [34], [21], [26]. To compare FAB-MAP against this baseline in the most transparent way possible, we examined performance on a pure retrieval task. For each image in our data sets where at least one valid match exists, we computed the ranking according to tf-idf weighted cosine distance and also according to the FAB-MAP likelihood $p(Z_k \mid L_i)$[3]. This ranking-only task is intended to examine the likelihood function alone, so does not involve new place detection, motion model effects or geometric re-ranking. For the 1,000 km data set there are 48,493 images that have at least one valid loop closure; for the 70 km set there are 4,757. Retrieval was performed against the set of images collected up to the point of loop closure. There are a variety of ways to perform the tf-idf weighting - we have followed [21] and have verified that our implementation gives results identical to those reported there.

Precision-recall curves showing relative performance are given in Figure 12. FAB-MAP substantially outperforms tf-idf, the difference being particularly dramatic on the 1,000 km data set.

The performance of tf-idf results could perhaps be improved by applying various known tweaks to the measure - for example by taking account of word burstiness [22] or using pivoted normalized document lengths [26] among others. However, it seems to us that increasing performance in this way essentially amounts to finding heuristics by trial-and-error. Indeed, to achieve the performance reported here

[3]Note that the tf-idf measure has access to word count (tf) information which is not used by FAB-MAP.

already required considerable experimentation with various aspects of the tf-idf measure, such as whether to use L1 or L2 normalization, whether to apply tf-idf weighting to query or document vectors or both, how the tf counts should be normalized, etc. Some of these choices, particularly the choice of vector normalization, have a dramatic impact on ranking performance, without any clear intuition as to why. FAB-MAP by contrast is a natural generative framework which provides clear rationale for the structure of the ranking function and offers paths to improved performance via extensions to the generative model. It also substantially out-performs tf-idf for our application of interest.

E. Timing

Timing performance is presented in Figure 14. Average filter update time over the 1,000 km data set, including the geometric check, was 14 ms. The time quoted was measured on a single core of a 2.40 GHZ Intel Core 2 processor. SURF feature extraction and kd-tree quantization adds an overhead of 484 ms on average, with typical variance illustrated in Figure 13. The cost is dominated by 423 ms for SURF. Recent GPU-based implementations can largely eliminate this overhead [10]. However, even including feature detection, our real time requirement of 480 ms could be achieved by simply spreading the processing over two cores.

F. Comparison to Original System

In comparison to the original system described in [12], the inference times of the system described here are on average 4,400 times faster, with comparable precision-recall performance. Equally important, the sparse representation means that location models now require only $O(1)$ memory, as opposed to $O(\#vocabulary)$. For the 100k vocabulary, a typical sparse location model requires 4 KB of memory as opposed to 400 KB previously. This enables the use of large vocabularies which improve performance, and is crucial for scalability because the size of the mappable area is effectively limited by available RAM.

VIII. Summary

This paper has outlined a new, highly scalable architecture for appearance-only SLAM. We have defined a new model that permits efficient inverted index implementation, while preserving the key benefits of our original Bayesian approach to the problem. The framework is fully probabilistic, and deals with challenging issues such as perceptual aliasing and new place detection. In addition to these benefits, as a pure ranking function it has been shown to considerably out-perform the baseline tf-idf approach. The paper also discussed techniques necessary for visual vocabulary generation and Chow Liu tree learning at large scale. On the issue of vocabulary learning, we have demonstrated the benefit of good cluster centre initialization on overall performance. Finally, we have evaluated the system on two substantial data sets, of 70 km and 1,000 km. Both experiments are larger than any existing result we are aware of. Our approach shows very strong performance on the 70 km experiment, in conditions of challenging perceptual aliasing. For practical purposes this set can be considered solved, and moreover this performance can be achieved on the basis of single images, without temporal information. The 1,000 km experiment is more challenging, and we do not consider it fully solved, nevertheless our system's performance is already sufficient to provide a useful competency for an autonomous vehicle operating at this scale. Our data sets are available to the research community, and we hope that they will serve as a benchmark for future systems.

Acknowledgments

The work reported in this paper was funded by the Systems Engineering for Autonomous Systems (SEAS) Defence Technology Centre established by the UK Ministry of Defence and by the EPSRC.

References

[1] A. Angeli, D. Filliat, S. Doncieux, and J.A. Meyer. A Fast and Incremental Method for Loop-Closure Detection Using Bags of Visual Words. *IEEE Transactions On Robotics, Special Issue on Visual SLAM*, 24(5):1027–1037, 2008.

[2] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded Up Robust Features. In *Proc 9th European Conf on Computer Vision*, volume 13, pages 404–417, Graz, Austria, May 7 2006.

[3] J.-L. Blanco, J.-A. Fernandez-Madrigal, , and J. Gonzalez. Towards a unified bayesian approach to hybrid metric-topological SLAM. *IEEE Transactions on Robotics*, 24:259–270, 2008.

[4] M. Bosse and R. Zlot. Keypoint design and evaluation for place recognition in 2D lidar maps. In *Robotics: Science and Systems Conference : Inside Data Association Workshop*, 2008.

[5] M. Bosse and R. Zlot. Map matching and data association for large-scale two-dimensional laser scan-based SLAM. *International Journal of Robotics Research*, 2008.

[6] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30, 1998.

[7] J. Callmer, K. Granstrom, J. Nieto, and F. Ramos. Tree of Words for Visual Loop Closure Detection in Urban SLAM. In *Proceedings of the Australasian Conference on Robotics and Automation*, 2008.

[8] C.K. Chow and C.N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, IT-14(3), May 1968.

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

[10] N. Cornelis and L. Van Gool. Fast scale invariant feature detection and matching on programmable graphics hardware. In *CVPR 2008 Workshop CVGPU*, 2008.

[11] M. Cummins and P. Newman. Accelerated appearance-only SLAM. In *Proc. IEEE International Conference on Robotics and Automation (ICRA'08)*, Pasadena, California, April 2008.

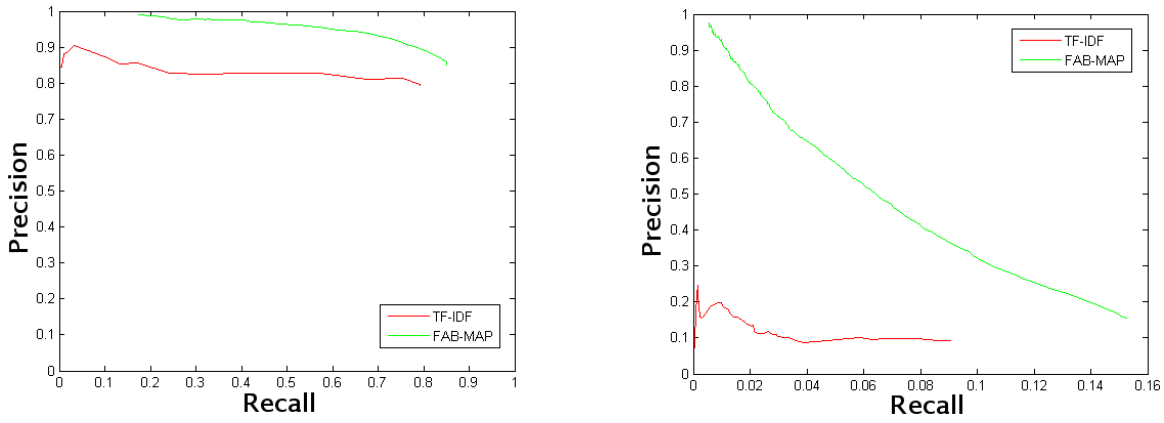
[12] M. Cummins and P. Newman. FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008.

[13] M. Cummins and P. Newman. Highly scalable appearance-only SLAM - FAB-MAP 2.0. In *Proceedings of Robotics: Science and Systems*, Seattle, USA, June 2009.

[14] Mark Cummins. *Probabilistic Localization and Mapping in Appearance Space*. PhD thesis, University of Oxford, 2009.

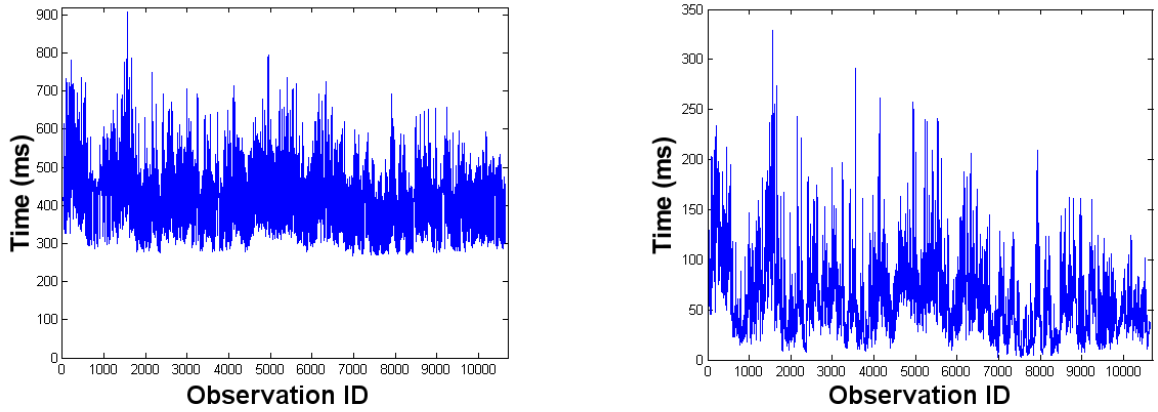
[15] S. Dasgupta and P.M. Long. Performance guarantees for hierarchical clustering. *Journal of Computer and System Sciences*, 70(4):555–569, 2005.

[16] R. Dementiev, L. Kettner, and P. Sanders. STXXL: Standard template library for XXL data sets. *Software: Practice and Experience*, 38(6), 2008.



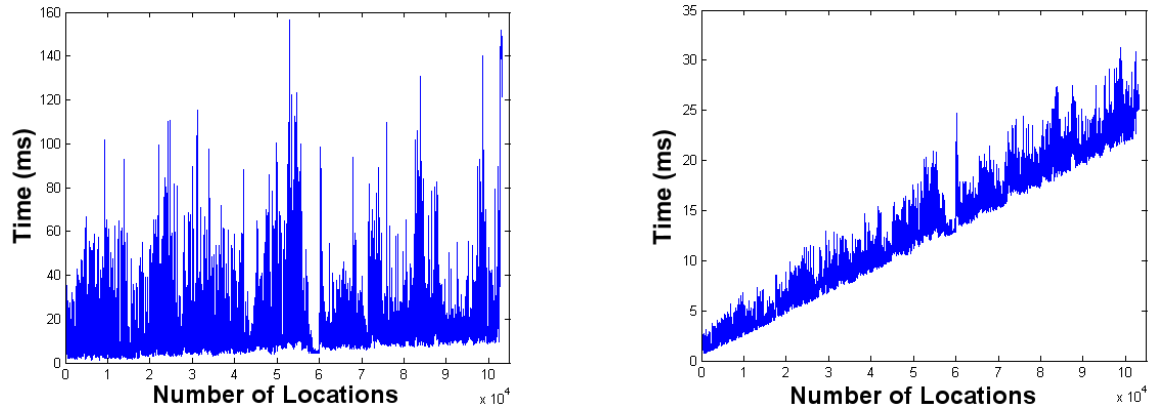
(a) 70 km set (b) 1,000 km set

Fig. 12: Comparison to tf-idf ranking. FAB-MAP substantially outperforms this standard ranking metric, particularly on the larger data set. To ensure the fairest possible comparison, these performance figures relate to a retrieval-only task that excluded new place detection. See Section VII-D for details.



(a) SURF generation time. (b) Quantization time using the kd-tree.

Fig. 13: Bag-of-words generation time (per Ladybug2 panoramic image) for a representative sample of the 1,000 km data set, using the 100k vocabulary. The time is dominated by SURF generation, (a), which takes 423 ms on average. Quantization using the randomized kd-trees, (b), takes on average 60 ms.



(a) Total inference time. (b) Inference time, excluding geometric verification.

Fig. 14: Filter update times on the 1,000 km data set for the 100k vocabulary, (a). Mean filter update time is 14 ms and maximum update time is 157 ms. The cost is dominated by the RANSAC geometric verification, which has O(1) complexity. The core ranking stage excluding RANSAC, (b), exhibits linear complexity but with a very small constant - taking 25 ms on average with 100,000 locations in the map.

[17] G. Dudek and D. Jugessur. Robust place recognition using local appearance based methods. In *Proceedings of IEEE International Conference on Robotics and Automation*, volume 2, 2000.

[18] E. Eade and T. Drummond. Unified loop closing and recovery for real time monocular slam. In *Proc. 19th British Machine Vision Conference*, 2008.

[19] M.A. Fischler and R.C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[20] F. Fraundorfer, C. Engels, and D. Nistér. Topological mapping, localization and navigation using image collections. In *International Conference on Intelligent Robots and Systems*, 2007.

[21] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In Andrew Zisserman David Forsyth, Philip Torr, editor, *European Conference on Computer Vision*, volume I of *LNCS*, pages 304–317. Springer, oct 2008.

[22] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. On the burstiness of visual elements. In *IEEE Conference on Computer Vision & Pattern Recognition*, June 2009.

[23] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *Tenth IEEE International Conference on Computer Vision, 2005. ICCV 2005*, volume 1, 2005.

[24] K. Konolige, J. Bowman, J.D. Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua. View-based maps. In *Proceedings of Robotics: Science and Systems (RSS)*, 2009.

[25] M. Magnusson, H. Andreasson, A. Nuchter, and A.J. Lilienthal. Appearance-Based Place Recognition from 3D Laser Data Using the Normal Distributions Transform. In *IEEE International Conference on Robotics and Automation*, 2009.

[26] C. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[27] M. Meilă. An accelerated Chow and Liu algorithm: Fitting tree distributions to high-dimensional sparse data. In *ICML '99: Proceedings of the Sixteenth International Conference on Machine Learning*, pages 249–257, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[28] M.J. Milford and G.F. Wyeth. Mapping a Suburb With a Single Camera Using a Biologically Inspired SLAM System. *IEEE Transactions on Robotics*, 24(5):1038–1053, 2008.

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

[30] P. Newman, G. Sibley, M. Smith, M. Cummins, A. Harrison, C. Mei, I. Posner, R. Shade, D. Schröter, L. Murphy, W. Churchill, D. Cole, and I. Reid. Navigating, recognising and describing urban spaces with vision and laser. *The International Journal of Robotics Research*, 2009.

[31] D. Nistér and H. Stewenius. Scalable recognition with a vocabulary tree. In *Conf. Computer Vision and Pattern Recognition*, volume 2, pages 2161–2168, 2006.

[32] E. Olson. *Robust and Efficient Robotic Mapping*. PhD thesis, Massachusetts Institute of Technology, June 2008.

[33] R. Paul and P. Newman. FAB-MAP 3D: Topological Mapping with Spatial and Visual Appearance. In *IEEE International Conference on Robotics and Automation*, 2010.

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

[35] I. Posner, M. Cummins, and P. Newman. Fast probabilistic labeling of city maps. In *Proc. Robotics: Science and Systems*, Zurich, June 2008.

[36] Ananth Ranganathan. *Probabilistic Topological Maps*. PhD thesis, Georgia Institute of Technology, 2008.

[37] G. Schindler, M. Brown, and R. Szeliski. City-Scale Location Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, 2007.

[38] C. Silpa-Anan and R. Hartley. Optimised kd-trees for fast image descriptor matching. In *Computer Vision and Pattern Recognition*, 2008.

[39] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the International Conference on Computer Vision*, Nice, France, October 2003.

[40] I. Ulrich and I. Nourbakhsh. Appearance-based place recognition for topological localization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1023 – 1029, April 2000.

[41] B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid, and J. Tardos. An image-to-map loop closing method for monocular SLAM. In *Proc. International Conference on Intelligent Robots and and Systems*, 2008.

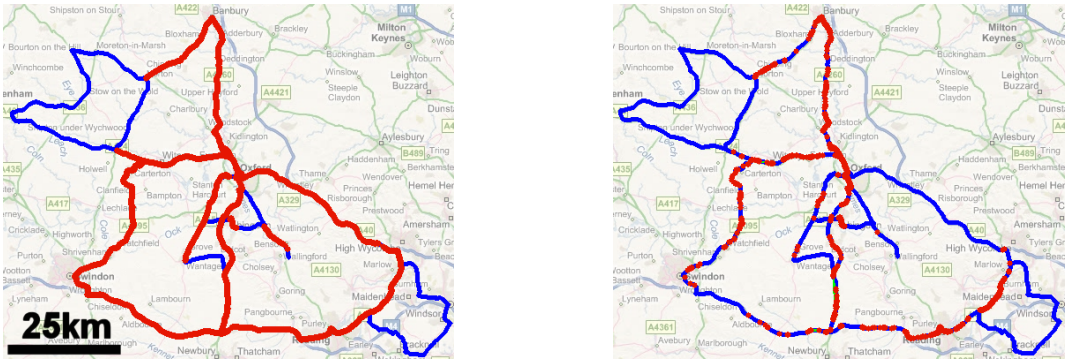
[42] Z. Zhu, T. Oskiper, S. Samarasekera, R. Kumar, and H.S. Sawhney. Ten-fold improvement in visual odometry using landmark matching. In *Intl. Conf. on Computer Vision (ICCV)*, 2007.

Appendix

The multimedia extensions to this article can be found online by following the hyperlinks from http://www.ijrr.org.

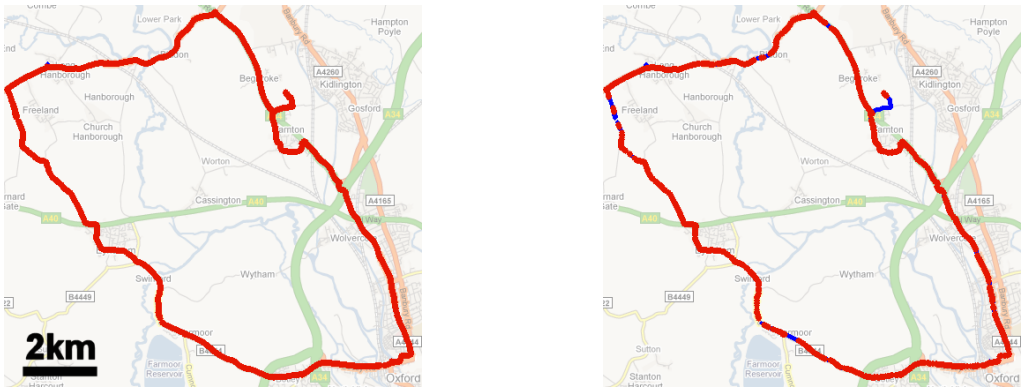
Extension	Media Type	Descriptio
1	Video	Results video for 70 km and
2	Data	70 km data set, available at http://www.robots.o
3	Data	1,000 km data set, availa

TABLE IV: Index of multimedia extensions



(a) (b)

Fig. 15: Loop closure maps for the 1,000 km data set. Best viewed in colour. Sections of the trajectory where loop closures exist are shown in red. (a) The ground truth. (b) Loop closures detected by FAB-MAP (100k CL), showing 99.8% precision and 5.7% recall. There are 2,819 correct loop closures and six false positives. False positives are marked with a green line between poses, however the six present here are spatially close, so are not readily visible on the map. The long section on the right with no detected loop closures is a motorway at dusk. The section on the bottom left with intermittent loop closures is also a motorway.



(a) (b)

Fig. 16: Loop closure maps for the 70 km data set. Best viewed in colour. Sections of the trajectory where loop closures exist are shown in red. (a) The ground truth. (b) Detected loop closures using FAB-MAP (100k CL), at 100% precision. The recall rate is 48.4%. However, the system clearly detects loop closures in almost all parts of the trajectory. The recall rate reflects the fact that not every possible image along the trajectory is matched. Two short sections of the trajectory generate fewer loop closures – one is in a forest, where imaging conditions were poor, the other is in open fields, with few visual landmarks. A total of 2,300 loop closures are detected, with no false positives.

(a) At the confidence value of this match, the precision is 99.6%.

(b) At the confidence value of this match, the precision is 100%.

Fig. 17: Some correct loop closures from the 1,000 km data set. The system typically finds correct matches in the presence of considerable scene change when the image content is distinctive.

(a) At the confidence value of this match, the precision is 100%.

(b) At the confidence value of this match, the precision is 99.9%.

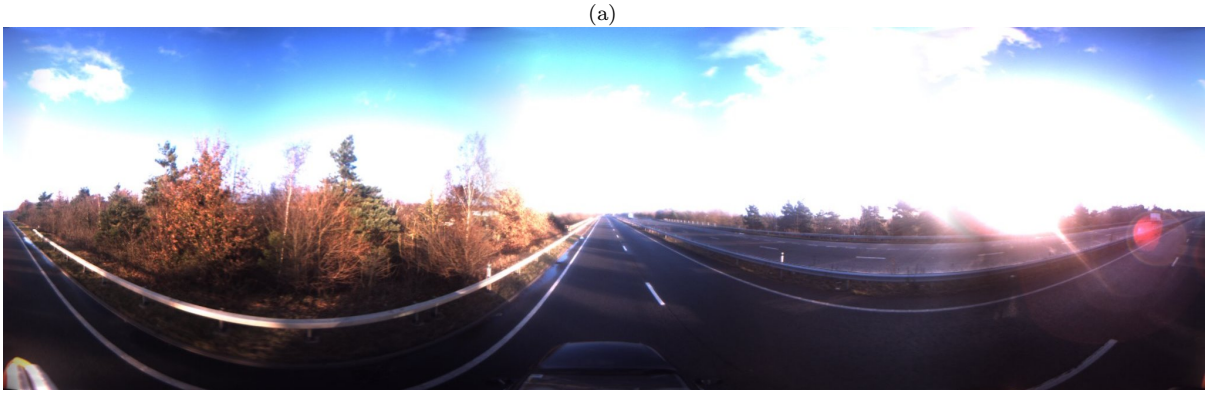
Fig. 18: Some correct loop closures from the 70 km data set. These are not unusual matches. The system typically finds correct matches in the presence of considerable scene change when the image content is distinctive.

(a)

(b)

Fig. 19: The two highest confidence false positives in the 1,000 km data set. Both matches are assigned probabilities very close to 1. In (a), we pass a similar-looking roundabout. The locations are 1 km apart. In (b), we encounter the same van twice. The locations are 9 km apart. Such rare-but-repetitive objects represent the most challenging class of perceptual aliasing.

(a)



(b)

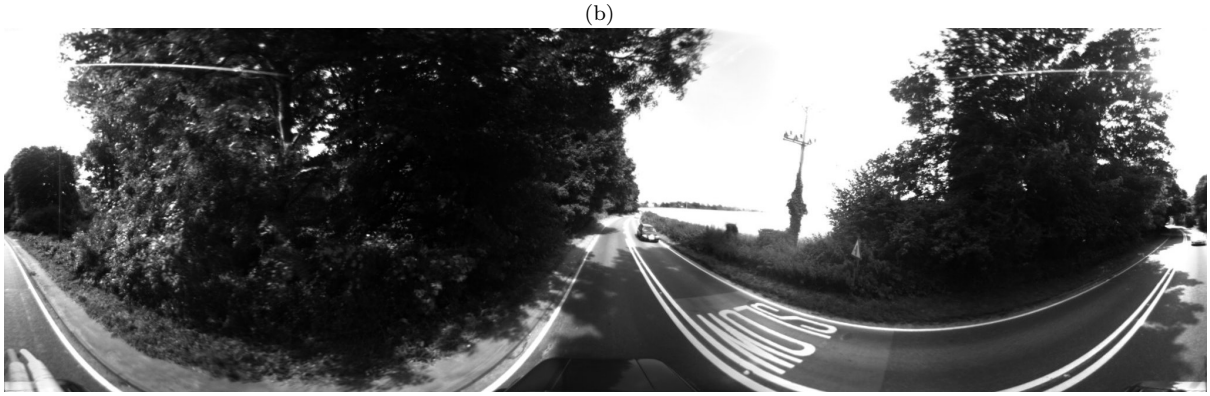


Fig. 20: Some examples of perceptual aliasing correctly handled by the system. In (a), the locations are 4 km apart. Their highly similar appearance is typical of motorway driving. However, this similarity does not lead to a false positive loop closure detection because this repetitive aspect of the environment has been captured in the sampling set used to evaluate the partition function $p(Z_k \mid \mathcal{Z}^{k-1})$ (see Section III). This allows the system to assign the newly collected image a "new place" probability of 0.9997. A similar case from the 70 km set is shown in (b).

(a) First image captured at the location.

(b) Image collected at loop closure.

(c) A second, unrelated location in the map, with very similar appearance.

Fig. 21: A typical false negative. Figures (a) and (b) come from the same location, but the loop closure is not detected by FAB-MAP. Figure (c) shows a second, unrelated location, to illustrate the self-similar character of the route. The 1,000 km sequence contains hundreds of kilometers of such motorway scenes, so the system's inability to correctly identify this loop closure is unsurprising. This effect depresses the recall in the 1,000 km results. However, the strong perceptual aliasing generates very few false positive detections. In the above case, the new observation is assigned to a new place with probability 0.9994. This is possible because these common modes of perceptual aliasing are easily captured by the sampling set.