

Probabilistic Localization and Mapping in Appearance Space

Mark Cummins
Balliol College



Supervisor:
Dr Paul Newman

Robotics Research Group
Department of Engineering Science
University of Oxford

October 2009

“Begin at the beginning”, the King said gravely, “and go on till you come to the end: then stop.”

– *Lewis Carroll, Alice’s Adventures in Wonderland*

Abstract

This thesis is concerned with the problem of place recognition for mobile robots. How can a robot determine its location from an image or sequence of images, without any prior knowledge of its position, even in a world where many places look identical? We outline a new probabilistic approach to the problem, which we call Fast Appearance Based Mapping or FAB-MAP. Our map of the environment consists of a set of discrete locations, each with an associated appearance model. For every observation collected by the robot, we compute a probability distribution over the map, and either create a new location or update our belief about the appearance of an existing location. The technique can be seen as a new type of SLAM algorithm, where the appearance of locations (rather than their position) is subject to estimation. Unlike existing SLAM systems, our appearance based technique does not rely on keeping track of the robot in any metric coordinate system. Thus it is applicable even when informative observations are available only intermittently. Solutions to the loop closure detection problem, the kidnapped robot problem and the multi-session mapping problem arise as special cases of our general approach.

Our probabilistic model introduces several technical advances. The model incorporates correlations between visual features in a novel way, which is shown to improve system performance. Additionally, we explicitly compute an approximation to the partition function in our Bayesian formulation, which provides a natural probabilistic measure of when a new observation should be assigned to a location not already present in the map. The technique is applicable even in visually repetitive environments where many places look the same.

Finally, we define two distinct approximate inference procedures for the model. The first of these is based on concentration inequalities and has general applicability beyond the problem considered in this thesis. The second approach, built on inverted index techniques, is tailored to our specific problem of place recognition, but achieves extreme efficiency, allowing us to apply FAB-MAP to navigation problems on the largest scale. The thesis concludes with a visual SLAM experiment on a trajectory 1,000 km long. The system successfully detects loop closures with close to 100% precision and requires average inference time of only 25 ms by the end of the trajectory.

Statement of Authorship

This thesis is submitted to the Department of Engineering Science, University of Oxford, in fulfilment of the requirements for the degree of Doctor of Philosophy. This thesis is entirely my own work, and except where otherwise stated, describes my own research.

Mark Cummins, Balliol College

Funding

The work reported in this thesis was funded partly by the Systems Engineering for Autonomous Systems (SEAS) Defence Technology Centre established by the UK Ministry of Defence and partly by the Engineering and Physical Sciences Research Council.

We would also like to acknowledge Google Inc. for providing the City Block data set, which was helpful during the development of the work reported here.

Acknowledgements

First and foremost, thanks are due to my supervisor Paul Newman. Paul was everything a good supervisor should be. Some wise words at the right times, but more than anything else, the freedom to explore interesting things. There was also an unforgettable moment in an airport.

Thanks are also due to the many people who have given me a helping hand at one point or another. To Iead Rezek for some early pointers that proved very fruitful. To Rohan Paul, without whom the 1,000 km data set would never have been possible. Thanks also for teaching me a lot of Indian history along the way – that was a good week, apart from the roundabouts. And of course, thanks to all the members of MRG and AVL; for discussions, for friendship and for good times.

Finally, but most importantly, to my family, who have been there through thick and thin, before and after.

Contents

1	Introduction	8
1.1	Motivation	8
1.2	Outline of the Problem	10
1.3	Contributions	10
1.4	Structure of the Thesis	11
1.5	Publications	12
2	Background	13
2.1	The Robot Navigation Problem	13
2.1.1	Overview	14
2.1.2	Extended Kalman Filter SLAM	14
2.1.3	Particle Filter SLAM	15
2.1.4	Other Estimation Techniques	16
2.1.5	Navigation and Mapping without a Global Coordinate Frame	17
2.1.6	Challenging Problems in SLAM	18
2.1.6.1	Local Data Association	18
2.1.6.2	The Loop Closure Detection Problem	20
2.1.6.3	The Kidnapped Robot Problem and Multi-Session Mapping	23
2.2	Content Based Image Retrieval Systems	23
2.2.1	The Bag of Words Representation	24

2.2.2	Interest Point Detectors	26
2.2.3	Interest Point Descriptors	27
2.2.4	Detector and Descriptor used in this Thesis	28
2.2.5	Partitioning of Descriptor Space	28
2.2.6	Notable Image Retrieval Systems	29
2.3	Appearance Based Techniques in Robotics	31
2.3.1	Localization using Appearance	32
2.3.2	Loop Closure Detection	33
2.3.3	Recent Developments	34
3	The FAB-MAP Model	36
3.1	Approximating High Dimensional Discrete Distributions	37
3.1.1	Chow Liu Trees	38
3.1.2	Sample Complexity of Learning	40
3.1.3	Smoothing Probability Estimates	40
3.1.4	What the Chow Liu Tree Learns	42
3.2	Probabilistic Navigation using Appearance	43
3.2.1	Representing Appearance	43
3.2.2	Representing Locations	44
3.2.3	Inference in the FAB-MAP Model	45
3.2.4	Evaluating The Model	46
3.2.5	Independence Assumptions, Approximations and Input Parameters	54
3.2.6	Discussion: Alternative Generative Models	56
3.3	Implementing the Model	60
3.3.1	Converting Images to Bag-of-Words Representation	60
3.3.2	Learning the Vocabulary Model	61
3.3.3	Algorithm Configuration	62
3.4	Performance Evaluation	63

3.4.1	Data Sets	65
3.4.2	Performance Metrics	66
3.4.3	Results	66
3.4.4	Generalization Performance	75
3.4.5	Use with Metric SLAM Systems	77
3.5	Chapter Summary	80
4	Approximate Inference using Concentration Inequalities	81
4.1	Introduction	81
4.2	Bail-Out Strategies	81
4.3	Probabilistic Bail-Out using Bennett's Inequality	82
4.4	Application to Appearance-Only SLAM	85
4.4.1	Applicability of Bennett's Inequality	86
4.4.2	Ranking Features	87
4.4.3	Calculating M and v	88
4.4.4	Calculating a pdf over Hypotheses	89
4.5	Results	89
4.6	Chapter Summary	95
5	Inverted Index Techniques for Large Scale Place Recognition	96
5.1	Related Work	97
5.2	System Description	98
5.2.1	Inverted Indices	98
5.2.2	FAB-MAP 2.0 - An Approximation to FAB-MAP for Large Scale Applications	98
5.2.3	Geometric Verification	103
5.2.4	Visual Vocabulary Learning At Large Scale	106
5.3	Data Set	108
5.4	Results	110
5.4.1	Test Conditions	110

5.4.2	Overall Performance	112
5.4.3	Comparison to tf-idf	114
5.4.4	Timing	117
5.4.5	Comparison to Original System	118
5.5	Chapter Summary	118
6	Discussion	127
6.1	Summary of Contribution	127
6.2	Future Work	128
6.3	Concluding Remark	131
A	Data Collection Platforms	132
B	Chow Liu Trees	135
C	Manipulations	138
C.1	FAB-MAP Model	138
C.2	Bennett Bound	139

Nomenclature

e_q	The event that an object (scene element) which generates observations of type z_q exists at a location.
L_i	The i^{th} location in the map.
$\mathcal{L}^k$	The set of all locations in the map at time k .
Z_k	The k^{th} observation collected by the robot.
z_q	The event that the q^{th} word of the visual vocabulary was observed.
z_{p_q}	An observation of word p_q , the parent of word q in the Chow Liu tree.
$\mathcal{Z}^k$	The set of all observations up to time k .

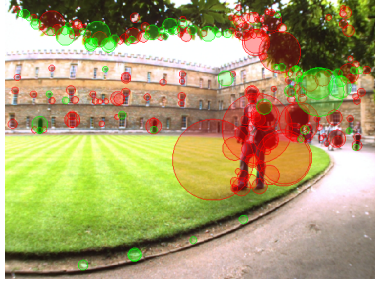
Chapter 1

Introduction

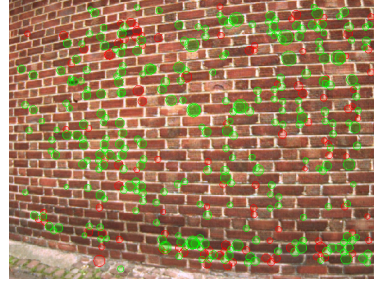
1.1 Motivation

The overall aim of this thesis is to contribute to the development of autonomous mobile robots. For several decades, the practical application of robotics has remained limited to carefully controlled environments such as assembly lines. Automation of the vast majority of real-world tasks remains out of reach, because our best systems cannot handle the complexity of unconstrained environments. The research community continues to make steady progress, but the critical level of robustness has yet to be attained. Crossing the threshold will open up a huge field of new possibilities – from personal service robots in the home to habitat monitoring in the oceans. We continue to pursue that goal.

In particular, this thesis will be concerned with autonomous navigation. To be useful, a robot must be able to reliably find its way between locations in its environment. For some applications, solutions based on human-constructed maps and external infrastructure such as the Global Positioning System (GPS) are sufficient, but in many cases the navigation problem must be solved on the basis of the robot's internal sensors alone. This might be dictated by considerations of flexibility and costs, or by necessity. For example, no infrastructure is available to planetary exploration robots, and GPS is unavailable in many terrestrial environments such as indoors, near tall buildings, under foliage, underground and underwater. In these situations the robotic system must solve the navigation problem without external



(a) Perceptual Variability



(b) Perceptual Aliasing



Figure 1.1: Appearance based navigation systems must assess the probability that two images come from the same place. This is difficult because place appearance can vary, (a), and more challengingly, because different places in the environment can appear identical, (b). The appearance based navigation problem is thus distinct from typical content based image retrieval tasks.

assistance. This problem of navigation without prior knowledge and without external infrastructure is known as Simultaneous Localization and Mapping (SLAM), and has been a core area of research in mobile robotics for several decades. While much progress has been made, a robust and general solution is still lacking. This thesis aims to increase the robustness of navigation systems by formulating a new approach to SLAM based on appearance rather than metric information. Our approach allows the robot to incrementally construct a model of the appearance of locations in the environment, and to globally re-localize by recognizing places it has previously visited. As a special case it solves a problem known as loop closure detection which has been a persistent difficulty for SLAM systems.

1.2 Outline of the Problem

Concretely, the problem that will be addressed in this thesis is the following: Given two observations or sequences of observations (typically images), what is the probability that these observations were collected at the same location?

This is a hard problem for a number of reasons. Firstly, the world is dynamic. Two images of the same location may look different due to changes in lighting or viewpoint, arrival or departure of vehicles, changes in weather, etc. Secondly, and more challengingly, the world is visually repetitive. Common textures such as brickwork and foliage are present everywhere. Mass produced objects and symbols such as road markings appear throughout the environment. These problems are illustrated in Figure 1.1. Under many reasonable metrics, the image pair shown in Figure 1.1b is more similar than the pair in 1.1a. In our bag-of-words representation (introduced later), 1.1b share 28% of their visual words, whereas 1.1a share only 15%. However, we would like to build a system that asserts with high confidence that 1.1a shows two images of the same location, whereas 1.1b probably does not. To do this, we must go beyond a simple computation of image similarity, and build a model that allows us to evaluate the *distinctiveness* of images. This will involve learning a probability distribution over the space of images, which captures the fact that the scene in Figure 1.1b is common in the environment, so despite the similarity the images are unlikely to come from the same location.

1.3 Contributions

This thesis makes the following contributions to the field:

- The first fully probabilistic framework for localization and mapping using appearance which is naturally robust to perceptual aliasing. Our technique can be used alone, or as a loop closure detection component of a larger robot navigation system.
- A direct approach to calculating the probability that a new observation comes from a location not already in the map, which proceeds by computing an approximation of the partition function in our probabilistic model. To our knowledge this is the first method that goes beyond simple

heuristics such as fixed similarity thresholds. In our experience the new approach is a dramatic improvement, particularly in that it naturally handles challenging cases such as perceptual aliasing.

- A novel technique for taking account of correlations between visual words when computing observation likelihoods. We learn an approximation to the joint distribution over visual words, which captures the fact that objects in the data generate correlated sets of features. We explore efficient learning methods which allow us to apply the technique to large visual vocabularies ($>100,000$ words), and demonstrate that it improves performance. The approach should have applicability beyond the problem at hand.
- An approximate inference technique based on the use of concentration inequalities. The technique is general in its applicability, and allows large speed increases to be realized in return for a small user-controlled probability of generating an incorrect result. For our problem, the technique yielded speed improvements in the range 25x-50x with minimal degradation in accuracy.
- Several public benchmark data sets for evaluating appearance-only SLAM systems, including a 1,000 km set which is, to our knowledge, the most extensive benchmark set available by a considerable margin. The set contains 20 Hz stereo imagery, omni-directional imagery every 5-10 m and GPS ground truth.

1.4 Structure of the Thesis

In Chapter 2 we present a review of relevant background material and related work in image retrieval and robot navigation. Chapter 3 describes a new probabilistic framework for appearance-based navigation, our “Fast Appearance Based Mapping” or “FAB-MAP” model. Chapter 4 describes a novel approach to approximate inference for more rapidly computing position under this model. Chapter 5 describes a second approximation technique suitable for very large scale applications of appearance-based navigation, and presents experimental results on a 1,000 km test trajectory. Conclusions and an outline of possible future research directions are presented in Chapter 6.

1.5 Publications

The system described in Chapter 3 was published in ICRA 2007 [30] with subsequent extension in IJRR [32]. The techniques in Chapter 4 appeared in ICRA 2008 [31], and were awarded the Best Vision Paper prize. Those of Chapter 5 appeared in RSS 2009 [33]. Various applications of the system and related work appeared in [110, 109, 91, 164, 163, 107, 125, 124].

Chapter 2

Background

This chapter introduces background material relevant to the thesis. We draw on two general strands of literature, namely the work in the robotics community on the Simultaneous Localization and Mapping (SLAM) problem, and work in the computer vision community on content-based image retrieval system. We begin by discussing the SLAM problem in robotics.

2.1 The Robot Navigation Problem

”The true logic of this world is in the calculus of probabilities.”

– *James Clerk Maxwell.*

A mobile robot exists in a state of continual uncertainty. Commands sent to actuators may not be reliably carried out. Sensor readings are subject to noise and bias. Motion models are only approximations. Yet for a robot to be able to carry out useful tasks, it must be able to reliably navigate between locations in its environment. For reasons of flexibility and cost, we would like the robot to achieve this task entirely autonomously, without the need for manually constructed maps or robot-specific navigational aids such as beacons. How this may be achieved has been at the core of robotics research for several decades, where it is referred to as the Simultaneous Localization and Mapping (SLAM) problem. Early navigation algorithms that ignored uncertainty were brittle and subject to silent failure. Essentially all

successful approaches to SLAM (and indeed most problems in robotics) embrace the uncertainty of the robot’s perceptual world by framing the problem as a Bayesian inference task. We begin this section by introducing the concept of probabilistic navigation and describe some of the different algorithms used, along with a discussion of their failure modes.

2.1.1 Overview

A SLAM algorithm must maintain a probability distribution x that describes the robot’s belief about its position and the position of objects in the world, given a stream of observations Z (typically laser range measurements, camera images, odometry, etc). If the k^{th} observation is denoted Z_k and the set of all observations up to time k is denoted $\mathcal{Z}^k$, then SLAM can be formulated as a recursive Bayesian estimation problem:

$$p(x|\mathcal{Z}^k) = \frac{p(Z_k|x)p(x|\mathcal{Z}^{k-1})}{p(Z_k|\mathcal{Z}^{k-1})} \quad (2.1)$$

Depending on the choice of underlying world representation and the representation of the pdf x , this equation leads to the various SLAM algorithms. The most prominent approaches are based on the Extended Kalman Filter and the Rao-Blackwellized Particle Filter. The details of these algorithms will not concern us in this thesis; we limit ourselves to a brief discussion of their salient performance characteristics only. A more detailed introduction can be found in [41, 3].

2.1.2 Extended Kalman Filter SLAM

The seminal 1990 paper of Smith, Self and Cheeseman [146] introduced the widely used Extended Kalman Filter (EKF) formulation of SLAM, where the world is represented as a set of landmarks, and the vehicle and landmark covariances are approximated by Gaussian distributions. The EKF-based approach to SLAM was a revolutionary improvement and continues to be widely used. However it suffered from a number of performance-limiting issues, most prominently computational scaling, linearization and the assumption of Gaussianity.

Scaling issues (the fact that the computation needed to maintain a map increases quadratically in the number of landmarks) have largely been resolved by subsequent research – successful approaches

include submapping strategies [120, 13] and information form methods [159]. The fastest methods claim $O(1)$ complexity per update [78], at the expense of slower filter convergence.

However, the issue of linearization largely remains. “Linearization errors” arise from the fact that the Kalman filter is fundamentally applicable only to linear processes. The Extended Kalman Filter extends the technique to the non-linear case by linearizing all functions of interest about their current estimates. This approximation is a source of error, particularly if the function being estimated is highly non-linear, or if the state estimate is biased. The problem is partly addressed by techniques such as the Unscented Kalman Filter [68] and Iterated Extended Kalman Filter [4], and partly by submapping techniques with appropriate choice of coordinate frames [21]. However, linearization remains an essentially unavoidable problem within the EKF framework.

Finally, for many purposes the largest limitation of EKF-based methods is the assumption that the underlying vehicle and landmark covariances are well-approximated by Gaussians. There are many situations where this is not the case, for example in the presence of large angular errors, or when the pdf becomes multi-modal. In such cases a navigation system must use techniques beyond the EKF framework, or suffer an inconsistent map estimate.

2.1.3 Particle Filter SLAM

An alternative solution to SLAM based upon a Rao-Blackwellized particle filter was first proposed by Montemerlo, Thrun, Koller and Wegbreit in 2002 [100] and later improved in [101]. The solution (known as FastSLAM) exploits the fact that the landmark position estimates are conditionally independent given the trajectory of the vehicle. The FastSLAM algorithm maintains the distribution over vehicle position using a set of particles, where each particle possesses N low-dimensional EKFs to represent the N landmarks. In contrast to EKF SLAM, the particle filter based approach can accurately represent the arbitrary distributions over vehicle position that may arise from non-linear vehicle kinematics or situations where the position uncertainty is multi-modal. Additionally, the approach does not suffer from significant linearization issues and in combination with some efficient data structures allows updates to be computed in $O(M \log N)$ time for M particles (as opposed to $O(N^2)$ for the EKF).

The main difficulty with FastSLAM is maintaining sufficient *diversity* in the particle set over

long trajectories. FastSLAM applies a resampling step during filter updates, which causes a loss of information about trajectory history and is strictly only applicable for systems which “*exponentially forget*” their past (i.e. where the current state of the system tends towards independence from earlier states) [29]. The SLAM problem does not have this exponential forgetting property, and as a result FastSLAM and related techniques can have difficulties building consistent maps over longer trajectories containing multiple loops.

2.1.4 Other Estimation Techniques

The core estimation problem of SLAM has been approached from a number of directions in addition to the popular EKF and particle filter formulations. Within the Bayesian filtering framework, the Thin Junction Tree Filter proposed by Paskin [118, 119] approximates inter-landmark correlations by a thin junction tree. The algorithm achieves linear complexity in both time and space, and is related to submapping approximations to the EKF.

Square Root Information Smoothing [37, 70] proposed by Dellaert in 2005 moves away from the Bayesian filtering perspective entirely and instead treats SLAM as incremental smoothing problem, where the complete trajectory of the robot is recovered. This is in contrast to filtering approaches to SLAM that only maintain an estimate of the robot’s current location. In the smoothing framework, the entire past trajectory of the robot is subject to revision at each update. This means that smoothing approaches, unlike filtering ones, never make irreversible linearization errors, so the full non-linear solution can be obtained. The disadvantage of this scheme is that, because no marginalization of past poses is ever carried out, the number of variables in the estimator increases continuously while the robot is operating. However, because the relevant matrices typically remain sparse, the system can be solved in real time even for large environments. The same theme of applying incremental non-linear optimization techniques to SLAM was also proposed by Frese *et al.* [49] in 2005, who introduce a multi-level relaxation algorithm for SLAM that is derived from multigrid methods used for solving partial differential equations.

Recently popular are so-called pose graph optimization methods, for example the work of Olson *et al.* [116] and Grisetti *et al.* [54, 53]. These techniques apply a variant of Stochastic Gradient Descent

(a non-linear optimization method) to minimize the error in the network of constraints between robot poses. The techniques recover maximum likelihood map estimates only, but typically converge rapidly and recover good trajectories even when initial pose estimates are poor. Some non-linear optimization of this kind is usually necessary to recover good global map structure after an event known as *loop closure*, where the robot revisits a previously mapped area after traversing a large loop through unmapped terrain. In this condition the robot’s position estimate and uncertainty may differ significantly from the ground truth, a condition which is not well handled by linearizing filters such as the EKF.

2.1.5 Navigation and Mapping without a Global Coordinate Frame

The dominant theme within the SLAM literature has been an approach known as *metric SLAM*, which aims to recover a globally consistent map of the environment in a single Cartesian coordinate frame. All algorithms discussed so far fall into this category. However, a robot can successfully perform most tasks with a map that satisfies much weaker requirements — for example, a map which is metrically consistent on the local scale, and merely topologically consistent on a global scale. In some cases a purely topological map, which specifies connectivity between landmarks but contains no metric information whatsoever, will suffice.

Topological theories of SLAM have existed for some considerable time [73, 132, 23], however these early topological mapping algorithms were not probabilistic, and hence difficult to make robust. Recently Ranganathan and Dellaert outlined a fully probabilistic framework for topological SLAM [127, 129, 128], which enabled principled treatment of the uncertainty in landmark identification. However, the computational complexity of the approach has so far limited its applicability to small environments containing only a few tens of landmarks.

Recent work by Sibley *et al.* [139, 138] proposes an interesting compromise which is metric in construction, but abandons the notion of a single Cartesian coordinate frame, thus leading to a global structure which is topological in character. Relative transformations are estimated between consecutive robot poses, but closed loops through the transformation graph are not constrained to be the identity transformation. This allows the map to represent unusual types of connectivity, such as between two floors of a building via an elevator, where the robot’s ego-motion through the elevator is unobservable.

The resulting “map” does not necessarily have a Cartesian embedding, yet is nevertheless perfectly applicable for navigation tasks such as path planning and obstacle avoidance.

2.1.6 Challenging Problems in SLAM

A shared difficulty for all the SLAM algorithms described so far, regardless of estimation technique or map representation, is an issue known as *data association*. Data association is the problem of relating sensor measurements to objects in the robot’s map — a distance measurement to an object doesn’t tell us much about where we are unless we know which object we are observing. Data association also involves determining if measurements are spurious, or belong to a previously unobserved object not yet in the map. Correct data association is essential for successful navigation; errors rapidly lead to an incorrect map.

Theoretically, data association could be treated as simply another uncertain quantity to be estimated concurrently with the rest of the robot state. However, in practice this approach is intractable because the possible assignments of measurements to objects grow combinatorially¹. In deriving SLAM algorithms, it is frequently assumed that an “oracle” is available that supplies the data association information. In practice, of course, no oracle is available and some method of relating measurements to objects must be found. This is a non-trivial problem, and the core topic of this thesis. The limitations of the heuristic data association methods currently in use are the root cause of some of the most traditionally challenging problems in SLAM — the *loop closure detection* problem, the *kidnapped robot* problem and the *multi-session mapping* problem. A truly reliable method for general data association solves all these problems at once.

2.1.6.1 Local Data Association

The problem of associating measurements to map landmarks is commonly solved with very simple heuristic methods. One of the most common approaches is nearest neighbour gating (sometimes known as maximum likelihood data association), where a measurement is assigned to the “most likely” landmark in the map. Typically “most likely” is defined as the nearest landmark in the sense of Mahalanobis

¹The closest approach to treating the problem directly is [128].

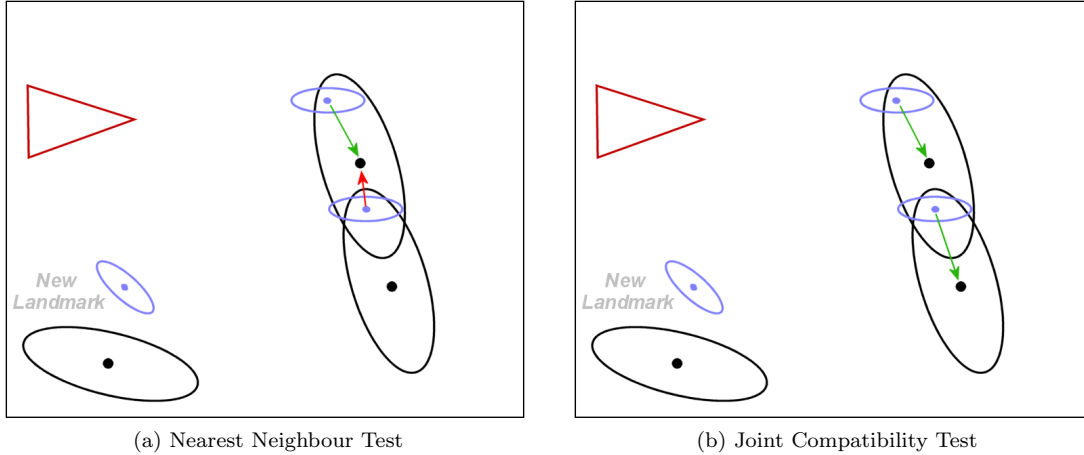


Figure 2.1: The local data association problem. Black points indicate estimated landmark locations with associated uncertainty ellipses. Measurements are shown in blue. In (a), nearest-neighbour assignment results in two measurements being assigned to a single landmark, corrupting the robot’s map estimate. In (b), the joint compatibility test [106] results in correct data association. Note that both the nearest neighbour and joint compatibility tests rely on a good prior estimate of robot location. Also shown is a measurement lies further than the gating threshold from any landmark, which will cause a new landmark to be added to the map.

distance — a distance metric which accounts for the fact that position uncertainty is not uniform in all directions [86]. Measurements are assigned to a new landmark if the most likely match is farther than some threshold. The scheme is illustrated in Figure 2.1.

The nearest neighbour gating scheme is very brittle. It is really applicable only when environmental clutter is low, sensor precision is high and vehicle pose uncertainty is low. Nevertheless, the basic approach is in common use. This is despite the fact that errors in data association typically cause catastrophic divergence of the map. To increase the robustness of the technique, a number of more advanced variants have been described. For example, the Joint Compatibility Branch and Bound technique [106] considers multiple measurements simultaneously, and so decreases the error rate by taking into account local compatibility constraints between measurements (Figure 2.1(b)). Another approach is to make use of additional sensory information associated with landmarks. Laser scan matching algorithms make use of larger scale shape information [134], visual SLAM systems make use of texture information associated with landmarks [36], etc. A second level of robustness can be achieved through multi-hypothesis tracking, where ambiguous data association cases are handled by tracking

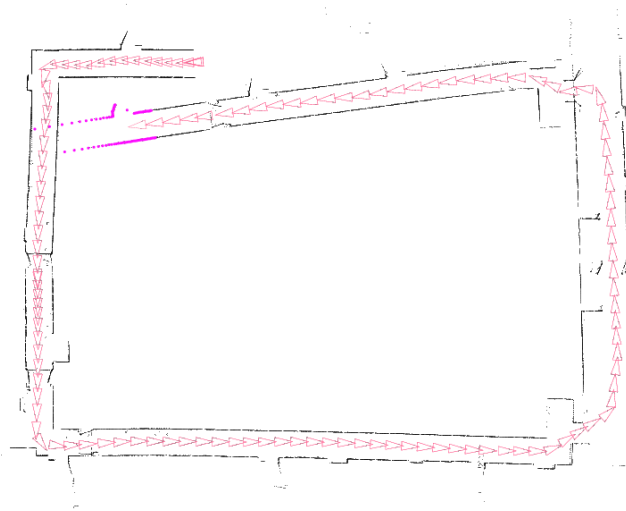


Figure 2.2: A example of loop closure detection failure in an indoor environment. Triangles show vehicle poses. As the robot travels along the corridor, linearization and perception errors accumulate. When the robot returns to the starting point, its true location lies outside the three-sigma bound on vehicle uncertainty (not shown). The nearest neighbour data association technique fails, and the robot’s map becomes corrupted. (Figure courtesy of Oxford Mobile Robotics Group)

both possibilities in the hope that the ambiguity will be resolved by subsequent measurements [131]. However, all of these techniques are essentially similar in that they address the local data association problem — that is, they work only if we have a good initial guess for the robot’s position within the map. Such information is usually not available in a situation known as loop closure, where other data association techniques are required.

2.1.6.2 The Loop Closure Detection Problem

A particularly difficult instance of the data association problem is known as *loop closure detection*. Loop closure detection is essentially the problem of data association without a good prior guess for the robot’s position within the map. The problem commonly arises when the robot is returning to a previously mapped location after traversing a large loop through an unmapped area. In such cases the robot’s position estimate can differ markedly from the ground truth. This does not imply that the robot made gross errors at any point, rather that small errors have been integrated over a large path length — for example in Figure 2.3, small angular errors along the path result in the end points

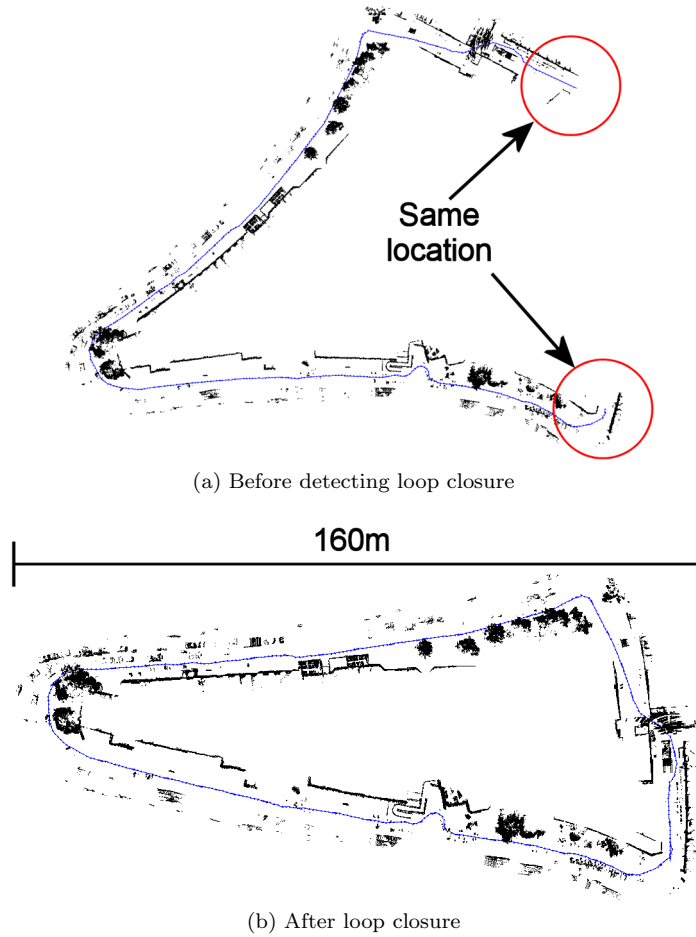


Figure 2.3: An illustration of the difficulty of loop closure detection using global metric estimates of vehicle position. (Figure courtesy of Alastair Harrison and Dave Cole.) Sub-figure (a) shows the vehicle trajectory estimate produced by a scan-matching SLAM system operating in an outdoor environment [111]. The end points of the trajectory are at the same location; however small angular errors along the trajectory have produced a large difference between the true and estimated positions of the vehicle. The corrected trajectory is shown in sub figure (b), where the loop closure has been detected on the basis of an image match. It would seem that in large-scale environments, such a loop closure detection system is essential for reliable navigation.

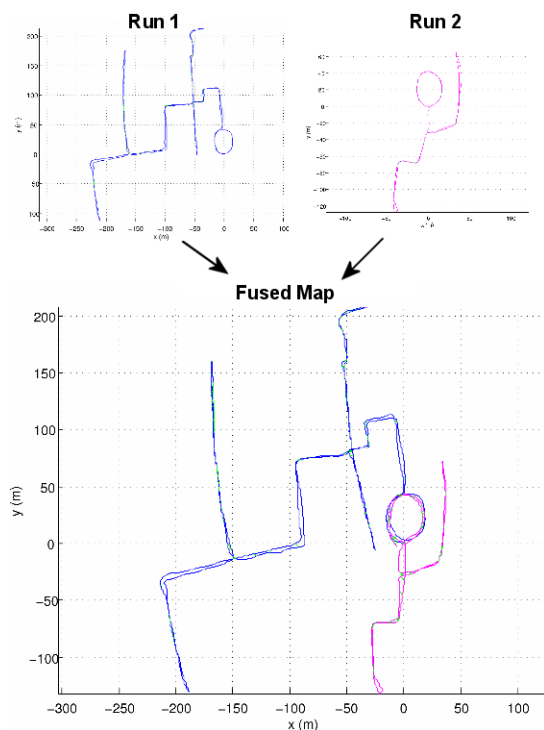


Figure 2.4: Multi-session mapping. Two partial maps of the environment were collected on different days. The transformation between the two map coordinate frames is unknown and must be recovered to produce the fused map.

of the loop having widely separated global position estimates. The problem is compounded by the fact that, for trajectories involving uncertain rotations, the vehicle position uncertainty cannot be correctly represented by Gaussian models such as the EKF. A second failure example, from an indoor environment, is shown in Figure 2.2.

In the case of loop closure, where the robot's estimate of its position is in gross error, local data association techniques such as those discussed in the previous section are inapplicable. Relying on such techniques alone will result in previously mapped areas being remapped in a new global location and a vehicle that is effectively lost. A great many techniques have been proposed to solve the loop closure detection problem; we will review them in detail in Section 2.3.

2.1.6.3 The Kidnapped Robot Problem and Multi-Session Mapping

The kidnapped robot problem and the multi-session mapping problem are two further tasks that have traditionally been challenging for SLAM systems. The kidnapped robot problem is the task of determining a robot’s pose within a map, without prior position information. It might occur if a robot is deactivated and then moved, experiences a temporary sensor failure, etc. The multi-session mapping problem is the task of aligning two partial maps of the environment collected by the robot during different periods of operation, illustrated in Figure 2.4. The rotation, translation (and in some cases scale) between the two maps is unknown and must be determined automatically. Both of these tasks are common and important elements of an autonomous system capable of life-long operation.

As with loop closure detection, both kidnapped robot and multi-session mapping can be seen as a subset of the data association problem without prior position information, although problem-specific techniques have been proposed that solve these problems individually without tackling the more general case. We will review some of these techniques when we consider the SLAM literature on place recognition in Section 2.3. First, however, we introduce relevant background work in computer vision.

2.2 Content Based Image Retrieval Systems

The second strand of literature drawn upon in this thesis is the work on content based image retrieval systems in the computer vision community. Given a “query image”, these systems attempt to rapidly locate other images of the same or similar objects within a large image collection, even in the presence of image transformations such as perspective or lighting change, partial occlusion, etc. Notable examples of such systems include Sivic and Zisserman [144], Nister and Stewénus [114], Philbin *et al.* [122] and Jégou *et al.* [63]. Essentially all of these systems are based on quantized invariant local features, also known as the “bag-of-words”, “bag-of-visual-words” or “bag-of-visual-features” representation. We begin our discussion by introducing this image representation.

2.2.1 The Bag of Words Representation

To efficiently find image matches in large databases under viewpoint transformation, the choice of representation of the image data is crucial. Some early image retrieval systems represented images using global features, where a single descriptor is computed for an entire image (for example, global colour histograms [149, 156] or PCA components [154, 72]). Such global features are not very robust to effects such as variable lighting, perspective change or partial occlusion of the scene. An alternative approach pioneered in the late 1990s by Schmid and Mohr [137] locates *interest points* in the image and computes *invariant features* (or “*descriptors*”) in the neighbourhood of these interest points. The interest point detectors and local descriptors can be designed such that they are repeatable even after to a variety of image transformations such as scaling, rotation and linear lighting changes. To perform image search, the local features are computed in the query image, and for each local feature a “vote” is cast for images in the image database which contains a similar local feature. The voting procedure enables object recognition even when the query contains a partially occluded view of the object being searched for, and also provides robustness against a detector/descriptor which is only stochastically repeatable. Note that the representation discards information about the relative geometric configuration of the interest points. This local-feature paradigm proved highly successful, especially after the introduction of the SIFT descriptor by Lowe [83].

Image search via direct local descriptor matching is effective in image collections of moderate size (hundreds to thousands of images), but becomes computationally difficult for larger image collections (tens of thousands to millions of images). The computational bottleneck is the matching of descriptors in the query image to near neighbours in the image database. A typical image may yield thousands of local descriptors, so matching a descriptor from a query image typically involves searching through tens to hundred of millions of descriptors in the image database. Descriptors are typically high-dimensional (for example, the SIFT descriptor is 128 dimensional), and nearest neighbour search for high dimensional data is computationally challenging. The most widely used practical search algorithm, the kd-tree [102], is not much more efficient than linear search when the dimensionality is high. Approximate nearest neighbours can be found much more quickly, for example using the Best Bin First variant of

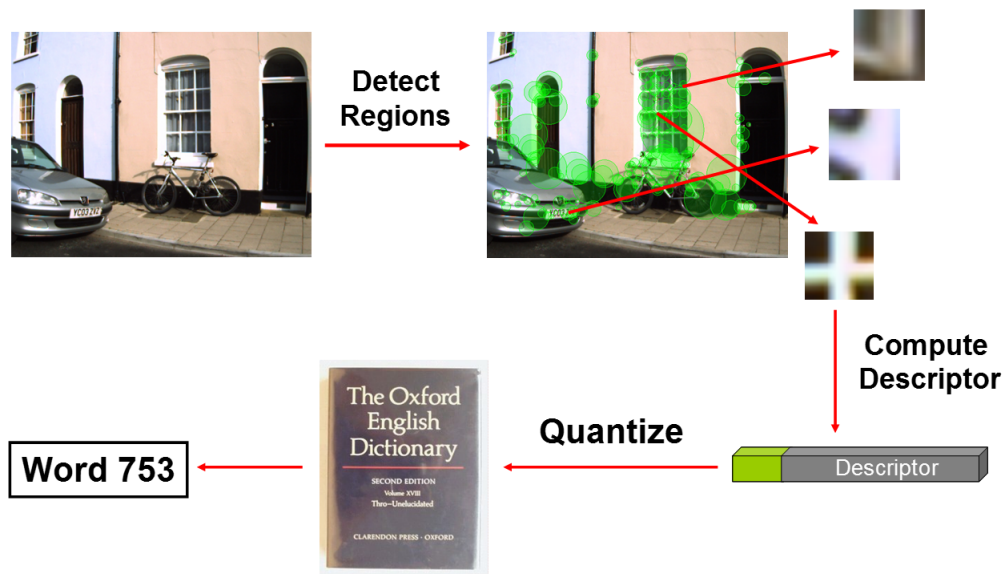


Figure 2.5: Illustration of how an image is converted to the visual word representation. First, regions of interest are detected in the image. An invariant descriptor is computed for each interest region. These descriptors are then quantized with respect to a dictionary of prototypical features, and the descriptor is replaced by an integer identifying the nearest prototypical descriptor in the dictionary (“*visual word id*”). The whole image can then be represented by a short list of integers specifying which visual words occurred in the image, and how often each word occurred.

kd-tree search [8], but for very large image search problems even these techniques are both too slow and (especially) too memory intensive to be practical.

The key innovation that enabled search of very large image collections was the “bag of words” model [136, 80, 144], which quantizes image descriptors against a “vocabulary” of prototypical descriptors. The vocabulary is learned by clustering descriptors from training data. Quantization allows a high-dimensional image descriptor to be replaced by a single integer (or *word ID*) specifying which prototypical descriptor in the vocabulary it is most similar to. The process is illustrated in Figure 2.5. A *word* thus corresponds to a region in the space of invariant descriptors, the assumption being that descriptors within this region can be considered identical for the purposes of the task at hand. Images can now be represented as a short list of integers specifying which visual words occurred in the image, and how often they occurred — a representation identical to that used for text documents in internet search engines. The representation thus enables fast visual search through the application of methods developed for

text retrieval. These techniques were first applied to large scale visual search in the work of Sivic and Zisserman [144].

For the purposes of this thesis we will take the visual word representation as fundamental. The details of the interest point detectors and descriptors that underlie the representation will not be of particular concern to us. It is sufficient to know that we have some algorithm that takes an image and produces a list of vector-valued features that are somewhat repeatable and somewhat invariant to common image transformations. Our primary concern will be in constructing a probabilistic inference framework on top of this representation. For completeness, however, we briefly review the various detectors and descriptors in the following sections.

2.2.2 Interest Point Detectors

A large number of interest point detectors have been proposed in the literature. Popular examples include Harris corners [56], Harris-Laplace and Hessian-Laplace points and their affine-invariant extensions the Harris-Affine and Hessian-Affine points [96], the Difference of Gaussians scale space extrema used in SIFT [84], Maximally Stable Extremal Regions (MSERs) [90], and the box filter approximation to the determinant of the Hessian used in SURF (known as Fast-Hessian) [6]. Most of these detectors select local maxima of simple operators in image scale space, and typically respond to structures such as corners or blobs. An extensive performance evaluation was carried out by Mikolajczyk *et al.* [98], where the repeatability of the interest points was tested with respect to viewpoint change, zoom, rotation, blur, JPEG compression and lighting change. Overall the Hessian-Affine and MSER detectors were found to perform best, MSERs being particularly robust to viewpoint and lighting change and Hessian-Affine being better in the presence of blur and JPEG compression. The Fast-Hessian operator used by SURF was later found to have performance comparable or slightly superior to Hessian-Laplace at much lower computational cost [6], though not competitive with Hessian-Affine in the case of large viewpoint changes.

2.2.3 Interest Point Descriptors

A wide variety of interest point descriptors have also been described. The dominant paradigm is the histogram-type descriptor introduced by Lowe [84], which is based on the spatial distribution of local features within the interest point neighbourhood. Lowe’s SIFT descriptor is a histogram of local image gradient responses in 8 orientations over a 4x4 spatial grid, yielding a 128D vector descriptor. Rotation and illumination invariance is achieved by intensity normalizing the image patch and rotating in the direction of dominant local gradient. These histogram-type descriptors have excellent performance because they capture sufficient information so as to be highly discriminative, yet are tolerant of small image deformations and errors in interest point localization. An extensive evaluation by Mikolajczyk and Schmid [95] found them to consistently outperform a wide variety of alternatives.

Several papers have investigated variations on the original SIFT design. The SURF descriptor, based on histograms of Haar wavelet responses, broadly retains the idea of a descriptor based on the spatial distribution of local gradients, but changes the details to aid efficient implementation. The resulting descriptor was shown to have matching performance superior to SIFT, while also being faster to compute [6, 7]. Further performance improvements were achieved by Winder and Brown [165, 166], where the form of the descriptor was learned directly so as to maximize matching performance on training data. The resulting descriptor designs substantially out-perform SIFT in terms of both error rate and descriptor size – the best configuration resulting in an error rate half that of SIFT and a memory footprint 10x smaller. Other recent innovations include GPU-based implementations of descriptors and detectors, which enables feature extraction in only a few milliseconds on typical 640x480 images [28]. Finally, a notable alternative to the SIFT-like descriptor is the randomized tree technique of Lepetit *et al.* [79], which poses keypoint matching as a classification problem. Randomized trees are favoured in some real-time systems because they can be computed very rapidly even on standard hardware.

2.2.4 Detector and Descriptor used in this Thesis

As noted previously, the work described in this thesis is descriptor-agnostic. During development we experimented with several possibilities, principally the Harris-Affine detector with SIFT descriptor, and the SURF detector/descriptor. The real-time performance requirements of our robotics application lead us to choose the SURF descriptor for all final results. We use the U-SURF variant with 128D descriptor [6]. We disable rotation invariance because this version of the descriptor is faster to compute and, because in-plane rotation is a rare occurrence on our robot platform, it also provides a modest increase in matching performance. Some of the more recent results in descriptor design, notably those of [166], might provide performance improvements for future versions of our system.

2.2.5 Partitioning of Descriptor Space

The final step in producing a bag-of-words representation of an image is to construct a partitioning of descriptor space that defines the mapping from interest point descriptor to visual word. The standard approach is to cluster a set of descriptors and use the cluster centres to implicitly divide the descriptor space into Voronoi regions. See Figure 2.6. The most common choice of clustering algorithm is k-means [144]. For larger vocabularies hierarchical k-means [114] or approximate k-means [122] are used for tractability. Others have investigated incremental radius-based clustering [69, 60, 45] to overcome some problems with k-means, an issue we will discuss in detail in Chapter 5. Other techniques such as agglomerative clustering have also been considered [77, 97], though primarily for problems such as object class recognition where vocabularies are smaller and so more expensive clustering procedures are feasible. An alternative to clustering was explored in [155] where a regular grid was used to discretize the descriptor space.

In our experience the choice of clustering algorithm is of secondary importance, at least for smaller retrieval problems. Indeed, an early version of our system had a bug in the clustering code that resulted in very poor quality clusterings. This went unnoticed for several weeks because the system continued to detect loop closures successfully. This behaviour should not be too surprising — the primary purpose of the clustering is to partition the descriptor space into regions, rather than to locate structure in

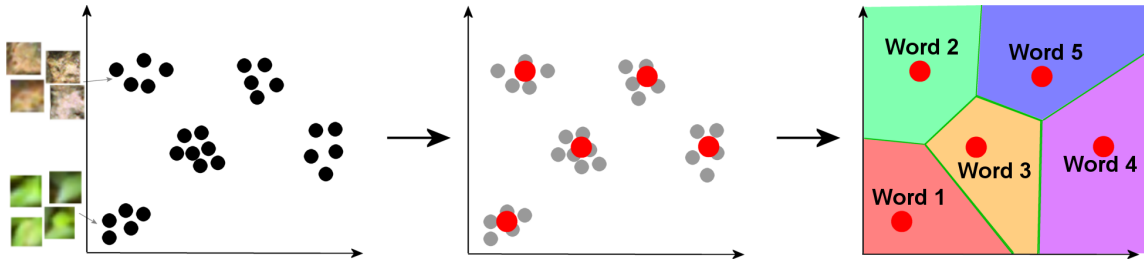


Figure 2.6: Descriptor space partitioning (shown in 2D for illustration). Descriptors from similar real-world objects tend to cluster in descriptor space - for example, foliage may form one cluster and brickwork another. Cluster centres (shown in red) are found via an unsupervised clustering algorithm such as k-means. The descriptor space is then implicitly divided into visual words, defined by the Voronoi regions of the cluster centres.

the data. While intuition would suggest that good quality clusterings should improve performance because they keep quantization boundaries away from dense regions of the data (which would tend to amplify quantization error due to small changes in descriptors), retrieval should still be possible with a random partitioning of descriptor space. Only highly degenerate partitionings where a small number of visual words occupy a large volume of descriptor space are likely to actually “break” the system. These issues are relevant when considering generalization performance – for example, should we expect a robot whose visual vocabulary has been trained on indoor images to be able to navigate in an outdoor environment? As might be clear from the preceding discussion, we expect system performance to degrade gracefully in such situations. We will observe this behaviour in practice in Chapter 3.

2.2.6 Notable Image Retrieval Systems

Large scale image retrieval using the bag-of-words approach was first introduced by Sivic and Zisserman in [144]. Later Nister and Stewénus [114] developed the vocabulary tree which combined vocabulary generation, quantization and indexing into a single structure based on a hierarchical k-means clustering of the descriptors in the image database. The paper explored the use of very large visual vocabularies (up to 10^6 words), and demonstrated that this gave a significant increase in performance. A hard drive based scheme for extending the approach beyond the RAM limits of a single machine was discussed in [48].

Philbin *et al.* [122] also experimented with the use of large visual vocabularies, and demonstrated that an approximate k-means clustering combined with a randomized forest of kd-trees for quantization was superior to the hierarchical k-means approach of [114]. Philbin *et al.* additionally introduced a fast method for verifying the geometric consistency of putative image matches returned by the bag-of-words ranking procedure. The approach took advantage of the shape information in the affine-invariant interest points, yielding a method fast enough to apply to the top several hundred images in the ranking. The geometric verification step gave a sizable performance improvement, particularly with smaller visual vocabularies or larger retrieval sets, however at the cost of increasing the memory requirements of the index. Subsequent work by the same authors extracted further performance improvements through the use of relevance feedback [26] and explored the impact of quantization accuracy on system performance [123] through the use of soft assignment. Soft assignment gave performance improvement, but again at the cost of increasing the storage requirements of the index. Recently Perd'och *et al.* [121] proposed a technique to reduce the memory required to store the shape information of the interest points used during the geometric verification stage.

Jégou *et al.* [63, 64] introduced a technique called the Hamming Embedding for improving quantization beyond the granularity of visual words, by storing a short binary code for each point descriptor. They also introduced a method to build weak geometric consistency directly into the ranking stage (rather than as a post-processing step). The geometric ranking required some restrictive assumptions, but proved to increase performance on typical data.

A common denominator of the systems discussed so far is that similarity between bag-of-words vectors used in the core ranking stage of retrieval is computed using a term-frequency inverse-document-frequency (tf-idf) weighted cosine distance. Tf-idf is a standard ranking measure used throughout information retrieval [87], which builds on the intuition that two documents containing a rare word should be more similar than those that share a common word. However, it has relatively little theoretical justification beyond the fact that it has good empirical performance on text retrieval tasks. Most production text retrieval systems apply many tweaks to the measure to overcome various deficiencies [87]. A core argument of this thesis will be that the tf-idf ranking is not particularly suitable for image data, and that by paying attention to the specific qualities of imagery, much better ranking measures

can be derived. Some very recent literature has explored similar ideas; for example [65], where some heuristic ranking measures are proposed to better account for the “bursty” statistics of visual words. Also of note are some recent papers which, with different motivation, have exploited the co-occurrence of visual features. For example [153] uses fast query-dependent AND searches to rapidly locate a small set of images for more complex ranking, and [168] is one of several examples of approaches that combine multiple visual features into meta-features.

Finally, some interesting techniques have been proposed in papers dealing with problems closely related to image search, such as image clustering and object recognition with large image databases. Chum *et al.* introduced geometric min-Hash [25], a hashing based retrieval scheme that incorporates the local geometry of interest points directly into the retrieval stage. This was demonstrated in the context of automatically discovering repeated views of objects in the image set. Torralba *et al.* [151, 161] investigated a number of techniques for learning small-code approximations to image descriptors. An entire image is represented by a code of only a few hundred bits, giving a retrieval scheme which is fast to search and extremely memory efficient. A search index for tens of millions of images can be stored in less than 1 GB RAM. However, the technique has been evaluated only for retrieving images that are weakly similar to the query image; it is not clear how the method would perform for the exact matching tasks considered by the other papers discussed here.

2.3 Appearance Based Techniques in Robotics

We now return to considering the appearance based navigation problem in robotics. In Section 2.1 we introduced the loop closure detection problem and discussed the difficulty of addressing it within a typical metric SLAM framework. The root cause of this difficulty is the simplicity of the heuristics used for data association, which typically require a very good prior position estimate to be effective. To better address the general data association problem, *appearance based techniques* have been developed which exploit the rich appearance information from sensors such as cameras to determine when the robot is revisiting a previously mapped place. Typically these techniques can perform a global search through previously seen locations, so they are not reliant on position priors and are a useful approach

to loop closure detection. Further, because location estimates based on appearance have uncertainty essentially orthogonal to that in metric estimates, they are a useful compliment to metric SLAM systems in general. We review the development of appearance based techniques in this section.

2.3.1 Localization using Appearance

We begin our review of the literature by considering appearance based localization techniques. Although the localization problem is considerably simpler than SLAM (the difficult issue of new place detection does not arise), appearance based approaches to the two problems share many common concerns. In particular, the appearance based localization literature has extensively examined the issue of how best to represent and match sensory appearance information. Early examples include representing appearance as a set of image histograms [156], and as ordered sequences of edge and colour features [76]. Several authors explored the use of more general dimensionality reduction techniques. For example, Kröse *et al.* [72] used Principal Component Analysis to reduce the dimensionality of the incoming images. Places are then represented as a Gaussian density in the low dimensional space, which enables principled probabilistic localization. Ramos *et al.* [126] employed a dimensionality reduction technique combined with variational Bayes learning to find a generative model for each place. (A drawback of both of these approaches is that they require significant supervised training phases to learn their generative place models). Bowling *et al.* [18] described an unsupervised approach that uses a sophisticated dimensionality reduction technique called Action Respecting Embedding; however the method suffered from high computational cost [17] and yielded localization results in a “subjective” representation which is not straight forward to interpret. Applying image-specific features has generally been a more successful approach — for example, Torralba *et al.* [152] represented places by a set of texture features, and described a system where localization is cast as estimating the state of a hidden Markov model.

The methods described so far represent appearance using global features, where a single descriptor is computed for an entire image. As discussed in our overview of the computer vision literature in Section 2.2, representations based on sets of local features were found to be a more robust approach. Recent appearance-based techniques use this representation. An early example of the approach was described in [143, 40]. More recently, [167] used an image retrieval system based on invariant features as the basis of

a Monte Carlo localization scheme. The issue of selecting the most salient local features for localization was considered in [82]. Most recently the bag-of-words representation has become popular, and many authors have proposed extensions to the basic approach. Wang *et al.* [160] confirmed putative matches in a geometric post-verification step. Filliant described a system where the visual vocabulary is learned online [45]. Ferreira *et al.* employed a Bernoulli mixture model to capture some of the conditional dependencies between words in the vocabulary [44]. Schindler *et al.* described how to tailor visual vocabulary generation so as to yield more discriminative visual words [135], and discuss the application of the technique to city-scale localization with a database of 30,000 images.

Finally, a recent paper proposed an interesting alternative to the local feature model, known as the *epitomic* representation [112]. The paper reports strong place recognition performance; however performance was compared only to the work of Torralba *et al.* [152]. Relative performance with respect to bag-of-words methods remains to be investigated.

2.3.2 Loop Closure Detection

The work discussed so far has been mainly concerned with the localization problem, where the map is known a-priori and the current sensory view is guaranteed to come from somewhere within the map. To use a similar appearance-based approach to detect loop closure in SLAM, the system must be able to deal with the possibility that the current view comes from a previously unvisited place and has no match within the map. As pointed out in [55], this makes the problem considerably more difficult. Particularly challenging is the *perceptual aliasing* problem — the fact that different parts of the workspace may appear the same to the robot’s sensors. As noted in [140], this can be a problem even when using rich sensors such as cameras due to repetitive features in the environment, for example mass-produced objects or repeating architectural elements. Gutmann and Konolige [55] tackle the problem by computing a similarity score between a patch of the map around the current pose and older areas of the map to identify possible loop-closures. They then employ a set of heuristics to decide if the putative match is a genuine loop closure or not. While these heuristic methods achieved some success, they did not provide a satisfactory solution to the perceptual aliasing problem. Goedemé described an approach to the issue using Dempster-Shafer theory, using sequences of observations to confirm or

reject putative loop closures [52]. The approach involved separate mapping and localization phases, so was not suitable for unsupervised mobile robotics applications.

A number of methods based on *similarity matrices* were proposed as a way to extend appearance-based matching beyond pure localization tasks. These methods define a similarity score between observations, then compute the pairwise similarity between all observations to yield a square similarity matrix. If the robot observes frequently as it moves through the environment and observations are ordered by time, then loop closures appear as off-diagonal stripes in this matrix. Levin and Szeliski describe such a method [81] that uses a cascade of similarity functions based on global colour histograms, image moments and epipolar geometry. Silpa-Anan and Hartley describe a similar system [140, 141] which employs SIFT features. Zivkovic *et al.* [170] consider the case where the order in which the images were collected is unknown, and use graph cuts to identify clusters of related images. A series of papers by Ho and Newman [111, 108, 59, 58, 60] advanced similarity matrix based techniques by considering the problem of perceptual aliasing. Their approach is based on a singular value decomposition of the similarity matrix that eliminates the effect of repetitive structure. They also address the issue of deciding if a putative match is indeed a loop closure based on examining an extreme value distribution related to the similarity matrix.

Similarity matrix based techniques were effective, however the only method which explicitly tackled the perceptual aliasing problem (Ho and Newman) was limited by $O(n^3)$ complexity. Further, the similarity matrix approaches were not truly probabilistic in construction, and so did not integrate very smoothly into a more general SLAM framework. In 2007 we outlined a new approach to appearance based navigation which addressed the perceptual aliasing problem in a probabilistic framework, and furthermore had $O(n)$ runtime complexity [30]. The details of this algorithm we will discuss in Chapter 3.

2.3.3 Recent Developments

While appearance-based navigation has a long history within robotics, there has been considerable development in the field within the timespan of this thesis. Appearance-based navigation and loop closure detection systems operating on trajectories on the order of a few kilometers in length are now

commonplace [2, 47, 20, 169, 85]. Indeed, place recognition systems similar in character to the one described in this thesis have become an important component even of single-camera SLAM systems designed for small-scale applications [42, 163].

Use of these systems on the scale of tens of kilometers or more has also begun to be feasible. For example Milford and Wyeth [99] described a large scale appearance-based navigation experiment using a system based on set of biologically inspired techniques. 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 [14], 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 [15].

A number of recent papers also describe integrated systems which tightly couple appearance and metric information. Olson, for example, 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 [115]. 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 [11]. More loosely coupled systems were also described in [71, 110].

Finally, authors have also begun to consider the problem of place recognition in the presence of major environmental variation — for example, initial results for cross-season place recognition were presented in [157].

Chapter 3

The FAB-MAP Model

This chapter introduces a new probabilistic model for navigation and mapping using appearance data, which we call Fast Appearance Based Mapping or FAB-MAP. The technique allows a mobile robot to compute a probability distribution over its location based on the sensory appearance of the local environment alone, without using any metric information. Navigation methods based on sensory similarity have been proposed before – this work goes beyond previously published methods in that it assesses not the similarity of two observations but rather the probability that they originate from the same location. This is achieved by learning a generative model of appearance, which enables us to assign identical but common observations a low probability of having come from the same place. The best previous solutions to the appearance-based navigation problem in visually repetitive environments had complexity cubic in the number of locations in the map [60]. The algorithm presented here has linear complexity, extending its applicability to much larger environments, and making it particularly suitable for online loop closure detection in mobile robotics.

The basic form of our system is inspired by the bag-of-words image retrieval systems developed in the computer vision community [144, 114]. However, we extend the approach by learning a generative model for the bag-of-words data. This generative model captures the fact that certain combinations of appearance words tend to co-occur, because they are generated by common objects in the environment. Our system effectively learns a model of these common objects (without supervision, see Figure 3.9),

which allows us to improve inference by reasoning about which sets of features are likely to appear or disappear together (see Figure 3.18). We present results which demonstrate that the system is capable of recognising places even when observations have few features in common, while rejecting false matches due to perceptual aliasing even when such observations share many features. Our approach also has reasonable computational cost – fast enough for online loop closure detection in realistic mobile robotics problems where the map contains several thousand places.

The focus in this chapter will be on the development and validation of the probabilistic model. We leave other issues relevant to improving system performance, such as geometric checks and details of visual word generation, for discussion in the context of the large scale experimental verification in Chapter 5.

3.1 Approximating High Dimensional Discrete Distributions

This section introduces some necessary background material on machine learning techniques used by our system. Readers familiar with Chow Liu trees may wish to skip to Section 3.2.

Consider a distribution $P(Z)$ on n discrete variables, $Z = \{z_1, z_2, \dots, z_n\}$. We wish to learn the parameters of this distribution from a set of samples. If $P(Z)$ is a general distribution without special structure, the space needed to represent the distribution is exponential in n — as n increases dealing with such distributions quickly becomes intractable. A solution to this problem is to approximate $P(Z)$ by another distribution $Q(Z)$ possessing some special structure that makes it tractable to work with, and that is in some sense similar to the original distribution. The natural similarity metric in this context is the Kullback-Leibler divergence [74], defined as:

$$D_{KL}(P, Q) = \sum_{x \in X} P(x) \log \frac{P(x)}{Q(x)} \quad (3.1)$$

where the summation is carried out over all possible states in the distribution. The KL divergence is zero when the distributions are identical, and strictly larger otherwise.

The Naive Bayes approximation is an example of a structural constraint that allows for tractable

learning of $Q(Z)$. Under this approximation $Q(Z)$ is restricted such that each variable must be independent of all others. This extreme structural constraint limits the degree to which $Q(Z)$ can represent $P(Z)$. A less severe constraint is to require each variable in $Q(Z)$ to be conditioned on at most one other variable — this restricts the graphical model of $Q(Z)$ to be a tree. Unlike the Naive Bayes case, where there is only one possible graphical model, in this case there are n^{n-2} possible tree-structured distributions that satisfy the structural constraint [162]. Chow and Liu [24] describe a polynomial time algorithm to select the best such distribution, which we elaborate on below.

More generally we could constrain $Q(Z)$ such that each variable is conditioned on at most n other variables. This structure is known as a *thin junction tree*. It was shown in [148] that the problem of determining the optimal thin junction tree is NP-hard, though there are several methods to find a good approximate solution (for example [150]).

3.1.1 Chow Liu Trees

The Chow Liu algorithm approximates a discrete distribution $P(Z)$ by the closest tree-structured Bayesian network $Q(Z)_{opt}$, in the sense of minimizing the Kullback-Leibler divergence. The structure of $Q(Z)_{opt}$ is determined by considering a mutual information graph $\mathcal{G}$. For a distribution over n variables, $\mathcal{G}$ is the complete graph with n nodes and $\frac{n(n-1)}{2}$ edges, where each edge (z_i, z_j) has weight equal to the mutual information $I(z_i, z_j)$ between variable i and j :

$$I(z_i, z_j) = \sum_{z_i \in \Omega, z_j \in \Omega} p(z_i, z_j) \log \frac{p(z_i, z_j)}{p(z_i)p(z_j)} \quad (3.2)$$

and the summation is carried out over all possible states of z . Mutual information measures the degree to which knowledge of the value of one variable predicts the value of another. It is zero if two variables are independent, and strictly larger otherwise.

Chow and Liu prove that the maximum-weight spanning tree [27] of the mutual information graph will have the same structure as $Q(Z)_{opt}$ (see Figure 3.1). Intuitively, the dependencies between variables that exist in $P(Z)$ but are excluded from $Q(Z)$ have as little mutual information as possible, and so are the best ones to approximate as independent. Chow and Liu further prove that the conditional

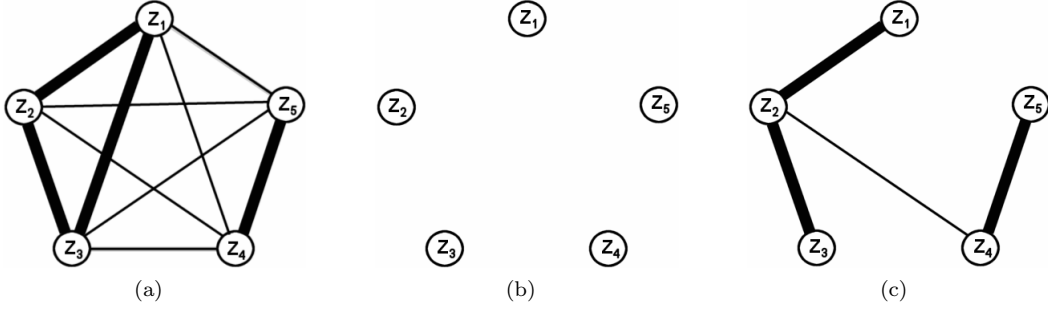


Figure 3.1: **(a)** Graphical model of the underlying distribution $P(Z)$. Mutual information between variables is shown by the thickness of the edge. **(b)** Naive Bayes approximation. **(c)** Chow Liu tree.

probabilities $p(z_i|z_j)$ for each edge in $Q(Z)_{opt}$ are the same as the conditional probabilities in $P(Z)$ – thus maximum likelihood estimates can be obtained directly from co-occurrence frequency in training data. To mitigate potential problems due to limited training data, we estimate these probabilities using a pseudo-Bayesian smoothing estimator described in Section 3.1.3 below, rather than the maximum likelihood estimator. This prevents any probabilities from having degenerate values of 0 or 1.

In the following section we will use Chow Liu trees to approximate large discrete distributions. Because we are interested in learning distributions over a large number of variables ($\sim 10,000$) the $O(n^2)$ mutual information graph to be computed is very large, typically too large to be stored in RAM. To deal with this, we use a semi-external spanning tree algorithm [39]. The mutual information graph is required only temporarily at learning time – it is discarded once the maximal spanning tree has been computed. Meilă [92] has described a more efficient, though complex, method of learning Chow Liu trees that avoids the explicit computation of the full mutual information graph, and provides significant speed-up for “sparse data” such as ours. In Chapter 5 we will work with even larger Chow Liu trees over $\sim 100,000$ variables, in which case use of Meilă’s algorithm will become essential.

We have chosen to use Chow Liu trees because they are tractable to compute even for very high-dimensional distributions, are guaranteed to be the optimal approximation within their model class, and require only first order conditional probabilities, which can be reliably estimated from available training data. However, we are essentially interested only in capturing dependencies between variables and there is nothing that restricts us to using a Chow Liu tree for this purpose. Our approach could

substitute a more complex model such as a mixture of trees [94], a thin junction tree of wider tree-width [150, 43], or with some modification make use of other classes of learning techniques, for example deep belief networks [130]. We will revisit these possibilities in Chapter 6.

A simple example to illustrate the performance of Chow Liu trees for density estimation is given in Appendix B.

3.1.2 Sample Complexity of Learning

When learning Bayesian networks from data, the question arises as to how much training data is required – this is known as the sample complexity of learning. The question can be asked separately for the problems of learning the *structure* of the network, and for the *parameters* of a network with known structure. The question will appear particularly relevant in Chapter 5 when we will consider learning Chow Liu trees over a 100,000 variables from only thousands of training examples.

There is much support for the practice of learning large networks from relatively small quantities of training data. Recent theoretical work by Abbeel [1] has shown that for bounded fan-in Bayesian networks (such as Chow Liu trees), the sample complexity of parameter learning is independent of the number of variables in the network. For the structure learning problem, sample complexity in the bounded fan-in case is known to be at worst logarithmic in the number of variables in the network, at least when the network structure is required to be only close to optimal [62]. Empirical results are in agreement with these predictions. In [94], Chow Liu trees were shown to be effective as classifiers even when available training data was limited relative to the dimensionality of the data. The results of [22] show similar behaviour when using Chow Liu trees as density estimators. Notably, they also demonstrate that in the regime where training data is limited relative to problem dimensionality, the Chow Liu algorithm consistently out-performs models which attempt to learn more complex large-tree-width Bayesian networks.

3.1.3 Smoothing Probability Estimates

Given the structure of the Bayesian network, the relevant conditional probabilities that specify the distribution must also be learned from the training data. When training data is limited, the maximum

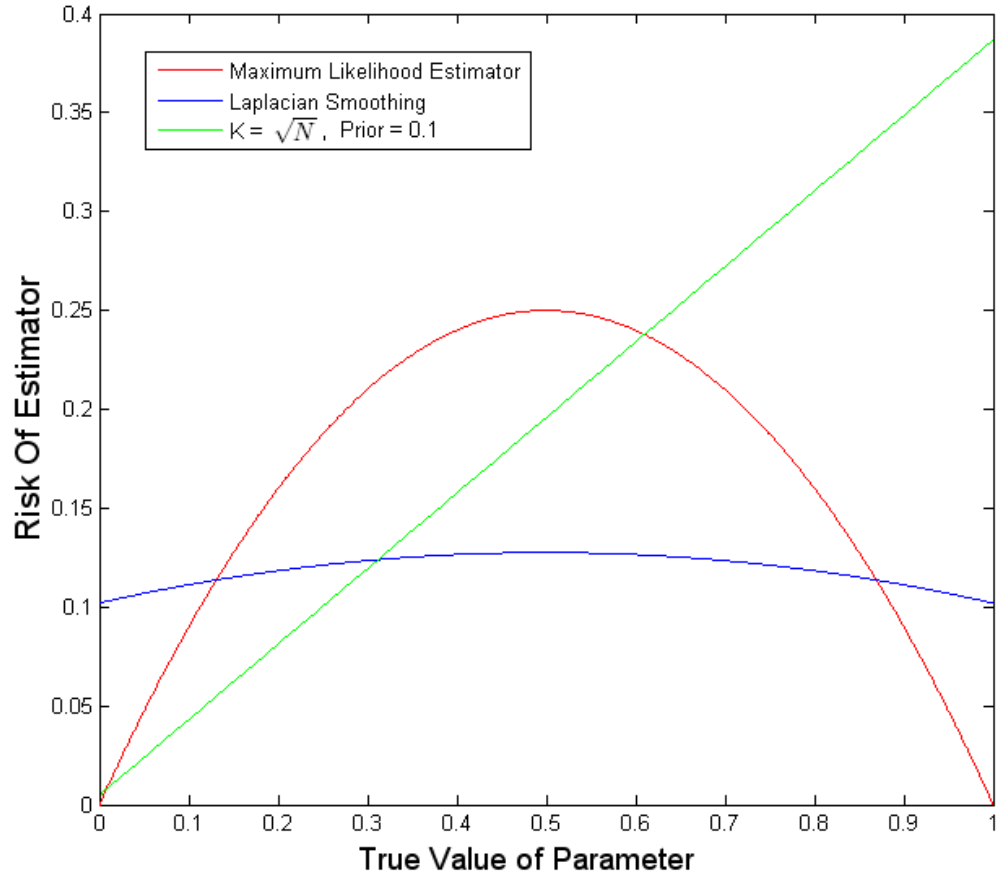


Figure 3.2: Risk function of different estimators, for a binary random variable estimated from five observations. The horizontal axis is the true probability of the random variable being observed, and the vertical axis is the risk of the estimator. The maximum likelihood estimator is shown in red, Laplacian smoothing in blue, and an estimator with $K = \sqrt{N}$ and $p_{prior} = 0.1$ in green.

likelihood estimate of some probabilities $p(z_i = s_i | z_j = s_j)$ can be 0 or 1, due to sampling error. To avoid this effect, the probability estimates must be smoothed. A variety smoothing techniques exist [34], typically of the form

$$p_{smooth} = (\frac{N}{N+K})p_{mle} + (\frac{K}{N+K})p_{prior} \quad (3.3)$$

where p_{mle} is the maximum likelihood estimate of the parameter, p_{prior} is a prior, N is the number of observations and K determines the relative weighting between the prior and the maximum likelihood estimate. For example, choosing $K = 2$ and $p_{prior} = 0.5$ gives Laplacian smoothing. We use the estimator described in [10], where $K = \sqrt{N}$ and the prior for $p(A|B)$ is set to $P(A)$.

The quality of such an estimator may be characterized by its risk function

$$R(\hat{p}, p) = E \left(\|\hat{p} - p\|^2 \right) \quad (3.4)$$

which is the expectation of the squared distance between the estimate $\hat{p}$ and the true value p . The risk functions for several different estimators are illustrated in Figure 3.2.

3.1.4 What the Chow Liu Tree Learns

To conclude this section, it is useful to build a little intuition about what the Chow Liu tree is capturing, and how it will be useful to an appearance-based SLAM system. The model we will employ in the rest of this chapter takes as input sets of visual words, which represent the content of images captured by the robot. When it comes to assessing the similarity of two of these bag-of-words vectors, a simple approach would be to compute some standard distance metric between them such as L2 distance or cosine distance. The commonly used tf-idf ranking measure essentially takes this approach, with some weighting applied to the dimensions. However, these simple metrics assume independence between visual words, which is far from the case. Real-world objects such as cars and pedestrians typically give rise to many visual word detections. Thus two images of a location which differ by the presence of some such object can lie far apart in visual word space, even though they are semantically close. By capturing some of the correlations between sets of visual words using the Chow Liu tree, we will arrive

at a similarity function that is sensitive to the fact that the real world is composed of objects rather than independent visual elements. The learned similarity function is thus in some sense closer to the underlying semantic distance between images. Typical correlations learned by the model are shown in Figure 3.9, and some examples of improved inference performance are shown in Figure 3.18.

3.2 Probabilistic Navigation using Appearance

We now describe the construction of a probabilistic framework for appearance-based navigation. In overview, the world is modeled as a set of discrete locations, each location being described by a distribution over appearance words. Incoming sensory data is converted into a bag-of-words representation; for each location, we can then ask how likely it is that the observation came from that location’s distribution. We also find an expression for the probability that the observation came from a place not in the map. This yields a pdf over location, which we can use to make a data association decision and update our belief about the appearance of each place. Essentially this is a SLAM algorithm in a discrete world. The method is outlined in detail in the following sections.

3.2.1 Representing Appearance

As previously mentioned, raw sensor data received by the system is converted into bag-of-words representation. In this representation, data is described by the presence or absence of “appearance words” chosen from a vocabulary of size $|v|$ (see discussion in Section 2.2). An observation of local scene appearance, visual or otherwise, captured at time k is denoted $Z_k = \{z_1, \dots, z_{|v|}\}$, where z_q is a binary variable indicating the presence (or absence) of the q^{th} word of the vocabulary. Furthermore $\mathcal{Z}^k$ is used to denote the set of all observations up to time k . Note that while we will use the term “bag-of-words” to refer to this representation, strictly this is a “set-of-words” approach which discards count information about the number of instances of an appearance word in an observation. We choose this representation because it decreases the complexity of learning and inference. We will discuss generalizations to the bag-of-words case in Chapter 6.

The results in this thesis employ binary features derived from imagery, based on quantized SURF

descriptors (see Section 3.3); however binary features from any sensor or combination of sensors could be used. For example, laser or sonar data could be represented using quantized versions of the features proposed in [51, 104, 66].

3.2.2 Representing Locations

At time k , our map of the environment is a collection of n_k discrete and disjoint locations $\mathcal{L}^k = \{L_1, \dots, L_{n_k}\}$. 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:

$$L_i : \{p(e_1 = 1|L_i), \dots, p(e_{|v|} = 1|L_i)\} \quad (3.5)$$

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

$$\mathcal{D} : \begin{cases} p(z_q = 1|e_q = 0), & \text{false positive probability.} \\ p(z_q = 0|e_q = 1), & \text{false negative probability.} \end{cases} \quad (3.6)$$

The generative model, which we will refer to as the FAB-MAP model, is illustrated in Figure 3.3.

The reason for introducing the hidden scene element variables e_q is twofold. Firstly, it provides a natural framework for incorporating data from multiple sensors with different (and possibly time-varying) error characteristics. Secondly, as outlined in the next section, it facilitates factoring the distribution $p(Z|L_i)$ into two parts — a simple model for each location composed of independent variables e_q , and a more complex model that captures the correlations between detections of appearance words $p(z_q|Z_k)$. Because the location model has a simple structure, it can be estimated online from the few observations available for any given location (combined with a prior). The more complex model that captures the correlations between observations is learned offline from abundant training data, and can be combined with the simple location models on the assumption that the conditional

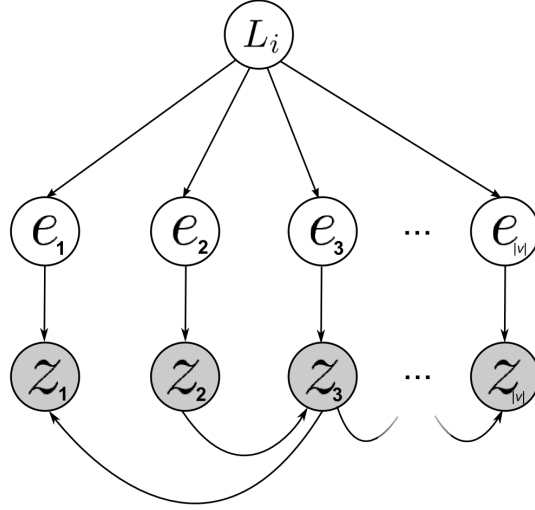


Figure 3.3: The FAB-MAP generative model. Observed variables are shaded, latent variables unshaded. The environment generates locations L_i . Locations independently generate scene elements e_q . Scene elements generate word observations z_q , which have tree-structured dependencies between them. Some additional independence assumptions are noted in Section 3.2.5.

dependencies between appearance words are independent of location. We describe how this is achieved in the following section. We will discuss alternative generative models in Section 3.2.6 where their relative merits may be more easily understood.

3.2.3 Inference in the FAB-MAP Model

We now describe how localization and mapping is achieved using our model. The algorithm we present is a recursive Bayes filter. We must begin our explanation somewhere within the recursive loop – so assume for the moment that a partial map of the environment has already been created. The robot receives a new observation, and we wish to calculate the probability of being at each of the locations in the map, given all the observations collected so far. Calculating this probability, $p(L_i|\mathcal{Z}^k)$ for each location L_i , can be formulated as a recursive Bayes estimation problem:

$$p(L_i|\mathcal{Z}^k) = \frac{p(Z_k|L_i, \mathcal{Z}^{k-1})p(L_i|\mathcal{Z}^{k-1})}{p(Z_k|\mathcal{Z}^{k-1})} \quad (3.7)$$

Here $p(L_i|\mathcal{Z}^{k-1})$ is our prior belief about our location, $p(Z_k|L_i, \mathcal{Z}^{k-1})$ is the observation likelihood, and $p(Z_k|\mathcal{Z}^{k-1})$ is a normalizing term. We discuss the evaluation of each of these terms below, with reference to the graphical model in Figure 3.3. We will also make some additional independence assumptions which are introduced in the course of the derivation. These additional assumptions are summarized in Section 3.2.5. Overview pseudocode is given in Algorithms 3.1 and 3.2.

3.2.4 Evaluating The Model

Observation Likelihood

To simplify evaluation of the observation likelihood $p(Z_k|L_i, \mathcal{Z}^{k-1})$, we first assume independence between the current and past observations, conditioned on the location. The simplified observation likelihood $p(Z_k|L_i)$ can then be expanded as:

$$p(Z_k|L_i) = p(z_n|z_1, z_2, \dots, z_{n-1}, L_i)p(z_{n-1}|z_1, z_2, \dots, z_{n-2}, L_i) \dots p(z_2|z_1, L_i)p(z_1|L_i) \quad (3.8)$$

or the equivalent expansion for any other variable ordering. This expression cannot be evaluated directly because of the intractability of learning the high-order conditional dependencies between appearance words. The simplest approximation is to make a Naive Bayes assumption, yielding:

$$p(Z_k|L_i) \approx \prod_{q=1}^{|v|} p(z_q|L_i) \quad (3.9)$$

Referring to the graphical model in Figure 3.3, each factor can be further expanded as:

$$p(z_q|L_i) = \sum_{s \in \{0,1\}} p(z_q|e_q = s, L_i)p(e_q = s|L_i) \quad (3.10)$$

Now, making the assumption that detector errors are independent of position in the world (i.e. $p(z_q|e_q, L_i) = p(z_q|e_q)$), this becomes:

$$p(z_q|L_i) = \sum_{s \in \{0,1\}} p(z_q|e_q = s)p(e_q = s|L_i) \quad (3.11)$$

which can now be evaluated directly — $p(z_q|e_q = s)$ are the components of the detector model specified in Equation 3.6 and $p(e_q = s|L_i)$ is a component of the place appearance model.

While a Naive Bayes approach yields reasonable results (see Section 3.3), substantially better results can be obtained by instead employing a Chow-Liu approximation to capture more of the dependencies between appearance words. Using a Chow Liu assumption, the appearance likelihood of Equation 3.8 can be approximated as

$$p(Z_k|L_i) \approx p(z_r|L_i) \prod_{q=2}^{|v|} p(z_q|z_{p_q}, L_i) \quad (3.12)$$

where z_r is the root of the tree and z_{p_q} is the parent of z_q in the tree. Contrasting Equation 3.12 with Equation 3.9, note that the Chow Liu model takes account of some of the dependencies between word observations z_q . As before, the observation factors in Equation 3.12 can be further expanded as:

$$p(z_q | z_{p_q}, L_i) = \sum_{s_{e_q} \in \{0,1\}} p(z_q | e_q = s_{e_q}, z_{p_q}, L_i) p(e_q = s_{e_q} | z_{p_q}, L_i) \quad (3.13)$$

Again making the assumption that detector errors are independent of position in the world (i.e. $p(z_q|e_q, L_i) = p(z_q|e_q)$), the expression becomes:

$$p(z_q | z_{p_q}, L_i) = \sum_{s_{e_q} \in \{0,1\}} p(z_q | e_q = s_{e_q}, z_{p_q}) p(e_q = s_{e_q} | z_{p_q}, L_i) \quad (3.14)$$

We now make the approximation that $p(e_q = s_{e_q} | z_{p_q}, L_i)$ can be replaced by $p(e_q = s_{e_q} | L_i)$ (We make this approximation for tractability. See discussion in Section 3.2.5), yielding:

$$p(z_q | z_{p_q}, L_i) = \sum_{s_{e_q} \in \{0,1\}} p(z_q | e_q = s_{e_q}, z_{p_q}) p(e_q = s_{e_q} | L_i) \quad (3.15)$$

The term $p(z_q | e_q, z_{p_q})$ can be expanded (see Appendix C.1) as:

$$p(z_q = s_{z_q} | e_q = s_{e_q}, z_p = s_{z_p}) = (1 + \frac{\alpha}{\beta})^{-1} \quad (3.16)$$

where $s_{z_q}, s_{e_q}, s_{z_p} \in \{0, 1\}$ and

$$\alpha = p(z_q = s_{z_q})p(z_q = \overline{s_{z_q}} \mid e_q = s_{e_q})p(z_q = \overline{s_{z_q}} \mid z_p = s_{z_p}) \quad (3.17)$$

$$\beta = \underbrace{p(z_q = \overline{s_{z_q}})}_{\text{prior}} \underbrace{p(z_q = s_{z_q} \mid e_q = s_{e_q})}_{\text{detector model}} \underbrace{p(z_q = s_{z_q} \mid z_p = s_{z_p})}_{\text{conditional from training}} \quad (3.18)$$

where $\overline{s_z}$ denotes the opposite state to s_z . The terms α and β are now expressed entirely in terms of quantities which can be estimated from training data (as indicated by the under-braces). Hence $p(Z_k \mid L_i)$ can now be computed directly.

New Place or Old Place?

We now turn our attention to the denominator of Equation 2.1, $p(Z_k \mid \mathcal{Z}^{k-1})$. Knowing $p(Z_k \mid L)$ as calculated in the previous section identifies the most likely location in the map to have generated the current observation. However, we would like to be able to deal with the possibility that the observation comes from a new location, not currently in the map. This requires the knowledge of the partition function $p(Z_k \mid \mathcal{Z}^{k-1})$, which converts the appearance likelihood into a probability of loop closure. Computation of the partition function is the key difference between a localization system and a true appearance-only SLAM system. It can be observed that $p(Z_k \mid \mathcal{Z}^{k-1})$ 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

$$p(Z_k \mid \mathcal{Z}^{k-1}) = \sum_{m \in \mathcal{L}^k} p(Z_k \mid L_m)p(L_m \mid \mathcal{Z}^{k-1}) + \sum_{u \in \overline{\mathcal{L}^k}} p(Z_k \mid L_u)p(L_u \mid \mathcal{Z}^{k-1}) \quad (3.19)$$

where we have applied our assumption that observations are conditionally independent given location. The second summation cannot be evaluated directly because it involves all possible unknown locations. We have compared two different approximations to this term. The first is a mean field approximation [67] :

$$p(Z_k|\mathcal{Z}^{k-1}) \approx \sum_{m \in \mathcal{L}^k} p(Z_k|L_m)p(L_m|\mathcal{Z}^{k-1}) + p(Z_k|L_{avg}) \sum_{u \in \overline{\mathcal{L}^k}} p(L_u|\mathcal{Z}^{k-1}) \quad (3.20)$$

where L_{avg} is the “average place” where the e_q values are set to their marginal probability, and $\sum_{u \in \overline{\mathcal{L}^k}} p(L_u|\mathcal{Z}^{k-1})$ is the prior probability that we are at a new location, which can be obtained from our previous position estimate and a motion model (discussed in the next section).

An alternative to the mean field approach is to approximate the second summation via sampling. The procedure is to sample location models L_u according to the distribution by which they are generated by the environment, and evaluate $\sum_{u \in \overline{\mathcal{L}^k}} p(Z_k|L_u)p(L_u|\mathcal{Z}^{k-1})$ for the sampled location models. In order to do this, some method of sampling location models L_u is required. Here we make an approximation for ease of implementation — we instead sample an observation Z and use the sampled observation to create a place model. The reason for doing this is that sampling an observation Z is extremely easy — our sampler simply consists of selecting at random from a large collection of observations. For our robot which navigates in outdoor urban environments using imagery, large collections of street-side imagery for sampling are readily available – for example from previous runs of the robot, or from sources such as Google Street View¹. In general this sampling procedure will not create location models according to their true distribution because models may have been created from multiple observations of the location. However, it will be a good approximation when the robot is exploring a new environment, where most location models will have only a single observation associated with them. In practice we have found that it works very well, and it has the advantage of being very easy to implement.

Having sampled a location model, we must evaluate $p(Z_k|L_u)p(L_u|\mathcal{Z}^{k-1})$ for the sample. Calculating $p(Z_k|L_u)$ has already been described. For convenience we assume $p(L_u|\mathcal{Z}^{k-1})$, the prior probability of our sampled place model with respect to our history of observations, to be uniform over the samples. Equation 3.19 thus becomes

$$p(Z_k|\mathcal{Z}^{k-1}) \approx \sum_{m \in \mathcal{L}^k} p(Z_k|L_m)p(L_m|\mathcal{Z}^{k-1}) + p(L_{new}|\mathcal{Z}^{k-1}) \sum_{u=1}^{n_s} \frac{p(Z_k|L_u)}{n_s} \quad (3.21)$$

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

¹<http://www.google.com/help/maps/streetview/>

place.

The approximation could be improved further by using a non-uniform prior $p(L_u|\mathcal{Z}^{k-1})$ over the samples. For example, if robot has recently seen a sequence of images from an urban area, we would expect the next observation to be similar, rather than of a forest, say. This kind of conditional sampling can easily be achieved by maintaining a “location estimate” over the set of images used for Monte Carlo sampling in the same way that we maintain a location estimate over the map, provided that the sampling set consists of sequentially collected images (i.e. that samples u and $u + 1$ were collected at physically close locations). In practice we have found it more convenient to use random collections of images for the sampling set, so have not taken advantage of this possibility.

To conclude, let us try to build some intuition about how the calculation of the partition function relates to perceptual aliasing. Consider what happens after the system has collected a new observation which is a good match for one of the locations in the map. To evaluate $p(Z_k|\mathcal{Z}^{k-1})$ we compute the likelihood that the observation came from randomly selected locations in the sampling set. If the similarity between the current observation and the location in the map is caused by some repetitive feature such as brickwork or foliage, it is likely that we will find an equally good match for our observation among the randomly sampled locations. This means that $p(Z_k|\mathcal{Z}^{k-1})$ will be comparable in magnitude to $p(Z_k|L_i)$ (the likelihood of the matched location), and so the probability assigned to loop closure will be low. On the other hand, finding a good match in the sampling set is very unlikely in the case of a distinctive loop closure. Thus the partition function serves to suppress spurious loop closures caused by perceptual aliasing. The procedure can break down when the robot first encounters a repetitive feature which is not part of its sampling set (e.g. Figure 3.19), however in the limit of sampling set size it will correctly handle progressively rarer causes of false loop closure.

Location Prior

The last term to discuss is the location prior $p(L_i|\mathcal{Z}^{k-1})$. This prior transmits information such that a sequence of observations can result in a higher confidence loop closure. The most straight forward way to obtain a prior is to transform the previous position estimate via a motion model, and use this as our prior for location at the next time step. For the purpose of calculating a prior we assume that

sequentially collected observations come from adjacent places, so that if the robot is at place i at time t , it is likely to be at one of the places $\{i - 1, i + 1\}$ at time $t + 1$, with equal probability. For places with unknown neighbours (e.g. the next place after the most recently collected observation, where place $i + 1$ may be a new place not already in the map), part of the probability mass is assigned to a “new place” node and the remainder is spread evenly over all places in the map. The split is governed by the probability that a topological link with unknown endpoint leads to a new place, which is a user-defined parameter in our system. Clearly this prior could be improved via a better topology estimate or through the use of some odometry information — however, we have found that the effect of the prior is relatively weak, so these issues are not critical to performance.

If the assumption that sequential observations come from adjacent places is not valid, the prior can be simply left uniform. While this results in some increase in the number of false matches, performance is still largely sufficient, especially over smaller trajectories (see quantities evaluation in Chapter 5). Alternatively, if we have an estimate for the topology of the environment, but no estimate of our position within this topology, we could obtain a prior as the limit of a random walk on the topology. This can be obtained efficiently as the dominant eigenvector of the normalized link matrix of the topology [117].

Dealing with Outliers

We have found that the performance of the inference procedure outlined above is strongly dependent on an accurate estimation of the denominator term in Equation 2.1, $p(Z_k | \mathcal{Z}^{k-1})$. Ideally we would like to perform the Monte Carlo integration in Equation 3.21 over a large set of place models L_u that fully capture the visual variety of the environment. In practice, we are limited by running time and available data. The consequence is that occasionally we will incorrectly assign two similar images high probability of having come from the same place, when in fact the similarity is due to perceptual aliasing.

An example of this is illustrated in Figure 3.19, where two images from different places, both of which contain iron railings, are incorrectly assigned high probability of having come from the same place. The reason for this mistake is that the railings generate a large number of features, and there were no examples of railings in the sampling set used to calculate $p(Z_k | \mathcal{Z}^{k-1})$, so the system cannot know these sets of features are correlated.

In practice, due to limited data and computation time, there will always be some aspects of repetitive environmental structure that we fail to capture. We thus employ some secondary techniques to make the system more robust to these occasional spurious matches. One of the most effective approaches is to check a putative loop closure for consistent epipolar geometry [57]. However, cases still exist where this technique may fail, particularly due to repetitive man-made objects such as the situation illustrated in Figure 3.19 (b). A complementary approach that we have found to be successful is to apply a smoothing operation to our likelihood estimates:

$$p(Z_k|L_i) \longrightarrow \sigma p(Z_k|L_i) + \frac{(1 - \sigma)}{n_k} \quad (3.22)$$

where n_k is the number of places in the map and σ is the smoothing parameter, which for our experiments we set to 0.99.

The effect of this smoothing of the data likelihood values is to prevent the system asserting loop closure with high confidence on the basis of a single similar image pair. Smoothing the likelihood values effectively boosts the importance of the prior term $p(L_i|\mathcal{Z}^{k-1})$, so that high probability of loop closure can now only be achieved by accumulating evidence from a sequence of matched observations. While in principle mistakes due to our approximate calculation of $p(Z_k|\mathcal{Z}^{k-1})$ can still occur, in practice we have found that this modification successfully rejects almost all outliers, while still allowing true loop closures to achieve high probability after a sequence of only two or three corresponding images.

Updating Place Models

We have now outlined how to compute a pdf over location given a new observation. The final issue to address is data association – at some point it is necessary to make a hard decision and associate the observation with a particular location. The rule used to make data association decisions is discussed in the next section. Once the data association decision has been taken, we can update the relevant place appearance model, which consists of the set $\{p(e_1 = 1|L_i), \dots, p(e_{|v|} = 1|L_i)\}$. When a new place is created its appearance model is initialized so that all scene elements exist with some prior probability $p(e_q = 1)$, which we derive from the word frequencies in training data. Note that this prior probability

is different for each word. Given an observation that relates to the place, each component of the appearance model can be updated according to:

$$p(e_q = 1|L_i, \mathcal{Z}^k) = \frac{p(Z_k|e_q = 1, L_i)p(e_q = 1|L_i, \mathcal{Z}^{k-1})}{p(Z_k|L_i)} \quad (3.23)$$

where we have again applied our assumption that observations are conditionally independent given location.

Data Association Rule

To decide data association we apply a fixed decision rule: if the current observation matches a location with probability higher than some user specified threshold (e.g. $p > 0.999$), we associate the observation with that location. If no location (including the “new place”) meets the probability threshold, we default to creating a new location. Note that the threshold used for triggering data association need not be the same as that at which the system outputs a loop closure notification. A decision rule of this form is preferable to doing maximum-likelihood data association because of the asymmetric consequences of incorrect data association decisions. Reminiscent of the situation with metric SLAM algorithms, we find that incorrectly creating a new location has only minor consequences, whereas incorrectly associating an observation with an existing location rapidly leads to failure of the filter.

Our intuition for this phenomenon is as follows. When an incorrect data association decision is made, the location whose appearance model is updated will now have two very different observations associated with it. The appearance model which results from this mixture of visually different observations tends towards the appearance of the “average place”, which is a reasonably good match for many observations. New observations collected by the robot which are not a good match for any location in the sampling set used in Equation 3.21 and have no good match within the map will tend to match best with this average location. The result is that further observations will be incorrectly associated with the location, making its appearance closer to average and reinforcing the effect. A single wrong data association decision can thus rapidly break the filter.

In contrast, the effect of incorrectly creating a new location is relatively minor. The result is that

Algorithm 3.1 Overview pseudocode for FAB-MAP. Supplementary functions are shown in Algorithm 3.2.

```

//This function processes each new observation collected by the vehicle.
// $\mathcal{L}^k$  is the current map, and  $\mathcal{Z}^k$  is the set of observations up to time  $k$ .
//The most recent observation in  $\mathcal{Z}^k$  is  $Z_k$ , and the  $i^{th}$  loca-
tion in the map is  $L_i$ .
Process_Observation( $\mathcal{L}^k, Z_k$ ):
    pdf = Compute_Pdf_Over_Location( $\mathcal{L}^k, Z_k$ )
    Decide_Data_Association(pdf)
     $\mathcal{L}^{k+1}$  = Initialize_New_Place_Or_Update_Existing_Place( $\mathcal{L}^k, Z_k$ )

```

a single location in the world will be represented by two distinct location models in the robot’s map. In principle this could prevent the robot from recognising this location in future because, given an observation from this location, the probability mass could be split evenly between the two location models in the map, so that neither reaches the threshold necessary to declare loop closure. In practice we find that the symmetry is rarely so exact – typically most of the probability mass will be assigned to one or the other of the locations in the map, and loop closure will still be detected. In fact, we find that it is possible to run the system while *never* performing data association, instead creating a new location model for every new observation. While this mode of operation will certainly lead to a gradual degradation in system performance when the robot remains in the same area for a long time, it is not subject to the unpredictable catastrophic failure that results from a bad data association decision. Therefore, in our typical operation of the system, and in the results presented in this thesis, we opt for this mode of operation.

Clearly the approach we have taken here is not ideal (although it is effective). A better approach might be to maintain a pdf over topologies using a particle filter, as described in [129]. In that way, data association decisions could effectively be postponed until more evidence is available to confirm or reject them. Note that this approach is only effective in conjunction with vehicle motion information.

3.2.5 Independence Assumptions, Approximations and Input Parameters

We briefly recap the approximations and user-defined terms in our model. For tractability, we have made a number of independence assumptions:

Algorithm 3.2 Pseudocode for functions involved in processing a new observation.

```
Compute_Pdf_Over_Location( $\mathcal{L}^k, Z_k$ ):  
    //First, compute the likelihood that each location in the map  
    //generated the current observation.  
    for  $L_i$  in  $\mathcal{L}^k$  do:  
        likelihood[ $L_i$ ] = (Compute as per Equation 3.12)  
        //(for numerical precision we actually work with log-likelihoods)  
    //Also compute the likelihood that the observation came from  
    //a location not in the current map (" $L_{new}$ ")  
    likelihood[ $L_{new}$ ] = (Compute as per Equation 3.19)  
    //Mix in the prior  
    location_prior = Get_Location_Prior()  
    for  $i$  in range(likelihood) do:  
        pdf[i] = likelihood[i] * location_prior[i]  
    //Finally normalize and optionally also apply  
    //smoothing as described in Equation 3.22)  
    return pdf  
  
Get_Location_Prior()  
    //Transform the last pdf over location through a motion model.  
    //See discussion under "Location Prior" in Section 3.2.4  
    //The prior transmits information such that a sequence of  
    //observations yields a higher confidence loop closure.  
    return Apply_Motion_Model(pdf_over_location_at_last_timestep)  
  
Decide_Data_Association(pdf):  
    //Decide if any location meets the data association criteria  
    //See discussion under "Data Association Rule" in Section 3.2.4  
    //Note that results reported in this thesis use a rule  
    //that creates a new place for each new observation.  
  
Initialize_New_Place_Or_Update_Existing_Place( $\mathcal{L}^k, Z_k$ ):  
    //Given the data association decision, update the map.  
    if(data association decision for  $Z_k$  is to a location already in the map)  
        //Update the relevant location model ( $L_{da}$ )  
        //according to Equation 3.23  
         $L_{da}$  = Update_Place_Model( $L_{da}, Z_k$ )  
         $\mathcal{L}^{k+1} = \mathcal{L}^k$   
        //No new locations were added to the map,  
        //(but one location in the map was updated).  
    else  
        //Create a new location  $L_{n_k+1}$   
         $L_{n_k+1}$  = (A new location model, where all  
            scene elements exist with prior probability).  
        //Update the location model according to Equation 3.23  
         $L_{n_k+1}$  = Update_Place_Model( $L_{n_k+1}, Z_k$ )  
        //Add the new location to the map  
         $\mathcal{L}^{k+1} = \mathcal{L}^k \cup L_{n_k+1}$   
         $n_k = n_k + 1$   
    return  $\mathcal{L}^{k+1}$ 
```


1. Sets of observations are conditionally independent given position: $p(Z_k|L_i, \mathcal{Z}^{k-1}) = p(Z_k|L_i)$.
2. Detector behaviour is independent of position: $p(z_q|e_q, L_i) = p(z_q|e_q)$.
3. Location models are generated independently by the environment.

In addition to these independence assumptions, we also make an approximation for tractability:

1. The existence of scene element e_q is not informed by observations other than z_q . That is, an observation of $z_p \forall p \neq q$ provides no information about the existence of e_q , so that $p(e_q|Z_k) \approx p(e_q|z_{q_k})$. We also make the related approximation that $p(e_q | z_p, L_i)$ can be approximated by $p(e_q | L_i)$, when $p \neq q$. While strictly this does discard some information that could be recovered at the cost of more complex inference, in effect these approximations only mean that we will be slightly less certain about the existence of scene elements than if we made full use of the available data.

Input Parameters

The only user-specified inputs to the algorithm are the detector model, $p(z_q = 1|e_q = 0)$ and $p(z_q = 0|e_q = 1)$ (two scalars), the smoothing parameter σ and a term that sets the prior probability that a topological link with an unknown endpoint leads to a new place. Of these, the algorithm is particularly sensitive only to the detector model, which can be determined from a calibration discussed in Section 3.3.3.

3.2.6 Discussion: Alternative Generative Models

Finally, before evaluating the performance of the system, we return briefly to our choice of generative model outlined in Section 3.2.2. Several alternative generative models can be imagined, some of which are more intuitive, and it is useful to discuss the reasons for our model choice.

Model A1

Perhaps the most intuitive generative model, illustrated in Figure 3.4 and which we refer to as “Model A1”, eliminates the hidden layer of e variables and maintains $p(Z|L_i)$ directly. Likelihood evaluations

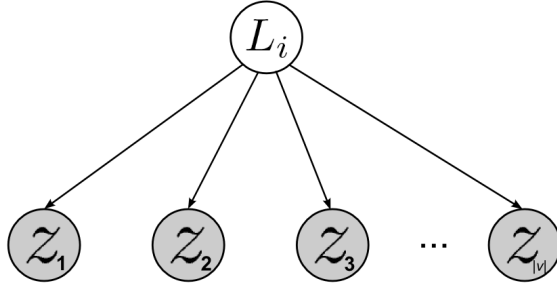


Figure 3.4: Model A1.

in the model would be of the form

$$p(Z_k|L_i) = \prod_{q=1}^{|v|} p(z_q|L_i)$$

In this model, which assumes independence between each appearance word, $p(z_q|L_i)$ could be maintained as a beta distribution initialized to some prior and whose parameters could be updated each time a new observation is associated with the place. The difficulty with this approach is that it implicitly assumes that $p(z_q|L_i)$ is a fixed process, and so does not provide a principled way to deal with observations of varying quality. By contrast the FAB-MAP model, by separating existence of underlying scene elements, e_q , from observations, z_q , can naturally deal with variable confidence observations by adjusting the detector terms in Equation 3.6. This provides a framework for dealing with sensor performance that varies with time, for example due to adverse lighting conditions or environmental effects such as fog, or for combining observations from sensors of different quality, for example cameras with different resolution or noise characteristics. While this thesis will not exploit this flexibility, it is nevertheless a more natural way to parameterize the generative model. Additionally, introducing the layer of e variables is crucial for efficient treatment of correlations between appearance words, as we will see in the discussion of Model A2.

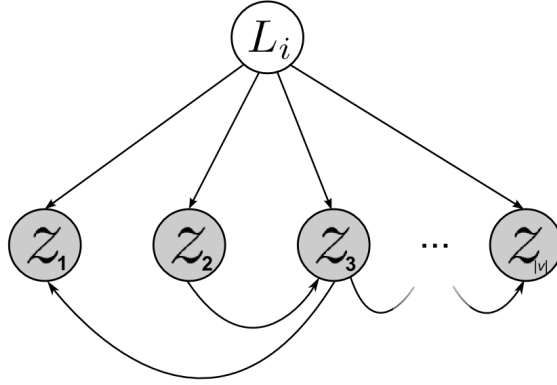


Figure 3.5: Model A2.

Model A2

Model A2 extends Model A1 by introducing tree-structured dependencies between the observed variables z . As in A1, the scheme maintains an estimate of $p(Z|L_i)$ directly for each location. Likelihood evaluations in the model would be of the form

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

where $p(z_q|z_{p_q}, L_i)$ would be explicitly maintained for each location. This is the TAN model discussed in [50].

The difficulty with this model is twofold. Firstly, due to the elimination of the e variables, it suffers from the same lack of flexibility as Model A1 in dealing with variable confidence observations. Secondly, and more pertinently, because in this model $p(z_q|z_{p_q}, L_i)$ cannot be factored but rather is required directly (cf. Equation 3.13), a separate Chow Liu tree must be constructed for each location in the map². This is in contrast to the FAB-MAP model where only a single global Chow Liu tree is required.

Learning separate Chow Liu trees for each location is not a desirable approach for a number of reasons. The core problem is that most locations in the map have very few associated observations (typically only one). Thus we could not hope to learn a separate Chow Liu model for each location

²It should be noted that the trees could differ in their parameters only, rather than in their structure.

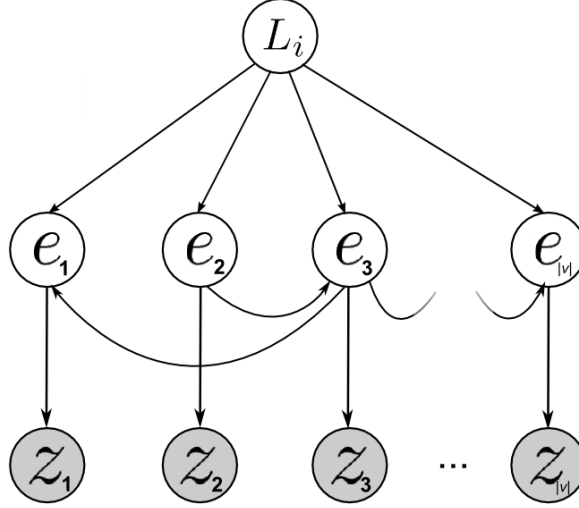


Figure 3.6: Model B.

from such limited training data, at least not by using the ML learning algorithm of [24]. One possibility would be to use the MAP Chow Liu learning technique of [94, 93], which makes use of a prior over Chow Liu trees. Interestingly, though the space of Chow Liu trees is super-exponential (n^{n-2}), a decomposable prior can be specified in only $O(n^2)$ space. The prior would be learned offline in much the same way as in the FAB-MAP model, and used to initialize each new location. The difficulty with this scheme is practical — maintaining the $O(n^2)$ prior for large numbers of variables requires enormous storage, whereas the FAB-MAP model makes use only of the final Chow Liu tree which requires only $O(n)$ space. Considering for example the 100,000 word visual vocabulary used in Chapter 5, maintaining the prior over Chow Liu trees would require approximately 80 GB of storage, whereas the Chow Liu tree itself requires only about 800 KB. For this reason, and because it also eliminates the flexibility of the detector, we discard model A2.

Model B

Finally we consider Model B, which has the same form as the FAB-MAP model, but imposes the tree-structured dependencies between the hidden variables e rather than the observed variables z . In some ways this is a more intuitive model than FAB-MAP, because the dependencies are imposed

between word existence rather than word observation. However, it shares with Model A2 the problem that the Chow Liu dependencies between variables may be different for each location, and so are much more expensive to represent. Additionally, dependencies between hidden variables e are less straight-forward to learn than those between observed variables z .

Thus we arrive back at the FAB-MAP model shown in Figure 3.3. The hidden variables e provide a natural way to deal with variable confidence observations, and also allow the likelihood $P(Z_k|L_i)$ to be factored so that it depends on a simple model for the locations (captured by $p(e_q|L_i)$) and a more complex but compactly representable model of the co-occurrence of visual words, $p(z_q|z_{p_q})$, which applies globally to all locations.

3.3 Implementing the Model

This section discusses some implementation details of the system: how images are converted to bag-of-words representation, how the Chow Liu tree is learned, and how the free parameters of the model are chosen.

3.3.1 Converting Images to Bag-of-Words Representation

Data must be supplied to the system in the bag-of-words representation. There are many possible ways to convert input sensory data to a bag-of-words. As described in Section 2.2.4, our system uses the SURF detector/descriptor [7] to extract regions of interest from the images, and compute 128D non-rotation-invariant descriptors for these regions. These descriptors are quantized against a visual vocabulary using a randomized forest of eight kd-trees [142, 105]. The visual vocabulary is learned by clustering all the descriptors from a set of training images. Note that these training images are generic, and do not come from the area where robot navigation is tested. However, they are similar in character to images for the navigation task – when navigating outdoors, we train the vocabulary on outdoor imagery, etc. In Section 3.4.3 we will discuss the consequences of learning the visual vocabulary on training data which is visually dissimilar to the test data.

The results presented in this chapter employ a fixed-radius incremental cluster algorithm 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. The threshold is chosen so as to give a visual vocabulary with approximately the desired number of words. In Chapter 5 we will examine the performance impact of different choices of clustering algorithm, including k-means and a hybrid of k-means and the radius-based approach. We find that these alternatives have a non-negligible, but relatively modest, impact on performance. Their effect is less pronounced at high levels of precision, which is the regime that will concern us most.

3.3.2 Learning the Vocabulary Model

Given a visual vocabulary, the next step is to construct a Chow Liu tree to capture the co-occurrence statistics of the visual words. To do this, we construct the mutual information graph as described in Section 3.1.1. Each node in the graph corresponds to a visual word, and the edge weights (mutual information) between node i and j are calculated as per Equation 3.2. The Chow Liu tree is then the maximum weight spanning tree of the graph. We define co-occurrence to mean that the two visual words are present in the same image.

If the Chow Liu tree is to be a good approximation to the true distribution, it must be computed from a large number of observations. To prevent bias, these observations should be independent samples from the observation distribution. We collected 2,800 images from 28km of urban streets using the robot’s camera. The images were taken 10m apart, perpendicular to the robot’s motion, so that they are non-overlapping and approximate independent samples from the observation distribution. From this data set we compute the clustering of SURF features that form our vocabulary, then compute the Chow Liu tree for the vocabulary. The clustering procedure generated a vocabulary of approximately 11,000 words. The combined process takes 2 hours on a 3GHz Pentium IV. This is a one-off computation that occurs offline.

Illustrative results are shown in Figures 3.7, 3.8 and 3.9, which show sample visual words learned by the system and a visualization of a portion of the Chow Liu tree. These words (which correspond to parts of windows) were determined to have high pairwise mutual information, and so are neighbours in

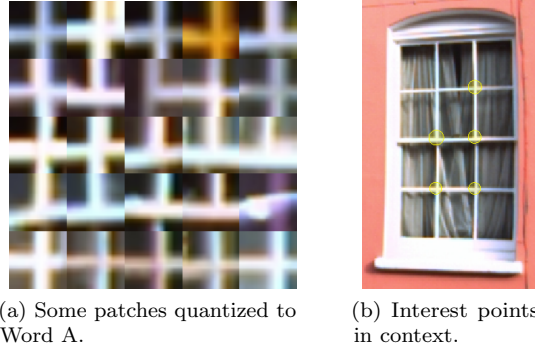


Figure 3.7: A sample word in the vocabulary, showing typical image patches (a) and an example of the interest points in context (b). Interest points quantized to this word typically correspond to the cross-piece of windows. The most correlated word in the vocabulary is shown in Figure 3.8.

the tree. The joint probability of observing the five words shown in the tree is 4,778 times higher under the Chow Liu model than under a Naive Bayes assumption. Effectively the system is learning about the presence of objects that generate the visual words. This significantly improves inference (see Figure 3.18).

3.3.3 Algorithm Configuration

The user defined parameters in our model were set as follows. We modeled our detector by $p(z_q=1|e_q=1) = 0.39$ and $p(z_q=1|e_q=0) = 0$ – i.e. a per-word true positive rate of 39%, and no false positives. To set these parameters, we first assumed a false-positive rate of zero. The false negative rate was then determined by collecting multiple images at a test location – $p(z_q = 0|e_q = 1)$ comes directly from the ratio of the number of words detected in any one image to the number of unique words detected in union of all the images from that location. Finally, we adjusted the value slightly so as to maximize performance on some trial data.

It is worth noting some alternative approaches that could be taken to the calibration of the detector terms. In principle different detector terms could be learned for each word – for example words generated by cars might be less reliably re-observable than words generated by buildings. Exploiting this possibility is discussed in Chapter 6. Additionally, a more precise (but more laborious) detector

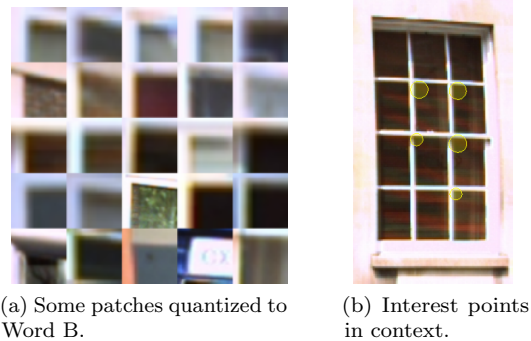


Figure 3.8: A sample word in the vocabulary, showing typical image patches and an example of the interest points in context. Interest points quantized to this word typically correspond to the top-left corner of windows. The most correlated word in the vocabulary is shown in Figure 3.7.

calibration could be achieved by hand-labelling image data and inspecting interest point regions to determine how often the interest point detector fires and how often an interest point is quantized incorrectly.

The number of samples used in the Monte Carlo integration for determining the denominator term $p(Z_k|Z^{k-1})$ was fixed at 2,800 – the set of images used to learn the vocabulary. The smoothing parameter σ was 0.99, and the prior that a new topological link leads to a new place was 0.9. The location prior used a motion model which assumed sequentially collected images, as discussed in Section 3.2. To evaluate the effect of the various other alternative approximations discussed in Section 3.2, the next section presents several sets of results comparing Naive Bayes vs. Chow Liu approximations to $p(Z|L_i)$ and Mean Field vs. Monte Carlo approximations to $p(Z_k|Z^{k-1})$.

3.4 Performance Evaluation

This section presents a performance evaluation of the FAB-MAP model, using imagery collected by a mobile robot. Note that the purpose of this chapter is to evaluate the performance of the FAB-MAP model alone. Therefore, the results presented here do not include a geometric check to verify putative matches, which might mask performance differences between different versions of the model. An evaluation of a more complete system is presented in Chapter 5.

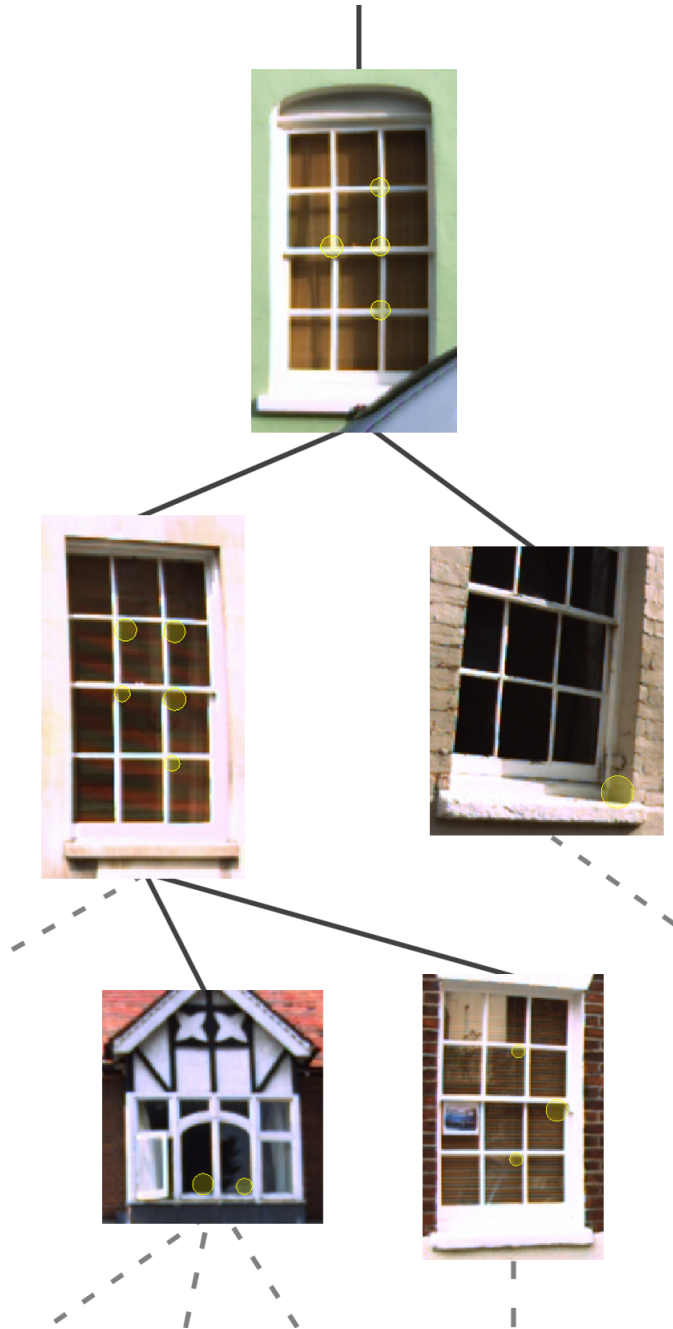


Figure 3.9: Visualization of a section the Chow Liu tree computed for our urban vocabulary. Each word in the tree is represented by a typical example. Clockwise from top, the words correspond to the cross-pieces of window panes, right corners of window stills, top-right corners of window panes, bottom-right corners of window panes and top-left corners of window panes. Under the Chow Liu model the joint probability of observing these words together is 4,778 times higher than under a Naive Bayes assumption.



Figure 3.10: Typical imagery from the environment of the New College data set.



Figure 3.11: Typical imagery from the City Centre data set.

3.4.1 Data Sets

We tested the system on two outdoor urban data sets collected by our mobile robot. Both data sets are available at http://www.robots.ox.ac.uk/~mobile/IJRR_2008_Dataset/. Details of the robot are given in Appendix A. As the robot travels through the environment, it collects images³ to the left and right of its trajectory approximately every 1.5m. The area of our test data sets did not overlap with the region where the training data was collected. The first data set – New College – was chosen to test the system’s robustness to perceptual aliasing. It features several large areas of strong visual repetition, including a medieval cloister with identical repeating archways and a garden area with a long stretch of uniform stone wall and bushes. Typical imagery is shown in Figure 3.10. The second data set – labeled City Centre – was chosen to test matching ability in the presence of scene change. It was collected along public roads near the city centre, and features many dynamic objects such as traffic and pedestrians. Additionally it was collected on a windy day with bright sunshine, which makes

³Image resolution: 640x480.

the abundant foliage and shadow features unstable. Typical imagery is shown in Figure 3.11. Ground truth loop closures were labeled manually for both data sets, on the basis of visual inspection and our knowledge of the robot trajectory.

3.4.2 Performance Metrics

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.

Average precision (AP), or the area under the precision-recall curve, is another commonly quoted performance measure for image retrieval systems. We find AP gives a misleading picture of relative algorithm performance with respect to our application, so do not quote it. AP tends to be heavily impacted by “terminal recall”; that is, the recall achieved when every observation is assigned to some location. A system which achieves a relatively low precision but a high terminal recall can have a high AP score, but would be of limited use for our SLAM application where high precision is critical. 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 the detections have uniform spatial distribution.

3.4.3 Results

We now discuss the overall performance of the system. Figures 3.12 and 3.13 show navigation results overlaid on an aerial photo. These results were generated using the Chow Liu and Monte Carlo approximations, which we found to give the best performance. The system correctly identifies a large proportion of possible loop closures with high confidence. There are no false positives that meet the

probability threshold. Videos of these results are available at the same URL as the data set.

Precision recall curves are shown in Figure 3.14. The curves were generated by varying the probability at which a loop closure was accepted. We are primarily interested in the recall rate at 100% precision – if the system were to be used to complement a metric SLAM algorithm, an incorrect loop closure could cause unrecoverable mapping failure. At 100% precision, the system achieves 48% recall on the New College data set, and 37% on the more challenging City Centre data set. As noted previously, because a typical loop closure is composed of multiple images, even a recall rate of 37% is sufficient to detect almost all loop closures.

Some examples of typical image matching results are presented in Figures 3.15 and 3.16. Figure 3.16 highlights robustness to perceptual aliasing. Here very similar images that originate from different locations are correctly assigned low probability of having come from the same place. We emphasize that these results are not outliers; they show typical system performance. The result is possible because most of the probability mass for the query image is captured by locations in the sampling set – effectively the system has learned that images like these are common in the environment. Of course, had these examples been genuine loop closures they might also have been assigned low probability values. We would argue that this is correct behaviour, modulo the fact that the probabilities in 3.16a and 3.16b seem too low. The very low probabilities for these examples are due to the fact that good matches for the query images are found in the sampling set, capturing almost all the probability mass. This is less likely in the case of a true but ambiguous loop closure.

Figure 3.15 shows matching performance in the presence of scene change. Many of these image pairs have far fewer visual words in common than the examples of perceptual aliasing, yet are assigned high probability of having come from the same place. To give a quantitative example, loop closures detected by the system sometimes have as few as 8% of their visual words in common, whereas perceptual aliasing examples often share most of their visual words – for example, the images shown in Figure 3.16b have 46% of their words in common.

Poor probability estimates do occasionally occur – some examples of images incorrectly assigned high probability of a place match are shown in Figure 3.19. Note however that the typical true loop

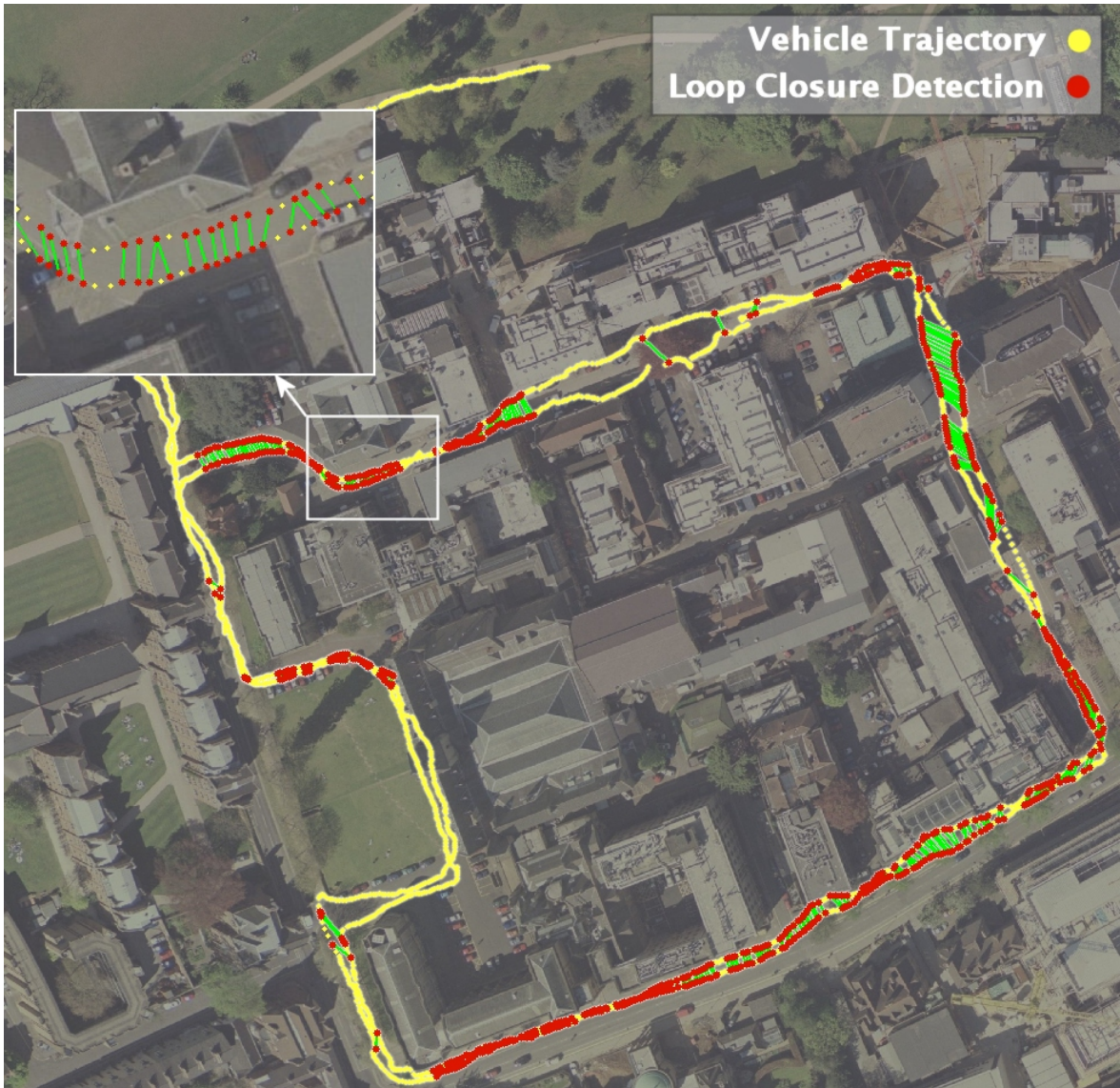


Figure 3.12: Appearance-based matching results for the City Centre data set overlaid on an aerial photograph. The robot travels twice around a loop with total path length 2km, collecting 2,474 images. Positions (from hand-corrected GPS) at which the robot collected an image are marked with a yellow dot. Two images that were assigned a probability $p \geq 0.99$ of having come from the same location (on the basis of appearance alone) are marked in red and joined with a line. There are no incorrect matches that meet this probability threshold. This result is best viewed as a video (available at the same URL as the data set).



Figure 3.13: Appearance-based matching results for the New College data set overlaid on an aerial photograph. The robot traverses a complex trajectory of 1.9km with multiple loop closures. 2,146 images were collected. Positions (from hand-corrected GPS) at which the robot collected an image are marked with a yellow dot. Two images that were assigned a probability $p \geq 0.99$ of having come from the same location (on the basis of appearance alone) are marked in red and joined with a line. There are no incorrect matches that meet this probability threshold. This result is best viewed as a video (available at the same URL as the data set).

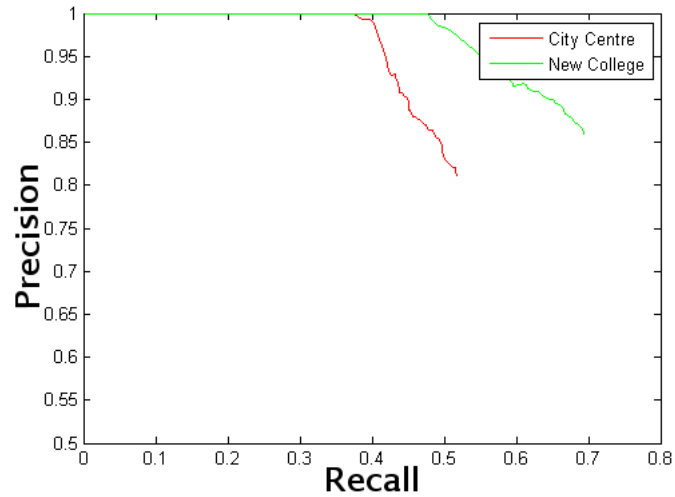


Figure 3.14: Precision-Recall curves for the City Centre and New College data sets. Notice the scale.

closure receives a much higher probability of a match than these examples, primarily due to temporal support from surrounding loop closures. Neither of the image pairs in Figure 3.19 met the loop closure notification threshold of $p > 0.99$.

Comparing Approximations

This section presents a comparison of the alternative approximations discussed in Section 3.2 — Naive Bayes vs. Chow Liu approximations to $p(Z|L_i)$ and Mean Field vs. Monte Carlo approximations to $p(Z_k|\mathcal{Z}^{k-1})$. Figure 3.17 shows precision-recall curves from the New College data set for the four possible combinations. Timing and accuracy results are summarized in Table 3.1. The Chow Liu and Monte Carlo approximations both considerably improve performance. Some examples of typical loop closures detected by the Chow Liu approximation but not by the Naive Bayes are shown in Figure 3.18.

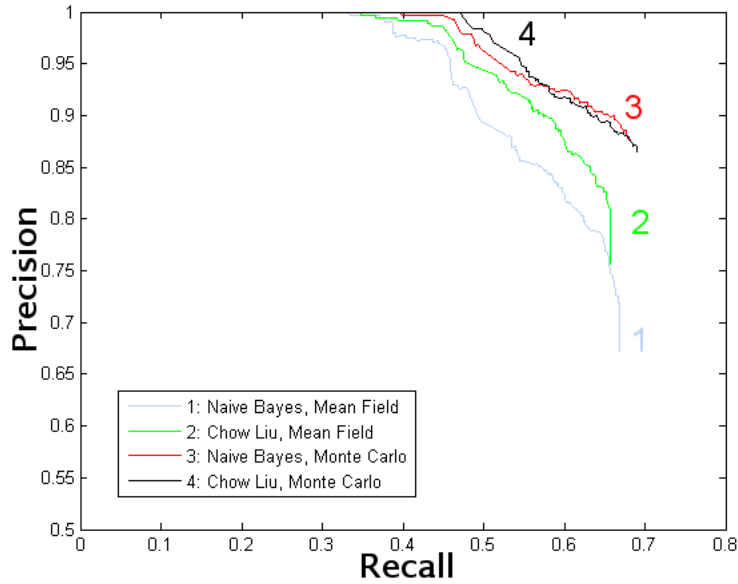


Figure 3.17: Precision-Recall curves for the four variant algorithms on the New College data sets. Notice the scale. All results include the location prior which allows evidence to accumulate through a sequence of observations. Relative performance on the City Centre data set is comparable.

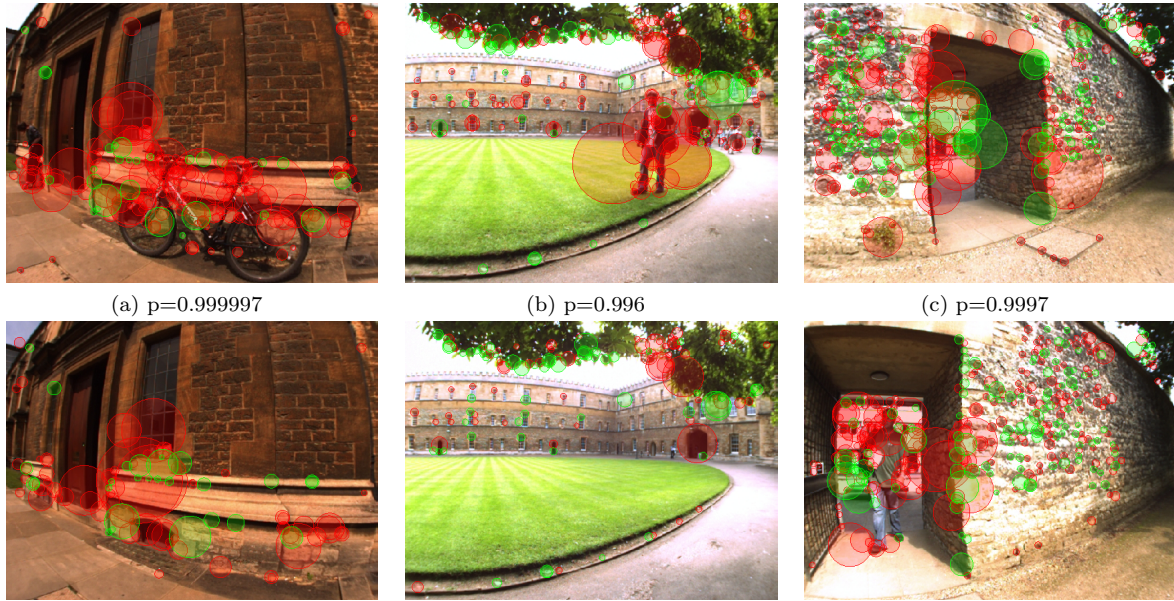


Figure 3.15: Some images assigned high probability of having come from the same place, despite scene change. Words common to both images are shown in green, others in red. Loop closure probability indicated between the pairs.

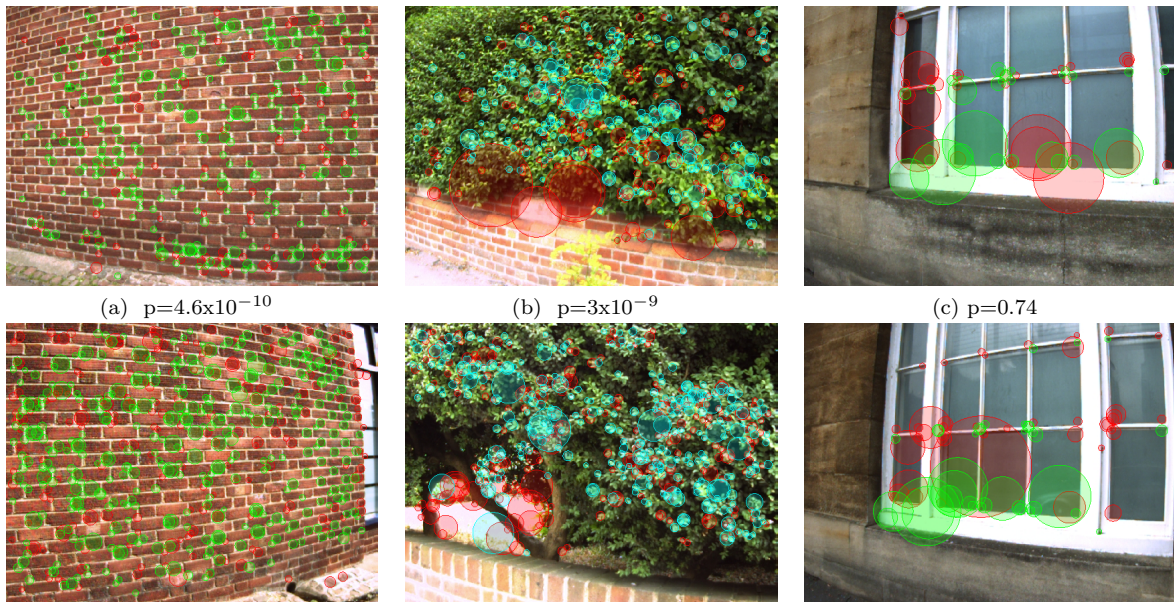


Figure 3.16: Some similar-looking images from different parts of the workspace correctly assigned low probability of having come from the same place. Words common to both images are shown in green (in blue for (b)), others in red.

Algorithm	Recall - New College	Recall - City Centre	Run Time
Mean Field, Naive Bayes	33%	16%	0.6ms/place
Mean Field, Chow Liu	35%	31%	1.1ms/place
Monte Carlo, Naive Bayes	40%	31%	0.6ms/place + 1.71 secs sampling
Monte Carlo, Chow Liu	47%	37%	1.1ms/place + 3.15 secs sampling

Table 3.1: Comparison of the four different approximations. The recall rates quoted are at 100% precision. The time to process a new observation is given as a function of the number of places already in the map, plus a fixed cost to perform the sampling. Timing results are for a 3GHZ Pentium IV.

The extra performance comes at the cost of some increase in computation time; however even the slowest version using Chow Liu and Monte Carlo approximations is still relatively fast. Running times are summarized in Table 3.1. The maximum time taken to process a new observation over all data sets was 5.9 seconds. As the robot collected images approximately every two seconds, this is somewhat slower than required. In Chapters 4 and 5 we will discuss approximate inference techniques that will enable real time performance with minimal accuracy penalty. Additionally, we note that the dominant computational cost is the calculation of $p(Z|L_i)$ for each place model in the map and each sampled place. Each of these calculations is independent; the algorithm is highly parallelizable and will perform well on multi-core processors.

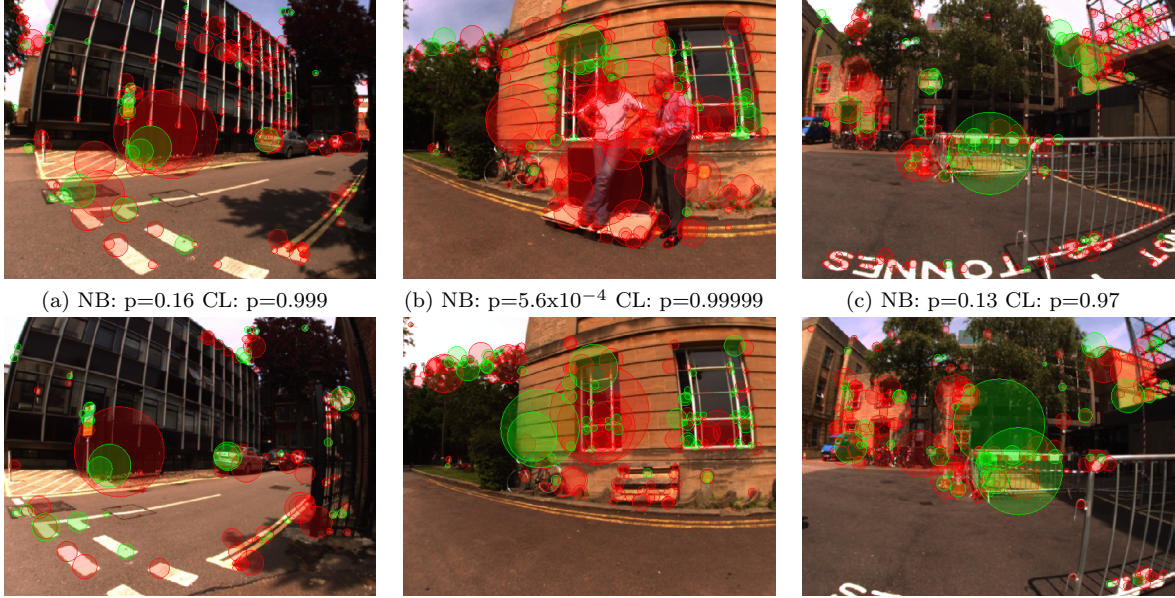


Figure 3.18: Some examples of situations where the Chow Liu approximation outperforms Naive Bayes. In (a), a change in lighting means that the feature detector does not fire on the windows of the building. In (b), the people are no longer present. In (c), the foreground text and the scaffolding in the top right are not present in the second image. In each of these cases, the missing features are known to be correlated by the Chow Liu approximation, hence the more accurate probability. Words common to both images are shown in green, others in red. The probability that the two images come from the same place (according to both the Chow Liu and Naive Bayes models) is indicated between the pairs.



Figure 3.19: Some images from different locations incorrectly assigned high probability of having come from the same place. In (a), the training set contains no examples of railings, so the matched features are not known to be correlated. In (b), we encounter the same truck again in a different part of the workspace. Errors of this type are particularly challenging. Notice that while both images are assigned high probability of a match, a typical true loop closure is assigned much higher probability. Neither of these image pairs met our $p = 0.99$ data association threshold.



(a) Indoor data from handheld video camera (Canon MVX450).



(b) Data from the "Wildcat" autonomous vehicle at BAE Systems, Filton.



(c) Sequence collected with a webcam, for a monocular SLAM experiment.



(d) Frames extracted from a video sequence (results video from [103]).



(e) Image sequence from NASA JPL Mars Yard.



(f) DLR Spatial Cognition data set.

Figure 3.20: Some data sets on which the system has been tested.

3.4.4 Generalization Performance

Because FAB-MAP relies on a learned appearance model to reject perceptual aliasing and improve inference, a natural question is to what extent system performance will degrade in environments very dissimilar to the training set. We have not fully investigated this issue because we have only a limited number of data sets for which labeled ground truth was produced. However, we have informally run the algorithm on a much larger number of data sets and our experience suggests that performance degrades gracefully. Recall typically drops somewhat, but the system almost always detects a useful fraction of the loop closures, with no or very few false positives.

A selection of the different types of data used in testing is shown in Figure 3.20. Note that we processed all of these data sets using the same visual vocabulary used for the New College and City Centre experiments. The data sets shown are:

1. A handheld video sequence collected in an indoor environment using a consumer camcorder (Figure 3.20a). Notable for motion blur.
2. Data from the “Wildcat” autonomous vehicle provided by BAE Systems, Filton (Figure 3.20b). Notable for severe interlacing artifacts.
3. A sequence collected with a webcam, for a monocular SLAM experiment (Figure 3.20c). Data is from [164].
4. Frames extracted directly from an internet-quality video (Figure 3.20d). This is the results video from [103].
5. An image sequence from the NASA Jet Propulsion Laboratory Mars Yard test facility (Figure 3.20e).
6. The DLR Spatial Cognition data set [75] from the Inside Data Association workshop at Robotics: Science and Systems 2008 (Figure 3.20f).

To provide some quantitative measure of system performance on data dissimilar to the training set, we manually labeled ground truth for the indoor image sequence shown in Figure 3.20a, and processed it using the same vocabulary and Chow Liu tree used for the New College and City Centre experiments. Videos of the indoor results can be found at http://www.robots.ox.ac.uk/~mobile/IJRR_2008_Dataset/. While the indoor data set is too small to draw strong conclusions from (only 188 images), results are nevertheless interesting. Firstly, the quality of the imagery in this indoor data set is considerably poorer than that from the robot, due to adverse lighting conditions and greater motion blur. Despite this, system performance is largely similar to our outdoor results (46% recall at 100% precision), although it should be noted that the numbers are not strictly comparable because of the different sizes of the two data sets. The fact that the system works indoors despite using an outdoor vocabulary should not be so surprising in light of the discussion in Chapter 2 (Section 2.6). To investigate how large an effect the vocabulary has on performance, we learned an indoor-specific vocabulary from a set of 1,679 images collected with the same hand-held video camera as used for the data set. The area where these images were collected did not overlap with the area used for the navigation experiment. As

expected, performance improves with the more appropriate vocabulary, reaching 59% recall at 100% precision.

One effect that we did not expect is that when using the outdoor vocabulary on the indoor data set, the Chow Liu model continues to outperform Naive Bayes (46% recall as opposed to 39%). This is surprising, and suggests perhaps that some of the correlations being learned by the Chow Liu tree model generic low-level aspects of imagery. We should emphasise however that the data set is not large enough to fully support such a conclusion.

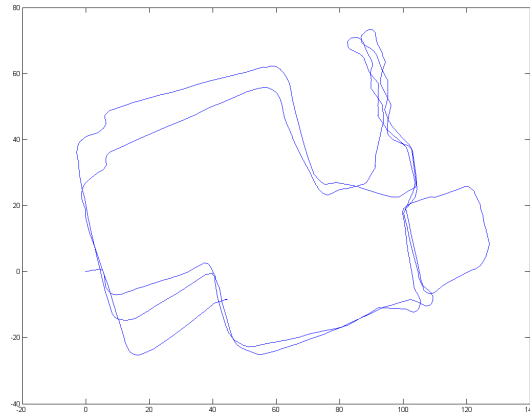
Overall, the quantitative results on this indoor sequence are in line with our experience on a large number of data sets – system performance degrades gracefully in environments dissimilar to the training data.

3.4.5 Use with Metric SLAM Systems

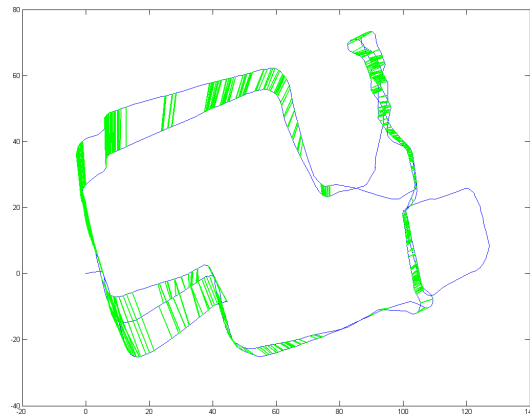
The motivation for the development of FAB-MAP was to solve the loop closure detection and multi-session mapping problems for metric SLAM systems. This section briefly presents some results demonstrating the use of FAB-MAP in that context. Typical results are shown in Figure 3.21. The vehicle trajectory, consisting of a set of vehicle poses and the relative metric transformations between them (with associated uncertainty), is determined by a visual SLAM system developed by Mei *et al.* [91]. Loop closure constraints between poses are determined by FAB-MAP as described in this chapter. These constraints are shown in Figure 3.21b as green links between different parts of the trajectory. The relative metric transformation for these constraints is determined by the visual SLAM system using stereo image pairs collected by the robot at each location. The final trajectory estimate is obtained by relaxing the pose graph using the method described in [110].

An example of multi-session mapping using the same basic approach is shown in Figure 3.22. Here, the alignment between two partially overlapping data sets collected on different days is recovered. This is achieved in exactly the same way as for the single-session mapping example. The only difference is that when the robot begins the second mapping session, we load the place models learned by FAB-MAP during the previous session, so that these places can be recognised.

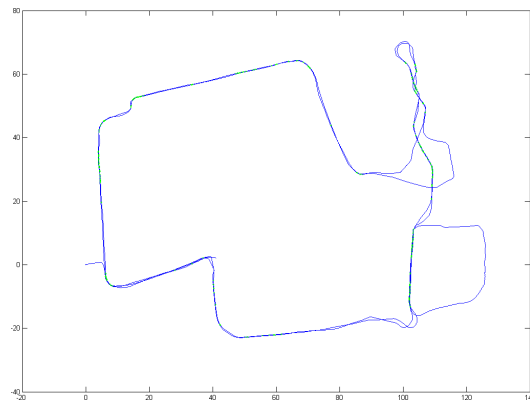
More details of the experiments summarized here may be found in [110]. The data set used is that of [145].



(a) Trajectory from the visual SLAM system. The robot travels twice around the loop. The position estimate is clearly drifting over time.

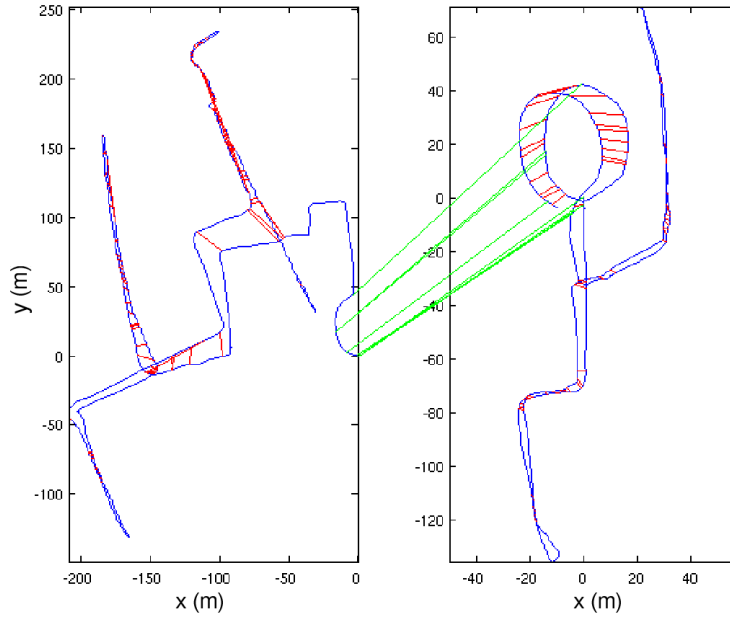


(b) Place recognition constraints from FAB-MAP.

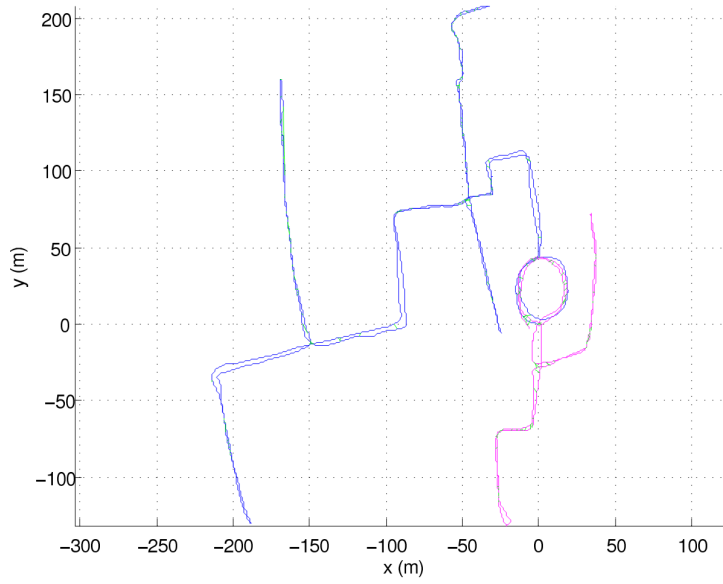


(c) Final trajectory estimate after relaxation.

Figure 3.21: Improving a metric SLAM estimate using FAB-MAP.



(a) Two input maps.



(b) Single output map.

Figure 3.22: Use of FAB-MAP for multi-session mapping. Sub-figure (a) shows two robot trajectories collected on different days (blue), which have an unknown transformation between them. Place recognition constraints between poses in these trajectories are detected by FAB-MAP. Constraints within each trajectory are shown as red links, and between the two trajectories as green links. Pose graph relaxation yields the final output trajectory shown in (b). The two trajectories now share a common coordinate frame.

3.5 Chapter Summary

This chapter has introduced the FAB-MAP algorithm, a probabilistic framework for navigation and mapping which relies on appearance information only. The system is robust even in visually repetitive environments and is fast enough for online loop closure detection in realistic mobile robotics applications. In our evaluation, FAB-MAP was successful in detecting large portions of loop closures in challenging outdoor environments, even when we require a confidence level such that there are no false positives.

Three aspects of our results are particularly noteworthy. Firstly, we outline a principled approach for determining when an observation comes from a location not already in the map. To our knowledge this is the first approach to the problem that goes beyond simple heuristics such as fixed similarity thresholds. The more principled approach gives a large performance improvement. It also naturally handles perceptual aliasing. Our second notable contribution was to show that learning a generative model of the bag-of-words observations yields a marked improvement in performance. We think that this technique will have applications beyond the specific problem addressed in this thesis. Thirdly, we have observed that the system appears to perform well even in environments quite dissimilar to the training data. This suggests that the approach is practical for deployment in unconstrained navigation tasks, as a natural complement to more typical metric SLAM algorithms.

Chapter 4

Approximate Inference using Concentration Inequalities

4.1 Introduction

This chapter is concerned with speed improvements to the appearance-only SLAM system introduced in Chapter 3. We show that by employing a probabilistic bail-out test in the core likelihood calculation required by FAB-MAP, speed improvements of between 25 and 50 times are possible, with only slight accuracy penalty. We achieve this performance improvement by introducing a novel approach to multihypothesis testing based on concentration inequalities, the applicability of which is quite general. Typical filter update times for the improved system are on the order of 150 ms for maps which contain several thousand locations. This enables real-time loop closure detection on a mobile robot for loops tens of kilometers in length.

4.2 Bail-Out Strategies

In Chapter 3 we introduced the probabilistic model for our appearance-only SLAM system. The limiting computational cost of the method was the calculation of the observation likelihood term $p(Z_t|L_i, \mathcal{Z}^{t-1})$

for each location in the map and for each sample in the Monte Carlo sampling set. Typically, only a small number of these locations will yield non-negligible probability of having generated the current observation. The core idea of this chapter is that, by identifying locations that will have insignificant likelihood before the calculation is fully complete, many locations can be excluded quickly and large speed increases can be realized.

Closely related ideas have been described elsewhere in computer vision, notably in the context of efficient RANSAC algorithms [113, 89]. Matas and Chum showed that for RANSAC, the sequential probability ratio test (SPRT) yields the optimal solution. The SPRT approach was originally designed for testing two hypotheses under a sequence of identical and equally informative observations [158]. Extensions exist for the multihypothesis case [5]. However, stopping boundaries for the SPRT are not easy to derive when the observations are not equally informative. We describe an alternative approach based on concentration inequalities [16]. Unlike the SPRT, this approach is straight-forward to apply even when there are multiple hypotheses and the observations are not equally informative. We have noted related ideas in other fields [88], however we believe our approach is novel in this context.

We begin by introducing our bail out test in the most general form in Section 4.3, before discussing how to apply it to our appearance-based navigation task in Section 4.4.

4.3 Probabilistic Bail-Out using Bennett's Inequality

Let $\mathcal{H} = \{H^1, \dots, H^K\}$ be a set of K hypotheses (models) and let $Z = \{z_1, \dots, z_N\}$ be an observation consisting of N features. The likelihood of the observation under the k^{th} hypothesis is given by

$$p(Z|H^k) = p(z_1|z_2, \dots, z_N, H^k) \dots p(z_{N-1}|z_N, H^k) p(z_N|H^k) \quad (4.1)$$

Define the log-likelihood of the first i features under the k^{th} hypothesis as

$$D_i^k = \sum_{j=1}^i d_j^k \quad (4.2)$$

where

$$d_i^k = \ln(p(z_i | z_{i+1}, \dots, z_N, H_k)) \quad (4.3)$$

is the log-likelihood of the i^{th} feature under the k^{th} hypothesis. We would like to determine, as rapidly as possible, the hypothesis H^* for which the total log-likelihood D_N^* is maximized. Finding H^* with certainty requires a complete evaluation of the likelihood of each hypothesis, which may be too slow for applications of interest. Consequently, we consider the problem of finding a hypothesis $H^\#$, subject to the constraint that $p(H^\# \neq H^*) < \epsilon$, where ϵ is some user-specified probability.

In overview, our approach is to calculate the likelihoods of all hypotheses in parallel, and terminate the likelihood calculation for hypotheses that have fallen too far behind the current leading hypothesis. “Too far” can be quantified using concentration inequalities, which yield a bound on the probability that a hypothesis will overtake the leader, given their current difference in likelihoods and some statistics about the properties of the features which remain to be evaluated. Figure 4.1 illustrates the idea.

Consider two hypotheses $H^x, H^y \in \mathcal{H}$ and let

$$X_i = d_i^x - d_i^y \quad (4.4)$$

the difference in the log-likelihood of feature i under hypothesis H^x and H^y . X_i can be considered as a random variable before its value has been calculated. This is useful because we can calculate some key statistics about X_i more cheaply than we can determine its exact value. Now define

$$S_n = \sum_{i=n+1}^N X_i \quad (4.5)$$

If after evaluating n features, the log-likelihood of some hypothesis is Δ less than the current best hypothesis, then the probability of failing to locate H^* if we discard this hypothesis is given by $p(S_n > \Delta)$ – i.e. by the probability that the trailing hypothesis will overtake the leader after the evaluation of the remaining features. Thus, knowing the distribution of S_n allows the creation of a probabilistic bail-out test for discarding hypotheses subject to an error constraint. Calculating an explicit distribution on S_n is infeasible, however concentration inequalities – which bound the probability that a function of

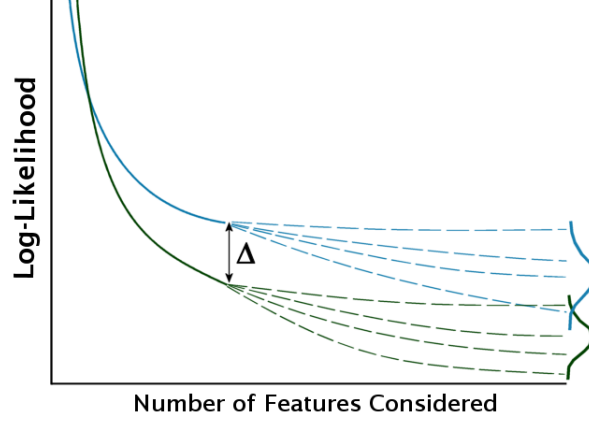


Figure 4.1: Conceptual illustration of the bail-out test. After considering the first i features, the difference in log-likelihoods between two hypotheses is Δ . Given some statistics about the remaining features, it is possible to compute a bound on the probability that the evaluation of the remaining features will cause one hypothesis to overtake the other. If this probability is sufficiently small, the trailing hypothesis can be discarded.

random variables will deviate from its mean value – can be applied to yield bounds on $p(S_n > \Delta)$.

A large variety of concentration inequalities exist, many of which apply under very general conditions, including cases where the component distributions are not identically distributed, not independent, and are combined using arbitrary functions. The more information available about the component distributions X_i , the tighter the bound. For an overview see [16]. For our application we will find it convenient to use the Bennett inequality for sums of symmetric random variables [9]. This inequality is specified in terms of two parameters — M , a bound on the maximum value of any component X_i , and v , a bound on the sum of the variances of the components X_i .

Formally, let $\{X_i\}_{i=n+1}^N$ be a collection of independent mean-zero random variables with symmetric distributions (corresponding to the log-likelihood changes due to those features not yet considered), and satisfying the conditions

$$p(|X_i| < M) = 1, \forall i : \{n+1 < i < N\} \quad (4.6)$$

$$\sum_{i=n+1}^N E[X_i^2] < v \quad (4.7)$$

and as before

$$S_n = \sum_{i=n+1}^N X_i \quad (4.8)$$

then the Bennett inequality states that

$$p(S_n > \Delta) < \exp \left(\frac{v}{M^2} \cosh(f(\Delta)) - 1 - \frac{\Delta M}{v} f(\Delta) \right) \quad (4.9)$$

where

$$f(\Delta) = \sinh^{-1} \left(\frac{\Delta M}{v} \right) \quad (4.10)$$

Thus, after considering the first n features, and given bounds on the maximum value (M) and sum of variances (v) of the inter-hypothesis log-likelihood changes (X_i) for the features yet to be evaluated ($n+1 : N$), we can solve the Bennett inequality for Δ , such that $p(S_n > \Delta)$ meets the user specified error probability ϵ . We solve for Δ using a combination of bisection and Newton-Raphson iteration. After considering a given feature, any hypothesis whose log-likelihood is at least Δ less than the current leading hypothesis can be discarded, because the probability of it overtaking the current leader before the end of the calculation is less than ϵ . Note that the values of M and v are defined on the features yet to be evaluated ($n+1 : N$), thus as the calculation progresses and the number of unconsidered features decreases, M and v will decrease. The bail-out threshold thus converges to zero as the number of unconsidered features decreases (see Figure 4.2).

4.4 Application to Appearance-Only SLAM

We now turn our attention to applying this bail-out condition to our appearance-only SLAM system. Recalling our notation from Chapter 3, we have a set of locations L_i whose likelihood we are evaluating with respect to some observation Z . Each of our features z_i is a binary variable indicating whether or not the i^{th} word of the visual vocabulary was present in the current observation. We will denote the difference in log-likelihood of two locations L_x and L_y under feature z_i by

$$\begin{aligned}
X_i &= d_i^x - d_i^y \\
&= \ln(p(z_i|z_{p_i}, L_x)) - \ln(p(z_i|z_{p_i}, L_y))
\end{aligned} \tag{4.11}$$

which we will consider as a random variable before its value has been calculated. To apply our bail-out test, we must establish that our random variables X_i meet the conditions for applicability of Bennett's inequality. We must also define an order in which to consider the features, and outline how M and v may be calculated.

4.4.1 Applicability of Bennett's Inequality

To make use of Equation 4.9, we require that the random variables $\{X_i\}_{i=1}^N$ be independent, mean-zero and have symmetric distributions. We also require that their maximum values be bounded.

To understand why these conditions are satisfied, we note that, given that the values of z_i and z_{p_i} are fixed, the likelihood of a feature being observed at a particular location, $p(z_i|z_{p_i}, L_x)$, depends only on the number of previous observations of feature z_i at location L_x (see Equation 3.13). Thus, assuming the number of observations relating to any given place is finite, X_i has a multinomial distribution because $p(z_i|z_{p_i}, L_x)$ can be one of a fixed set of discrete values. If we now make the assumption that, a-priori, all locations are equally likely to generate any given feature, X_i must have a symmetric mean-zero distribution. This is because if $X_i = \delta$ for some history of previous observations at L_x and L_y , then $X_i = -\delta$ for the symmetric case where the observation history of L_x and L_y is swapped, which occurs with equal probability.

To illustrate this, consider an example. Let z_i be a feature which is observed with probability η , and let L_x and L_y be two location models which each have exactly one associated observation. Further assume that the state of the feature z_i in the observation associated with L_x and L_y (i.e. $z_i = 0$ or $z_i = 1$) is unknown. For compactness, denote the event that location L_x has a previous associated observation with $z_i = 0$ by $L_x\{0\}$. There are four possible cases:

Case	Probability	Value of X_i
$L_x\{0\}, L_y\{0\}$	$p = (1 - \eta)^2$	0
$L_x\{0\}, L_y\{1\}$	$p = \eta(1 - \eta)$	δ
$L_x\{1\}, L_y\{0\}$	$p = \eta(1 - \eta)$	$-\delta$
$L_x\{1\}, L_y\{1\}$	$p = \eta^2$	0

which occur with the probabilities shown. Also shown is the log-likelihood difference (X_i) in the four cases, where δ is the value of $\ln(p(z_i|z_{p_i}, L_x\{0\})) - \ln(p(z_i|z_{p_i}, L_y\{1\}))$. Observe that X_i has a symmetric, mean-zero multinomial distribution:

$$X_i : \begin{cases} \delta & \text{with probability } \eta(1 - \eta) \\ 0 & \text{with probability } (1 - \eta)^2 + \eta^2 \\ -\delta & \text{with probability } \eta(1 - \eta) \end{cases} \quad (4.12)$$

When the location models have more than one associated observation this multinomial distribution has a similar form, but a larger number of states.

Finally, Bennett's inequality requires that the variables X_i be independent. Our Chow Liu model captures much but not all of the conditional dependence between features. Thus the variables X_i may have weak residual dependence. However, our experimental results (Section 4.5) would appear to indicate that this is not a problem in practice.

4.4.2 Ranking Features

Next, we must define an order in which to consider the features during the bail-out test. While the bail-out test applies to any ordering, it is natural to rank the features by information gain. That way, the hypotheses will converge most rapidly toward their final log-likelihood values and poor hypotheses can be identified earliest (see Figure 4.2).

Under our observation model, each feature z_i is conditionally dependent on one other feature z_{p_i} . If we observe $z_i = s_i$ and $z_{p_i} = s_{p_i}$ (with $s \in \{0, 1\}$), then the information gain associated with this

observation under our model is

$$I = -\ln p(z_i = s_i | z_{p_i} = s_{p_i}) \quad (4.13)$$

Typically observations of rare words are the highest ranked features, though, perhaps surprisingly, failure to observe a word can sometimes also have high information gain – for example, if two words are almost always observed together, then failure to observe one while observing the other is an informative observation.

Note that in our implementation the probabilities in Equation 4.13 come from the training data on which we learnt the model of our visual words. As such we are calculating the information gain with respect to the places in the training data. Strictly we should consider the information gain with respect to the set of places in our current map – for example, some feature might be very rare in the training set but very common in the map. In practice we observe that the difference between the two values is usually small, so maintaining a separate set of probabilities is unnecessary. As the bail-out test applies to any feature ordering, the effect of this simplified implementation is merely that hypotheses may be discarded slightly later than optimal.

4.4.3 Calculating M and v

Finally, to apply Bennett’s inequality, we must calculate M and v , the parameters of the concentration inequality which depend on the component random variables X_i . As per Equation 4.6, M is a bound on the maximum value of $X_i, \forall i : \{n + 1 < i < N\}$ – that is, the maximum of the log-likelihood differences due to feature i between some trailing location hypothesis L_x and the leading hypothesis at the time when feature i is considered, over all indices i between $n + 1$ and N . Calculating an exact value for M would require knowledge of the leading hypothesis at a future point in the calculation, which is unknown. However, as M is a *bound* on the maximum value of X_i , it can be calculated as the maximum inter-hypothesis log-likelihood change over *all* hypothesis pairs, and all remaining features. This bound can easily be calculated by keeping track, for each feature i , of the location that was most and least likely to have generated that feature, as these location pairs maximize X_i .

The value of v can be determined directly as the sum of the variances of the distributions shown

in Equation 4.12. This is practical when the number of observations associated with each location in the map is small. However, as exploration continues and the number of observations associated with a typical place in the map increases, the calculation of this variance rapidly becomes expensive. At some point it may be beneficial to switch from using Bennett’s inequality to Hoeffding’s inequality [61], a similar concentration inequality that requires knowledge only of the maximum value of each X_i . Hoeffding’s inequality gives a weaker bound, but this is compensated for by the fact that by the time the variance becomes expensive to compute, the place models themselves are more differentiated, and so their likelihoods will diverge faster.

4.4.4 Calculating a pdf over Hypotheses

One remaining issue is that our appearance-only SLAM system requires a pdf over hypotheses, whereas our discussion so far has concerned locating only the best hypothesis. Computing a pdf requires a simple modification to the bail-out scheme. Consider that instead of locating only the best hypothesis H^* , we would like to locate all hypotheses whose log-likelihood is at most C less than that of H^* . C is a user-specified constant chosen so that hypotheses less likely than this can be considered to have zero probability with minimal error. Simply increasing our bail-out distance by C will retain all those hypotheses whose final likelihood may be within this likelihood range, thus giving us a close approximation to the pdf over hypotheses.

4.5 Results

We tested the system on the New College and City Centre data sets introduced in Chapter 3. The New College data set is 1.9km and features a number of challenging cases of perceptual aliasing. The City Centre data set is 2km in length and is particularly rich in dynamic objects. We also add a third data set – Parks Road – featuring a typical suburban environment. In all three data sets the robot collected images to the left and right of its trajectory approximately every 1.5m. The robot’s appearance model was built from a fourth data set collected in a different region of the city, the area of which did not overlap with the test sets.

Navigation results for these data sets were generated using both the original system outlined in Chapter 3 and the accelerated system incorporating the bail-out test. All data sets were processed using the same visual vocabulary and algorithm parameters. The bail-out boundary was set so that the probability of incorrectly discarding the best hypothesis at any step was $< 10^{-6}$. This value can be varied to trade off speed against accuracy.

Results are summarized in the figures below. Figure 4.2 illustrates the bail-out calculation on some real data. Ordering the features by information gain clearly has a dramatic effect on the effectiveness of the bail-out test. Precision-recall curves for the full and accelerated algorithms on the City Centre data set are shown in Figure 4.4. The curves were generated by varying the probability at which a loop closure was accepted. Recall rates are quoted in terms of image-to-image matches. As a typical loop closure is composed of multiple images, even a recall rate of 35% is sufficient to detect almost all loop closures. The relative performance of the two algorithms on the other data sets is summarized in Table 4.1.

Figure 4.3 visualizes the overall performance of the accelerated algorithm on the City Centre data set. The system correctly identifies a large proportion of possible loop closures with high confidence. There are no false positives that meet the probability threshold. Figures 4.5 and 4.6 show some examples of place recognition performance, highlighting the fact that matching ability in the presence of scene change and robustness to perceptual aliasing is not significantly compromised by the bail-out test. The robustness to perceptual aliasing is particularly noteworthy. Of course, had the examples shown in Figure 4.6 been genuine loop closures they might also have received low probability of having come from the same place. We would argue that this is correct behaviour, modulo the fact that the probabilities in (a) and (b) seem too low. The very low probabilities in (a) and (b) are due to the fact that the best matches for the query images are found in the sampling set, capturing almost all the probability mass. This is less likely in the case of a true but ambiguous loop closure, particularly because in the case of a true loop closure the ambiguity can be resolved by temporal information via the prior term in Equation 2.1.

Dataset	Full Calculation		Fast Bail-Out		Speed-Up
	Recall	Mean Time	Recall	Mean Time	
City Centre	37%	5015 ms	35%	141 ms	35.5
New College	46%	4818 ms	42%	178 ms	27.0
Parks Road	44%	4267 ms	40%	79 ms	53.6

Table 4.1: Comparison of the performance of the SLAM system using full and accelerated likelihood calculations. The recall rates quoted are at 100% precision. Timing results are for the filter update, on a 3GHz Pentium IV. Feature generation adds an extra 330 ms on average. Update time for the accelerated calculation is data dependent and varies from observation to observation. Time quoted is the average over the data set.

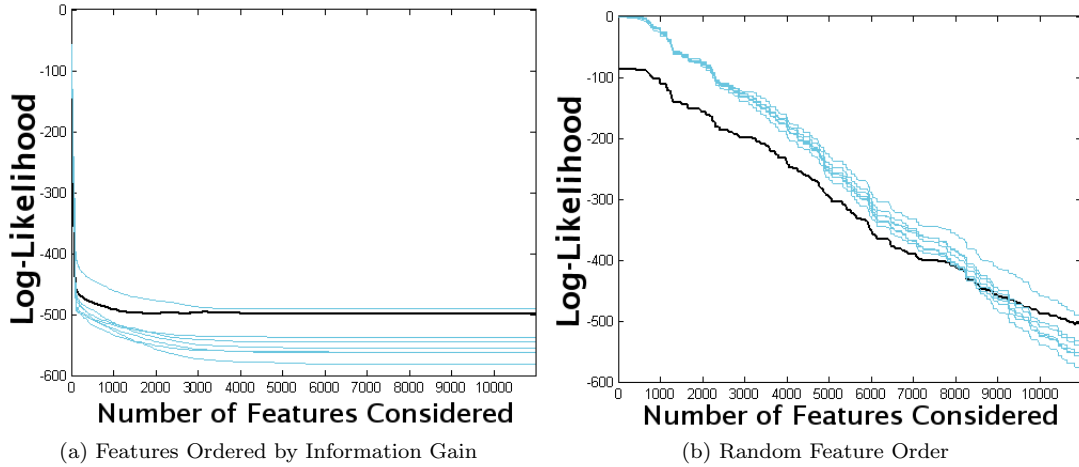


Figure 4.2: Bail-out test on real data. Here the blue lines show the log-likelihoods of each place versus number of features considered. Typically there are thousands of places - here only a few are shown for clarity. The black line is the bail-out threshold. Once the likelihood of a place hypothesis falls below the bail-out threshold, its likelihood calculation can be terminated (the remainder of the likelihood calculation is shown above for illustration). The calculations shown in (a) and (b) are the same, the only difference being the order in which features are considered. In (a), observations are ordered by information gain; in (b) they are ordered randomly. Note that ordering the features by information gain results in much faster convergence toward final likelihood values, and hence a much more effective bail-out test. The bail-out threshold does not converge to the leading hypothesis because of the offset constant C .

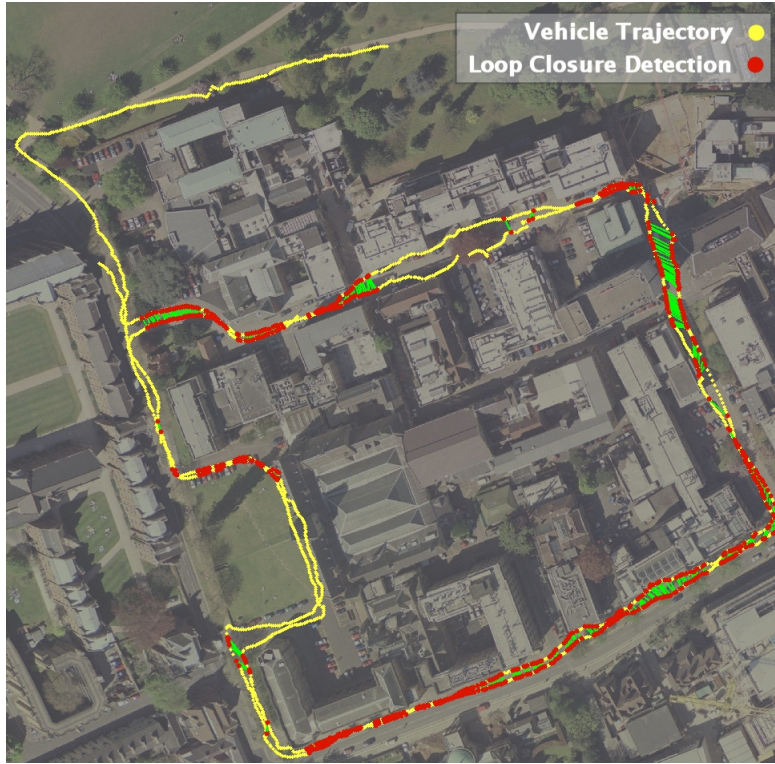


Figure 4.3: Appearance-only matching results (using the accelerated algorithm) for the City Centre data set overlaid on an aerial photograph. The robot travels twice around a loop with total path length 2km, collecting 2,474 images. Each of these images is determined to be either a new place or a loop closure. Positions (from hand-corrected GPS) at which the robot collected an image are marked with a yellow dot. Two images that were assigned a probability $p \geq 0.99$ of having come from the same location are marked in red and joined with a green line. There are no incorrect matches that meet this probability threshold.

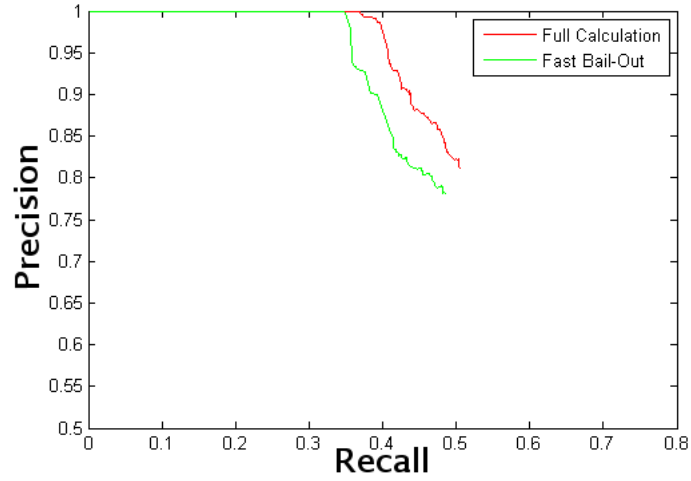


Figure 4.4: Precision-Recall curves for the City Centre data set, showing the full likelihood calculation (red) and the accelerated calculation using the bail-out test (green). Notice the scale.

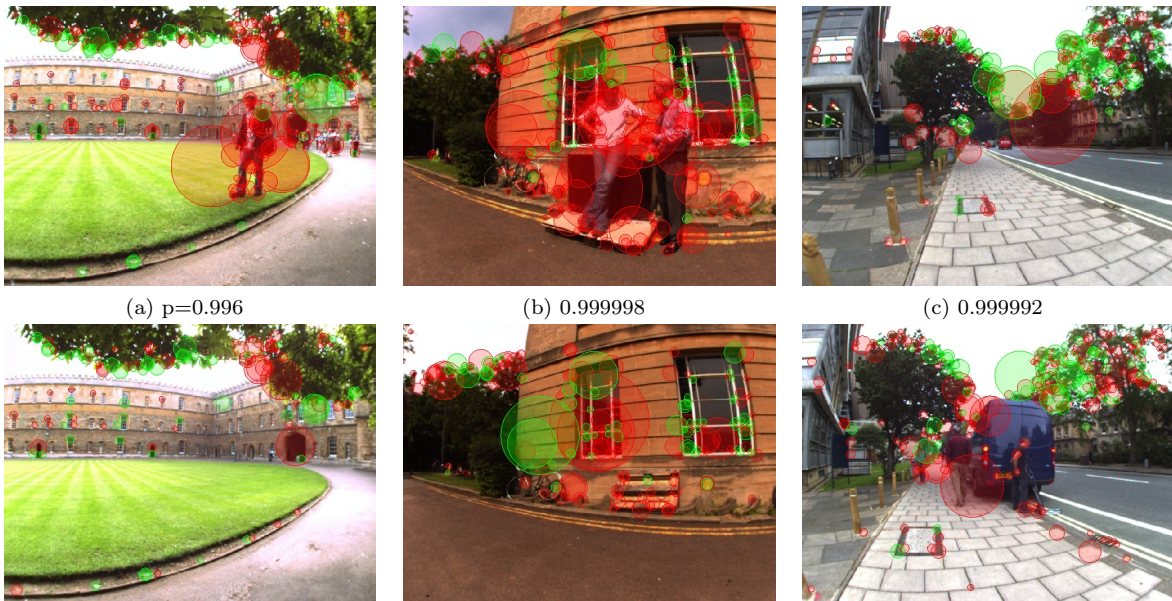


Figure 4.5: Some examples of images that were assigned high probability of having come from the same place, despite scene change. Results were generated using the accelerated likelihood calculation. Words common to both images are shown in green, others in red. The probability that the two images come from the same place is indicated between the pairs.



Figure 4.6: Some examples of remarkably similar-looking images from different parts of the workspace that were correctly assigned low probability of having come from the same place. Results were generated using the accelerated likelihood calculation. The examples represent typical system performance. This result is possible because most of the probability mass is captured by locations in the sampling set – effectively the system has learned that images like these are common in the environment. Words present in both images are shown in green, others in red. (Common words shown in blue in (b) for better contrast). The probability that the two images come from the same place is indicated between the pairs.

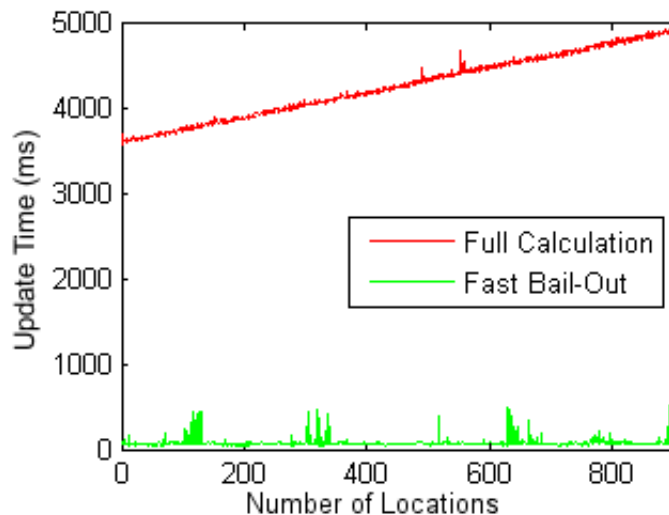


Figure 4.7: Filter update time versus the number of locations in the map, for the Parks Road data set. Update time with zero locations is non-zero due to the fixed cost of evaluating the partition function. Calculation time with the bail-out test grows linearly, however the slope is too small to be seen on this graph.

4.6 Chapter Summary

This chapter has presented a new approach to rapid multihypothesis testing using a probabilistic bail-out condition based on concentration inequalities. Concentration inequalities exist that apply under very general conditions, even for arbitrary functions of non-iid random variables, hence our basic idea should be applicable to a wide variety of problems. We have applied the bail-out test to accelerate our appearance-only SLAM system. The speed increase is data-dependent, but acceleration factors in the range 25x-50x are typical in our tests. The location recognition performance of the accelerated system is only marginally less than that of the full solution, and more than sufficient for reliable online loop closure detection in mobile robotics applications. We have presented results demonstrating online loop-closure detection over 2km loops, though the system is fast enough to scale to loops of tens of kilometres in length while maintaining sub-second filter update times. However, for truly long-term navigation on the order of hundreds or thousands of kilometers, a different approach will be necessary. We investigate appearance-only navigation in this regime in the next chapter.

Chapter 5

Inverted Index Techniques for Large Scale Place Recognition

This chapter is concerned with the problem of appearance-based place recognition at very large scales. Thus far we have considered systems suitable for navigation over a few kilometers (Chapter 3), and approximate inference techniques which extend applicability to a few tens of kilometers (Chapter 4). This chapter builds on the probabilistic framework introduced in those chapters, 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 considered before 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 chapter 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.

5.1 Related Work

Appearance-based place recognition for trajectories much above a kilometer long has only recently begun to be considered by SLAM researchers. The small number of relevant papers were introduced in Chapter 2 (Section 2.3.3), and notably include the contributions of Milford and Wyeth [99], and Bosse and Zlot [14]. In the computer vision community, image retrieval from large databases is more commonplace [144, 114, 122, 63]. However, the problem of retrieval from a fixed database is considerably easier than the appearance-based navigation 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 chapter currently represent the largest scale system that fully addresses these issues of incrementality and perceptual aliasing.

5.2 System Description

This section discusses the development of a system suitable for appearance-based navigation for environments where the map may contain hundreds of thousands of places or more. We first introduce a modified probabilistic model, and then discuss systems issues that arise. Visual vocabulary and Chow Liu tree learning are revisited in light of the scalability requirements of the new system, and a geometric verification stage is introduced which we found to be almost essential on our largest data sets.

The probabilistic model employed in the new system builds directly on the scheme outlined in Chapter 3. 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.

5.2.1 Inverted Indices

An inverted index is a simple data structure used throughout information retrieval, which enables efficient search of large document collections [87]. 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 [19]. The data structure is illustrated in Figure 5.2.

5.2.2 FAB-MAP 2.0 - An Approximation to FAB-MAP for Large Scale Applications

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 $p(Z_k|L_i)$ requires evaluation of Equation 3.12, $\prod_{q=2}^{|v|} p(z_q|z_{p_q}, L_i)$. The computation pattern is illustrated in Figure 5.3. 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

Document	Words
1	"September", "rain", "falls", "on", "the", "house"
2	"the", "little", "moons", "fall", "down", "like", "tears"
3	"With", "crayons", "the", "child", "draws", "a", "rigid", "house"
4	"Time", "to", "plant", "tears", "says", "the", "almanac"
5	"and", "the", "child", "draws", "another", "inscrutable", "house"

Word	Documents
a	3
almanac	4
and	5
another	5
child	3, 5
crayons	3
down	2
draws	3, 5
fall	2
falls	1
house	1, 3, 5

tears	2, 4
the	1, 2, 3, 4, 5
Time	4
to	4

(a) Documents in the database.

(b) Inverted index.

Figure 5.2: Illustration of an inverted index data structure. A mapping from words to documents is maintained, enabling fast retrieval.

to scale. 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 3.5, location models are parametrized as $\{p(e_1 = 1|L_i), \dots, p(e_{|v|} = 1|L_i)\}$, 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 $p(e_q|L_i)|_{\{0\}}$ 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 $p(e_q|L_i)|_0$ indicate that in all observations associated with the location, $z_q = 0$.

In the general probabilistic model described in Section 3.2, $p(e_q|L_i)|_0$ can take on a range of values – for example, $p(e_q|L_i)|_{\{0\}} \neq p(e_q|L_i)|_{\{0,0\}}$, as the belief in the non-existence of the scene elements increases as more supporting observations become available. In FAB-MAP 2.0, the model is restricted

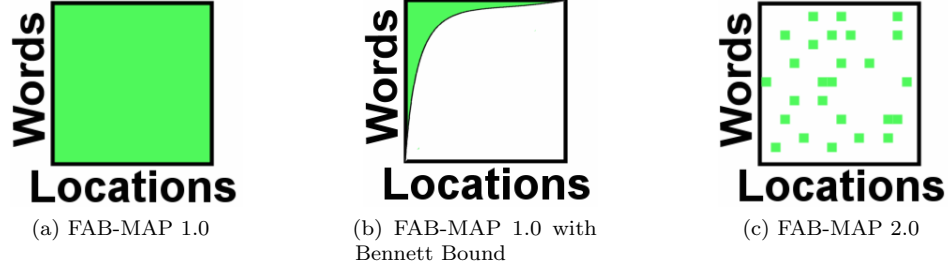


Figure 5.3: Illustration of the amount of computation performed by the different models. The shaded region of each block represents the appearance likelihood terms $p(z_q|z_{p_q}, L_i)$ which must be evaluated. In (a), FAB-MAP 1.0 (Chapter 3), the likelihood must be computed for all words in all locations in the map. Using the Bennett bound approximate inference procedure defined in Chapter 4, unlikely locations are discarded during the computation, yielding the evaluation pattern shown in (b). The restrictions imposed in FAB-MAP 2.0 allow a fully sparse evaluation, (c).

so that $p(e_q|L_i)|_0$ must take the same value for all locations; it is clamped at the value $p(e_q|L_i)|_{\{0\}}$. This restriction enables an efficient likelihood calculation, illustrated in Figure 5.4.

Consider the calculation of one term of the observation likelihood, as per Equation 3.13, across all locations in the map. That is, we wish to compute the term $p(z_q|z_{p_q}, L_i)$ for some visual word q , for all L_i in the map. Recalling Chapter 3, the term is given by

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

In the unrestricted model, this will involve computing one term for each location, as illustrated in Figure 5.4a.

L_1	L_2	L_3	L_4
$p(z_q z_{p_q}, L_1)$	$p(z_q z_{p_q}, L_2)$	$p(z_q z_{p_q}, L_3)$	$p(z_q z_{p_q}, L_4)$

In the restricted model, Figure 5.4b, the term takes a single common value for all locations where word q was not previously observed (in those locations, the word exists with probability $p(e_q|L_i)|_0$, which determines the value of Equation 3.13. We denote this value by $p(z_q|z_{p_q}, L)|_0$).

$p(z_q z_{p_q}, L_1)$	$p(z_q z_{p_q}, L_2) _0$	$p(z_q z_{p_q}, L_3) _0$	$p(z_q z_{p_q}, L_4)$
-----------------------	--------------------------	--------------------------	-----------------------

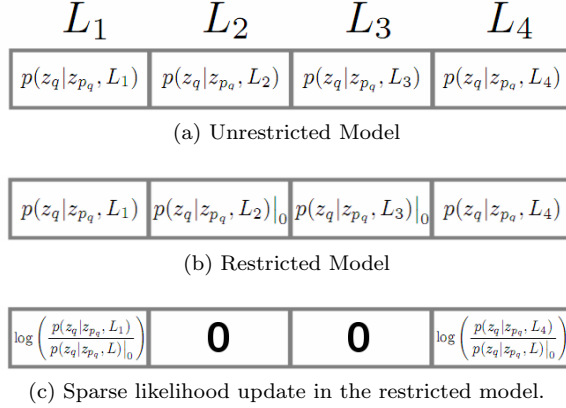


Figure 5.4: Illustration of the calculation of one term of Equation 3.12, i.e. the observation likelihood due to a particular word in the vocabulary. This example shows a map with four locations. The total observation likelihood for a given location, $p(Z|L)$, is calculated by evaluating the illustrated 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).

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 5.4c).

$\log \left(\frac{p(z_q z_{p_q}, L_1)}{p(z_q z_{p_q}, L) _0} \right)$	0	0	$\log \left(\frac{p(z_q z_{p_q}, L_4)}{p(z_q z_{p_q}, L) _0} \right)$
--	----------	----------	--

This allows for efficient implementation using an inverted index. The terms shown in Figure 5.4c 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 identical¹. The “fixed” terms $p(e_q|L_i)|_0$ 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 3.23, updating the beliefs $p(e_q|L_i)$. Effectively this amounts to capturing the

¹Assuming the detector model does not change with time.

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 3.5, with each new observation associated with the location defining a new such model. This change is necessary because of the restrictions placed on scene element existence probabilities, but is also largely beneficial. In addition to improving our ability to deal with multi-modal location appearance, it also makes data association a reversible operation, and makes the implementation of a scheme that maintains multiple hypotheses over data association decisions (e.g. [128]) very 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.

Algorithm 5.1 gives pseudocode for the calculation. 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 5.1, $\log\left(\frac{p(z_q=s_q|z_{p_q}=s_{p_q},L_i)}{p(z_q=s_q|z_{p_q}=s_{p_q},L)}\right)_0$, have four possible states:

$$\begin{aligned}
\text{Case 1 : } & (s_q = 1, s_{p_q} = 1) \\
\text{Case 2 : } & (s_q = 1, s_{p_q} = 0) \\
\text{Case 3 : } & (s_q = 0, s_{p_q} = 1) \\
\text{Case 4 : } & (s_q = 0, s_{p_q} = 0)
\end{aligned} \tag{5.1}$$

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

Algorithm 5.1 Log-likelihood update 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[ $L_i$ ] +=  $\log\left(\frac{p(z_q=s_q|z_{p_q}=s_{p_q},L_i)}{p(z_q=s_q|z_{p_q}=s_{p_q},L)_0}\right)$ 
```

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 5.2. In our experiments using a 100,000 word vocabulary, we observed an order of magnitude speed increase with this approach.

5.2.3 Geometric Verification

While a navigation system based entirely on the bag-of-words likelihood is possible (Chapter 3), we have found in common with others [122] 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 $p(Z_k|L_i, \mathcal{Z}^{k-1})p(L_i|\mathcal{Z}^{k-1})$) and to the 100 most likely samples (the location models used to

Algorithm 5.2 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  $z_q = 0, \forall q$ 
//This can be thought of as the sum of "default votes"  $D_q$ 
//for each observed word at the location
// $D_q = \log\left(\frac{p(z_q=0|z_{pq}=0, L_i)}{p(z_q=0|z_{pq}=0, L)}\right)_0$ 
//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[ $L_i$ ] +=  $\log\left(\frac{p(z_q=1|z_{pq}=s_{pq}, L_i)}{p(z_q=1|z_{pq}=s_{pq}, L)}\right)_0 - D_q$ 

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[ $L_i$ ] +=  $\log\left(\frac{p(z_q=0|z_{pq}=1, L_i)}{p(z_q=0|z_{pq}=1, L)}\right)_0 - D_q$ 

```



Figure 5.5: 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.

evaluate the normalizing term $p(Z_k | \mathcal{Z}^{k-1})$. For each of these locations we check geometric consistency with the current observation using RANSAC [46]. 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 5.5. 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.

5.2.4 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 [122], where, at the beginning of each k-means iteration, a randomized forest of kd-trees [142, 105] 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 [114] in terms of visual vocabulary retrieval performance.

As k-means clustering typically converges only to a 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 [69]. Examples are shown in Figure 5.6.

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 [35], 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

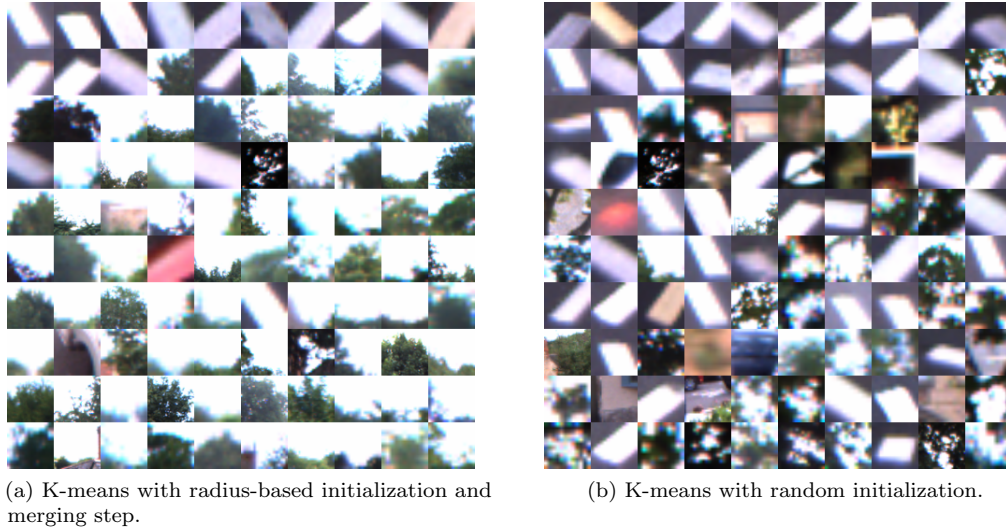


Figure 5.6: The 100 most common visual words in the vocabularies used for the car-based experiments, showing one exemplar per word. With random initialization (Sub-figure (b)) k-means tends to over-segment the densest regions of feature space, leading to a visual vocabulary with many highly similar visual words (in this case, many words corresponding to near-identical views of road markings). Using radius-based initialization and cluster merging (Sub-figure (a)) produces a visual vocabulary with words that are better separated in feature space.

random location.

The modified clustering gives a robust boost to system performance (Figure 5.7). 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 5.4, the relevant graph would require 80 GB of storage. Happily, there is an efficient algorithm for learning Chow Liu trees when the data of interest is sparse [92]. Meilă’s algorithm has complexity $O(s^2 \log s)$,

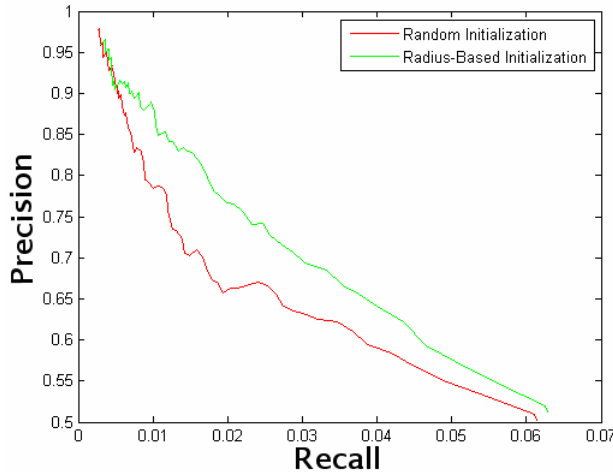


Figure 5.7: Precision-recall curves showing the effect of k-means initialization on overall performance. Performance shown is for a subset of the 1,000 km data set, ranked according to the baseline FAB-MAP 2.0 model (i.e. not including motion model or geometric check). Detector terms were tuned separately 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.

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 5.4 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 [38].

5.3 Data Set

For a truly large scale evaluation of the system, the experiments in this chapter make use of a 1,000 km data set. The data was collected by a car-mounted sensor array (see Appendix A), and consists of omni-directional imagery from a Point Grey Ladybug2, 20Hz stereo imagery from a Point Grey

Bumblebee², 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 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 3.21. The area where this training set was collected did not overlap with that of the test data sets. A second smaller 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 and could also be used for testing cross-season matching performance. The data sets are summarized in Table 5.1.

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

²Not used in these results.

Table 5.1: 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

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.

All data sets are available to researchers upon request.

5.4 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.

5.4.1 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 5.2.4 and the 1,921 image training set described in Section 5.3. 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 3.6), 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 $p(z = 1|e = 1) = 0.39$ and $p(z = 1|e = 0) = 0.005$. For the 100,000 word vocabulary, the values were $p(z = 1|e = 1) = 0.2$ and $p(z = 1|e = 0) = 0.005$. The likelihood smoothing term σ introduced in Section 3.2.4 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 5.8).

We report precision-recall metrics for the system. The evaluation metrics are the same as those outlined in Chapter 3 (Section 3.4.2). To recap, 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. 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.

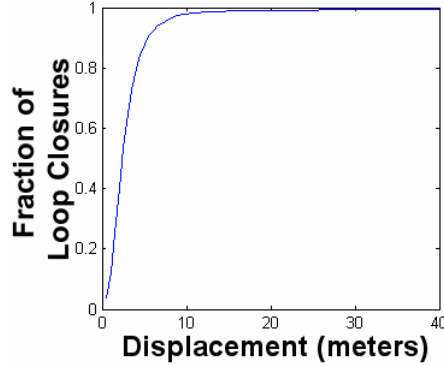


Figure 5.8: The typical displacement in meters between two images identified as loop closures. While any pair separated by less than 40 m is accepted as a true positive, because the ground truth separation can occasionally be this large, 89% of detected loop closures are separated by less than 5m, and 98% by less than 10 m.

5.4.2 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 data sets are shown in Figure 5.9, and given numerically in Table 5.2. Loop closing performance is also visualized in the maps shown in Figures 5.14 and 5.15. Loop closures are often detected even in the presence of large changes in appearance, typical examples are shown in Figures 5.16 and 5.17. 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 5.9 and presented numerically in Table 5.2. 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 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 5.10 and Table 5.3. In common with other authors [114, 122], 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. The reason for this effect 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.

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 5.20). 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 5.2 and 5.3. 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 5.14), 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 100% and 99% precision. This drop is caused

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 5.2: 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 5.9. “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.

by particularly challenging cases of perceptual aliasing, such as encountering rare-but-repetitive objects in environment. Figure 5.18 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 5.20 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 5.18. 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 5.18b because the matched object is a vehicle.

5.4.3 Comparison to tf-idf

Term-frequency inverse-document-frequency (tf-idf) is a standard ranking metric used in most existing visual search engines [144, 122, 63, 87]. 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

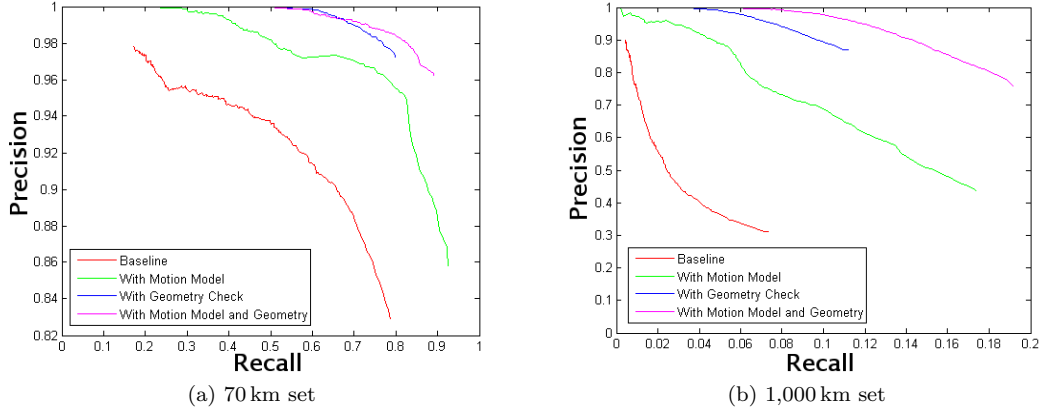


Figure 5.9: Precision-recall curves showing the effect of the different system components on performance. Note the scaling on the axes. Results shown are for the 100k vocabulary with Chow Liu tree. Relative performance in other configurations is similar. “Baseline” refers to the system without the geometric check and with a uniform position prior at each timestep. “Motion model” includes the position prior $p(L_i|Z^{k-1})$, 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.

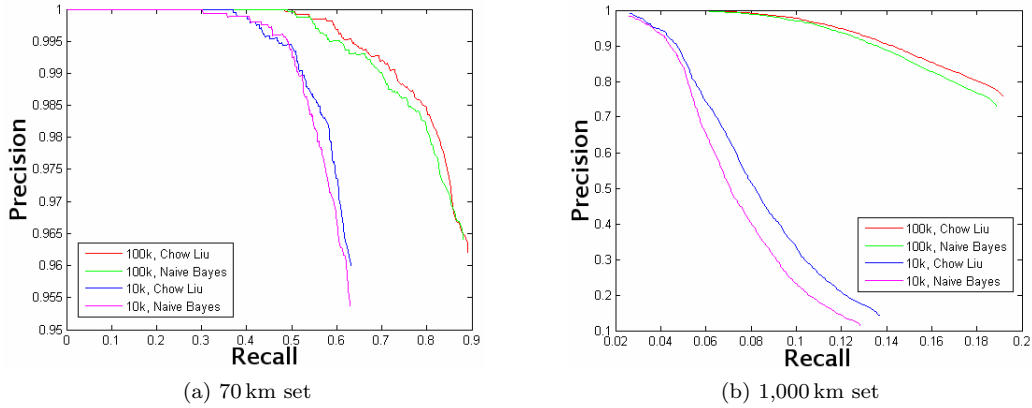


Figure 5.10: Precision-recall curves showing the effect of the vocabulary size and the Chow Liu tree on performance. Note the scaling on the axes. Performance shown includes motion model and geometric check. Performance increases strongly with vocabulary size. The Chow Liu tree also increases performance, 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 5.3: 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 5.10. 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 be masked by a small number of outliers.

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 $p(Z_k|L_i)^3$. 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 [63] and have verified that our implementation gives results identical to those reported there.

Precision-recall curves showing relative performance are given in Figure 5.11. 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 [65] or using pivoted normalized document lengths [87] 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 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

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

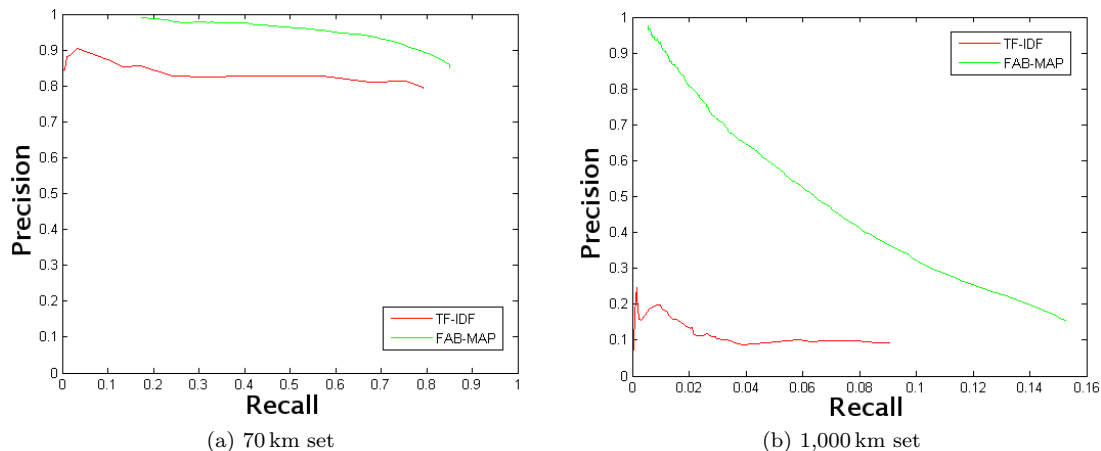


Figure 5.11: Comparison to tf-idf ranking. FAB-MAP substantially outperforms this standard ranking metric, particularly on the larger data set. To ensure the fairest possible comparison, these performance figures relate to a retrieval-only task that excluded new place detection. See Section 5.4.3 for details.

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 (see Chapter 6). Of course, the existing model also substantially out-performs tf-idf for our application of interest.

5.4.4 Timing

Timing performance is presented in Figure 5.13. 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 5.12. The cost is dominated by 423 ms for SURF. Recent GPU-based implementations can largely eliminate this overhead [28]. However, even including feature detection, our real time requirement of 480 ms could be achieved by simply spreading the processing over two cores.

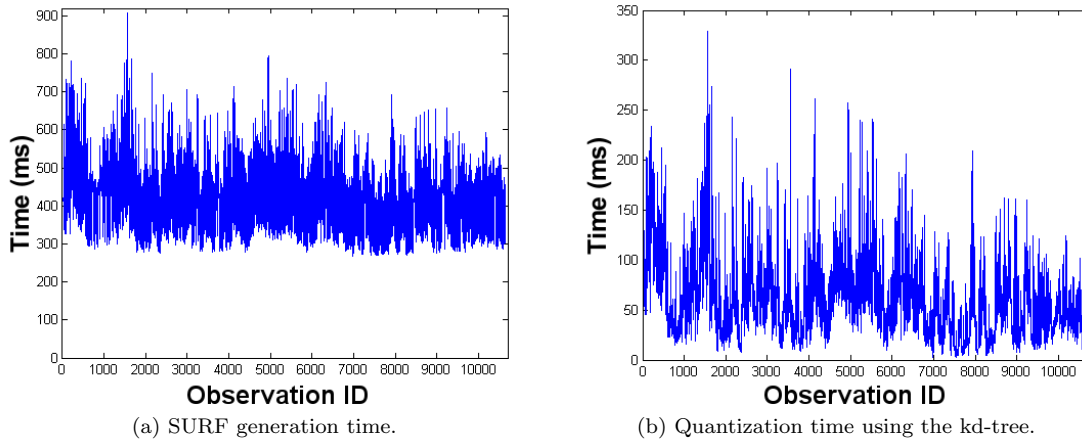


Figure 5.12: Bag-of-words generation time (per Ladybug2 panoramic image) for a representative sample of the 1,000 km data set, using the 100k vocabulary. The time is dominated by SURF generation, (a), which takes 423 ms on average. Quantization using the randomized kd-trees, (b), takes on average 60 ms.

5.4.5 Comparison to Original System

In comparison to the original system described in Chapter 3, 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.

5.5 Chapter Summary

This chapter 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

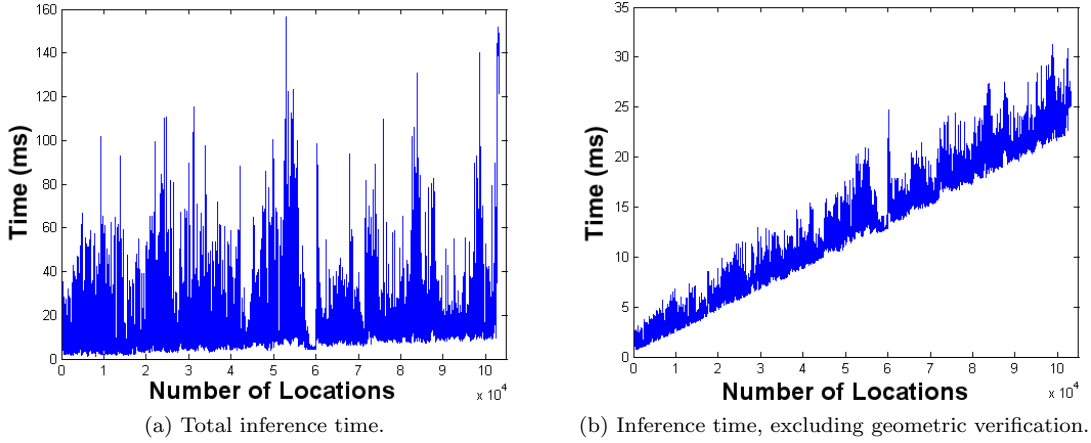
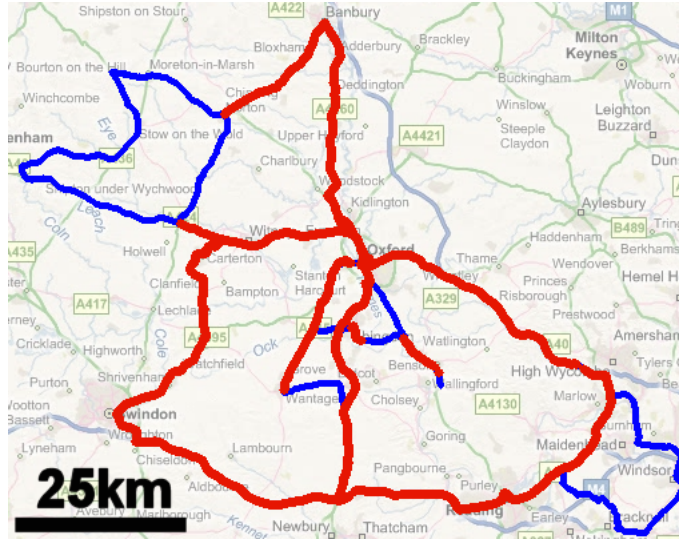
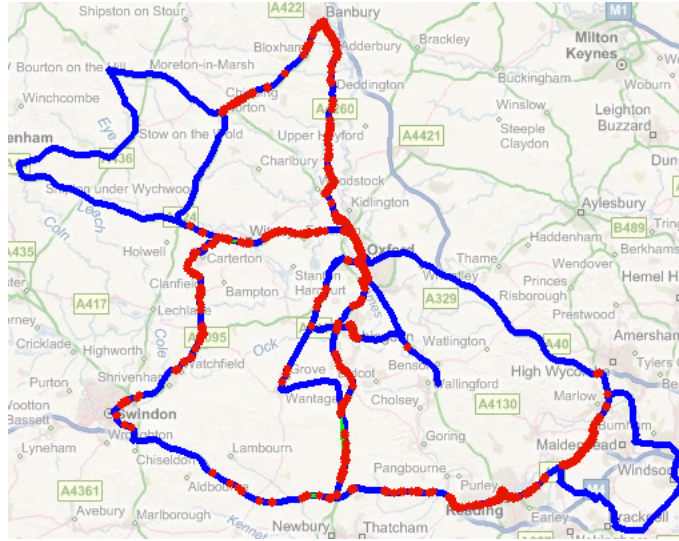


Figure 5.13: Filter update times on the 1,000 km data set for the 100k vocabulary, (a). Mean filter update time is 14 ms and maximum update time is 157 ms. The cost is dominated by the RANSAC geometric verification, which has $O(1)$ complexity. The core ranking stage excluding RANSAC, (b), exhibits linear complexity but with a very small constant - taking 25 ms on average with 100,000 locations in the map.

approach. The chapter 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.



(a)

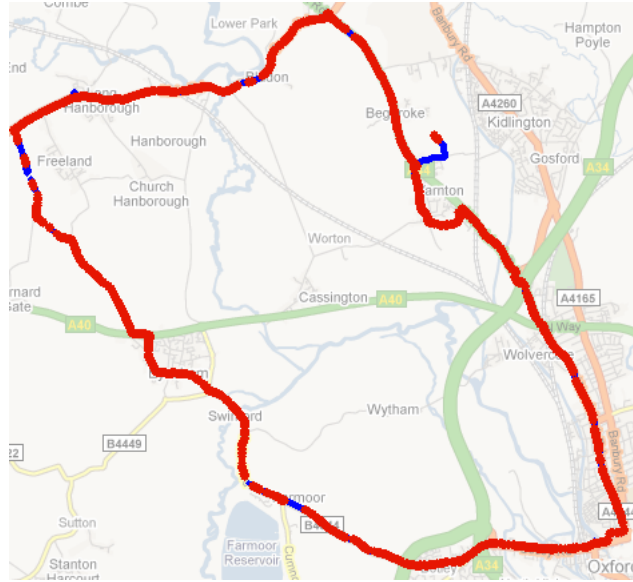


(b)

Figure 5.14: Loop closure maps for the 1,000 km data set. 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)

Figure 5.15: Loop closure maps for the 70 km data set. 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%.



Figure 5.16: 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%.



Figure 5.17: 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)



Figure 5.18: 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)



Figure 5.19: 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 $p(Z_k | \mathcal{Z}^{k-1})$ (see Section 3.2.4). 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.

Figure 5.20: 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.

Chapter 6

Discussion

This chapter summarizes the contribution of the thesis and discusses possible directions for future research.

6.1 Summary of Contribution

This thesis has made the following contributions to the state of the art:

1. **The first probabilistic framework for localization and mapping using appearance that handles perceptual aliasing (Chapter 3).** We have introduced a natural model for the appearance-based navigation task, which empirically achieves excellent performance for loop closure detection. In addition it provides probabilities for all loop closure assignments, easing integration with metric SLAM systems. Our model is also applicable to some types of image retrieval task. We have compared it to the standard tf-idf ranking function and show superior performance for place recognition.
2. **A novel approach to dealing with correlations between visual words (Chapter 3).** To our knowledge, this is the first time an explicit approximation to the joint distribution of visual word occurrences has been considered in the context of large-scale image retrieval. The technique

we have introduced for modelling word dependencies scales even to large visual vocabularies, and we have demonstrated that it improves performance.

3. **A principled method for deciding when to initialize a new location in the map (Chapter 3).** By computing an approximation of the partition function in our model, we naturally obtain the probability that the current observation comes from a new location not already in the map. To our knowledge this is the first method that goes beyond simple heuristics such as fixed similarity thresholds. We find that our approach is a dramatic improvement over these simpler techniques, particularly in that it handles challenging cases such as perceptual aliasing without any special logic.
4. **An approximate inference technique based on concentration inequalities (Chapter 4).** We show that in return for a small user-controlled probability of generating an incorrect result, inference in our model can be greatly accelerated. Speed improvements in the range 25x-50x were typical for our data, with minimal degradation in accuracy. The approximate inference technique is general in its applicability and should find use beyond the specific problem considered in this thesis.
5. **An efficient version of our model suitable for use with an inverted index (Chapter 5).** By modifying the model to allow an implementation based on inverted indices, we enable the use of our technique on longer trajectories and with larger visual vocabulary sizes. The final system is capable of real time loop closure detection on trajectories over 1,000 km in length.
6. **Four public benchmark data sets (Chapter 5).** This includes a 1,000 km data set which is the most extensive benchmark available. The data set contains 20 Hz stereo imagery, omni-directional imagery every 5-10 m and GPS ground truth.

6.2 Future Work

There are many directions in which the system could be developed. Some interesting possibilities include:

1. **Making greater use of metric information.** In this thesis we have chosen not to use metric information in order to concentrate on what can be achieved with appearance. In a practical system, it seems likely that performance could be improved considerably by replacing our simple topological motion model with one based on real vehicle motion data, for example from a visual SLAM system. While metric position estimates may not be reliably globally (the main motivation for this thesis), they are typically highly accurate locally. This relative motion information is precisely what is required for our position prior. Several authors have recently begun to explore systems which couple metric and appearance information more tightly; for example see [42, 11, 99, 115, 91, 110].
2. **Maintain a distribution over topologies.** As discussed in Chapter 3, the data association step of our method is error-prone. The solution we proposed is effective but not ideal. A better solution might be to maintain a pdf over topologies, as proposed in [128]. An alternative approach is the solution proposed by Olson [115], which selects a single consistent set of loop closures from a larger set of noisy candidates by jointly considering appearance similarity and relative metric information.
3. **Use of depth information.** Often a robot is equipped with either a stereo camera or a laser range finder, which provides depth information. Unlike the 2D geometry of interest points in an image, their relative 3D geometry in space is an invariant property of a location. It should be possible to integrate this relative 3D distance information directly into the ranking function, rather than using 2D geometry in post-verification step. Some initial results by colleagues in our lab suggest that this can significantly improve system performance.
4. **Learning more of the parameters of our generative model.** In this thesis we have used a user-specified constant for $p(z_i|e_i)$, the detector model. In principle this term could instead be learned, and indeed it could be learned as a conditional probability so as to incorporate additional information. For example, certain features are more trustworthy than others — we might not expect words that correspond to parts of dynamic objects such cars or pedestrians to be re-observed at loop closure — and this is naturally captured in a $p(z_i|e_i)$ term which is not constant

across the vocabulary. Lower level properties of an interest point, such its saliency score or vertical height within an image might also be predictive of stability. However, it may be that the general noise in interest point detection dominates these effects. If so, learning might only be worthwhile in combination with a tracking stage to select stable interest points, as was described in [144]. There are other effects that could also be captured by control of $p(z_i|e_i)$, for example if an image is blurry or poorly exposed then $p(z_i|e_i)$ could be dynamically adjusted so as to capture the fact that we expect to see fewer visual words than in a high quality image of the location.

5. **Robustness to rare-but-repetitive features.** It seems likely that the most persistent challenge for appearance-based systems will be dealing with cases similar to those shown in Figure 5.18, where the robot encounters some rare-but-repetitive environmental feature such as a brand logo, advertising billboard, etc. It does not seem feasible to learn a-priori that these are repetitive features, especially in the case of advertising billboards which change regularly. Possible solutions include relying more heavily on sequences of images, detecting false loop closures on the basis of the metric map, or a classifier-based approach to recognising image regions that are likely to contain such untrustworthy features.
6. **Learning stronger approximations to the visual word distribution.** As mentioned in Chapter 3, there is nothing that requires us to use a Chow Liu tree to approximate the joint density over visual words, and it would be interesting to compare performance to alternative learning algorithms. We suspect that the performance improvements that can be gained in this way are modest, given the relative performance of the Naive Bayes and Chow Liu approximations. However, this is only a weak intuition and it would be illuminating to examine the issue thoroughly. In particular, mixture models such as Latent Dirichlet Allocation [12] have quite different representational strengths to the approach we have used, and a comparison would be interesting.
7. **Generalization to non-binary word occurrence.** Our model discards information on the number of visual words present in an image. We chose this simplified representation in order to make the problem of learning correlations between words more tractable. However, a generalization to case of non-binary word occurrence would be interesting and likely offer some performance

improvement. A suitable model, though for the Naive Bayes case only, was described in [128].

8. **Online learning.** At present our visual vocabulary and associated probabilistic model is learned once, and remains static thereafter. It should be possible to update both the visual vocabulary and the model online. Updating the marginal and conditional probabilities that define the probabilistic model is straightforward, though care needs to be taken that the observations used are independent. Incrementally adding new visual words to the vocabulary was explored in [42, 2]. Updating the structure of the Chow Liu tree is efficient for the case of addition of new visual words to the vocabulary [147], but not for the case in which the estimate of the mutual information between existing words changes [133]. Thus relearning the Chow Liu tree might best be achieved as a periodic batch process. We note, however, that our experience with the system’s generalization performance suggests that online learning is not a critical issue.

6.3 Concluding Remark

The appearance-based approach to loop closure detection has proven to be remarkably successful. It now seems firmly established as an essential component of SLAM systems. This thesis set out to provide a coherent probabilistic framework in which to treat appearance data. We hope that our approach will be useful for improving the robustness of appearance-based systems and provide a natural way in which to integrate them with metric SLAM techniques.

Appendix A

Data Collection Platforms

This appendix includes details of the vehicles used to collect the data used in this thesis.

The data sets used in Chapters 3 and 4 were collected with a modified iRobot ATRV (*“Marge”*), shown in Figure A.1. The camera used is a Marlin unit from Allied Vision Technologies, mounted on pan-tilt mechanism from Directed Perception.

The 70 km and 1,000 km data sets used in Chapter 5 were collected with a car-mounted sensor rig shown in Figure A.2. 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.



Figure A.1: Robot (*"Marge"*) used to collect the data sets used in Chapters 3 and 4.



Figure A.2: Vehicle and sensor rig used to capture the 70 km and 1,000 km data sets used in Chapter 5.

Appendix B

Chow Liu Trees

This appendix illustrates the form and performance of the Chow Liu tree as a general density estimator. For the purposes of explanation we use a system with eight binary features, thus 256 possible distinct observations. Features one to six correspond to the faces of a die. For each observation, the die is rolled and the feature corresponding to the shown face is assigned value 1, the others are assigned value 0. Features seven and eight are generated in a similar way by tossing a coin.

Figure B.1 shows the graphical model for this system, and the ground truth probability of each of the possible 256 observations (Many of these have probability 0 since they correspond to impossible states, such as observing the two faces of the coin simultaneously.)

1000 random samples were generated from this system and used to construct approximations to the pdf. The Chow Liu approximation is shown in Figure B.2. The naive Bayes approximation is shown in Figure B.3.

The three distributions are compared in Figure B.4. The Chow Liu approximation is visibly closer to the true distribution than the naive Bayes approximation. The Kullback-Leibler divergence from the naive approx to the ground truth is 1.61, whereas the Chow Liu approximation has a KL divergence of 0.748.

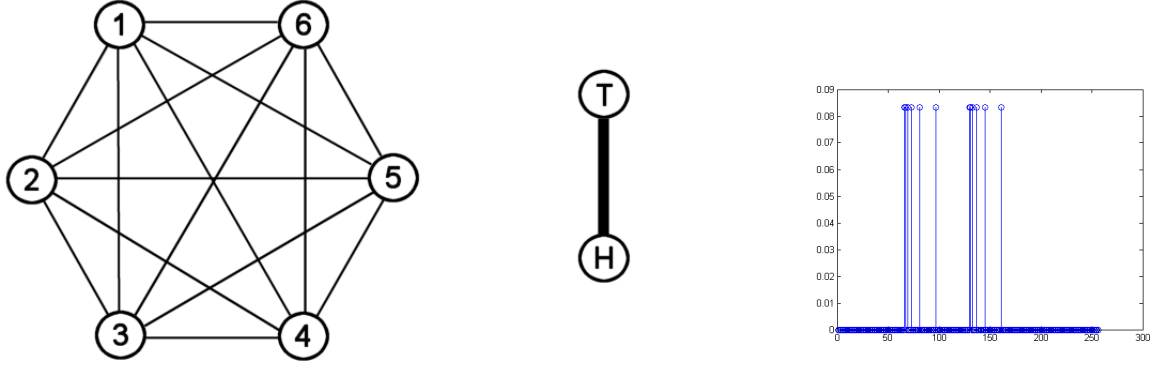


Figure B.1: The underlying system, with its graphical model (left), and the probability of each of the 256 possible observations (right). The thickness of the line in the graphical model corresponds to the mutual information between the variables. There is 1 bit of mutual information between the sides of the coin, and approximately 0.05 bits of mutual information between the faces of the die. The die and coin are independent so share no mutual information.

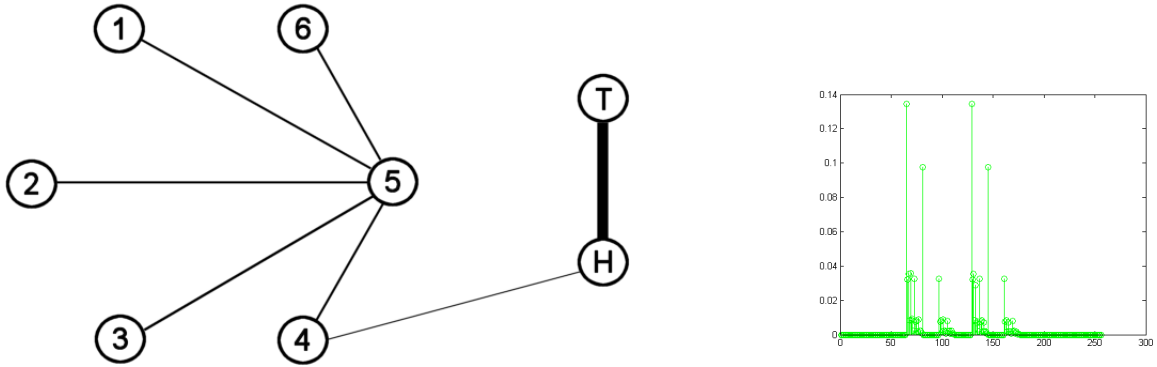


Figure B.2: Chow Liu tree estimated from 1000 samples. Note that the many impossible observations are correctly assigned very low probability. The spurious edge in the graphical model between 4 and H is due to the fact that mutual information between these variables, as estimated from 1000 samples, is not exactly zero. However, the variables have been estimated to be close to independence – the conditional probability $p(H|4) \approx P(H)$, so the effect of this spurious edge in the model is minimal.

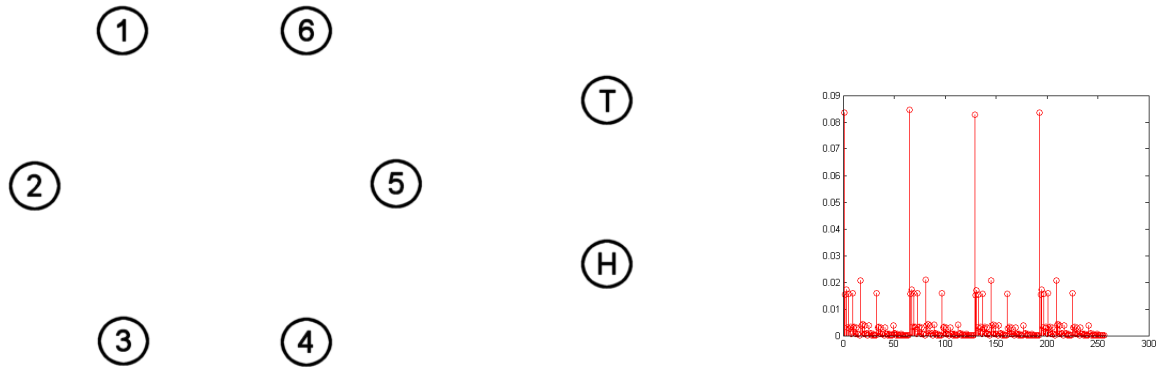


Figure B.3: Naive Bayes. All variables assumed independent. This assumption is far from true in this system, so many impossible observations are estimated to have significant probability (right).

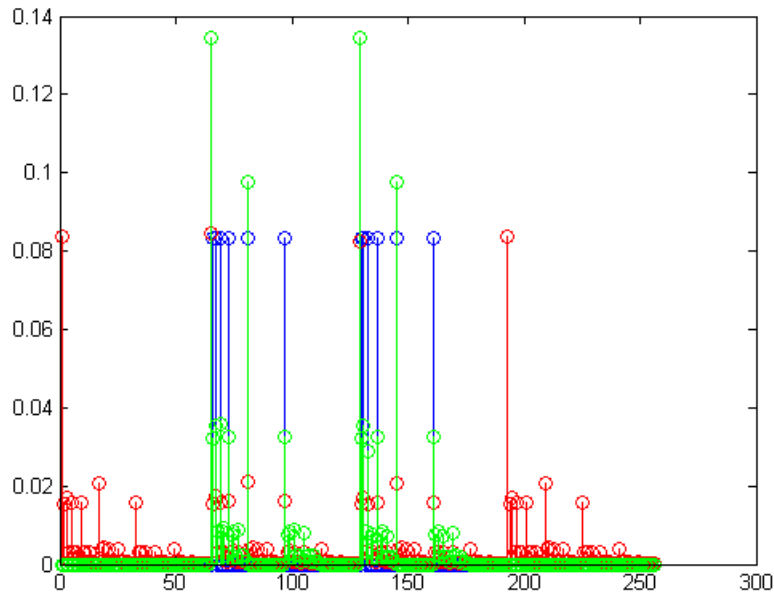


Figure B.4: Comparison of the three distributions – ground truth (blue), Chow Liu (green) and naive Bayes (red). The KL divergence from the naive approx to the ground truth is 1.61, whereas the Chow Liu approximation has a KL divergence of 0.748.

Appendix C

Manipulations

C.1 FAB-MAP Model

This appendix presents the manipulations involved in the expansion of Equation 3.16 from Section 3.2.4. For compactness of notation, in this appendix $z_q = s_{z_q}$ will be denoted z_q and $z_q = \overline{s_{z_q}}$ by $\overline{z_q}$, etc.

We seek to express the term $p(z_q|e_q, z_p)$ as a function of the known quantities $p(z_q)$, $p(z_q|e_q)$, $p(z_q|z_p)$.

Applying Bayes Rule:

$$p(z_q|e_q, z_p) = \frac{p(e_q|z_q, z_p)p(z_q|z_p)}{p(e_q|z_p)}$$

now expanding $p(e_q|z_p)$ as

$$p(e_q|z_p) = p(e_q|z_q, z_p)p(z_q|z_p) + p(e_q|\overline{z_q}, z_p)p(\overline{z_q}|z_p)$$

and making the approximation $p(e_q|z_q, z_p) \approx p(e_q|z_q)$, which is closely related to the approximation we make between Equations 3.14 and 3.15, the expression becomes

$$\begin{aligned}
p(z_q|e_q, z_p) &\approx \frac{p(e_q|z_q)p(z_q|z_p)}{p(e_q|z_q)p(z_q|z_p) + p(e_q|\bar{z}_q)p(\bar{z}_q|z_p)} \\
&= \left(1 + \frac{p(e_q|\bar{z}_q)p(\bar{z}_q|z_p)}{p(e_q|z_q)p(z_q|z_p)}\right)^{-1}
\end{aligned}$$

Now

$$p(e_q|z_q) = \frac{p(z_q|e_q)p(e_q)}{p(z_q)}$$

and similarly for $p(e_q|\bar{z}_q)$, so

$$\frac{p(e_q|\bar{z}_q)}{p(e_q|z_q)} = \frac{p(\bar{z}_q|e_q)p(z_q)}{p(z_q|e_q)p(\bar{z}_q)}$$

substituting back yields

$$p(z_q|e_q, z_p) \approx \left(1 + \frac{\alpha}{\beta}\right)^{-1}$$

where

$$\alpha = p(z_q)p(\bar{z}_q|e_q)p(\bar{z}_q|z_p)$$

$$\beta = p(\bar{z}_q)p(z_q|e_q)p(z_q|z_p)$$

C.2 Bennett Bound

The form of the Bennett inequality used in Chapter 4 differs notationally from the usual statement of the inequality. For completeness, the relevant manipulations are presented here.

Beginning with the specialized form of the Bennett inequality that applies to sums of random variables with symmetric distributions, as per [9]¹:

¹Note that the typesetting in [9] is slightly confusing, as the bracketing might seem to indicate that the argument of the exponential is $\left(\frac{\sigma}{M}\right)^2$. However, the argument is in fact all following terms, i.e. $\left(\frac{\sigma}{M}\right)^2 \left(\cosh(cM) - 1 - cM\frac{M}{\sigma}t\right)$.

$$p(S \geq t\sigma) < \exp\left(\left(\frac{\sigma}{M}\right)^2 \left(\cosh(cM) - 1 - cM \frac{M}{\sigma} t\right)\right)$$

with

$$t = \frac{\sigma}{M} \sinh(cM)$$

Our statement of the inequality simply changes some variables for a more convenient expression.

Define:

$$\Delta = t\sigma$$

so

$$c = \frac{\sinh^{-1}(\frac{\Delta M}{\sigma^2})}{M}$$

and define:

$$f(\Delta) = \sinh^{-1}\left(\frac{\Delta M}{\sigma^2}\right)$$

Substitution yields

$$p(s \geq \Delta) < \exp\left(\left(\frac{\sigma}{M}\right)^2 \left(\cosh(f(\Delta)) - 1 - \frac{M\Delta}{\sigma^2} f(\Delta)\right)\right)$$

Bibliography

- [1] P. Abbeel, D. Koller, and A. Ng. Learning factor graphs in polynomial time and sample complexity. *The Journal of Machine Learning Research*, 7:1743–1788, 2006.
- [2] 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.
- [3] T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (SLAM): Part II. *IEEE Robotics & Automation Magazine*, 13(3):108–117, 2006.
- [4] Y. Bar-Shalom, X.R. Li, and T. Kirubarajan. *Estimation with applications to tracking and navigation*. Wiley-Interscience, 2001.
- [5] C.W. Baum and V. Veeravalli. A sequential procedure for multihypothesis testing. *IEEE Transactions on Information Theory*, 40(6):1994–2007, 1994.
- [6] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.
- [7] 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.
- [8] J. Beis and D. Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1997.

- [9] G. Bennett. Probability inequalities for the sum of independent random variables. *Journal of the American Statistical Association*, 57:33–45, March 1962.
- [10] Y.M. Bishop, S.E. Fienberg, and P.W. Holland. *Discrete Multivariate Analysis: Theory and Practice*. The MIT Press, June 1977.
- [11] 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.
- [12] D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.
- [13] M. Bosse, P. Newman, J. J. Leonard, and S. Teller. SLAM in large-scale cyclic environments using the Atlas framework. *International Journal of Robotics Research*, 23:1113–1139, 2004.
- [14] 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.
- [15] 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.
- [16] S. Boucheron, G. Lugosi, and O. Bousquet. *Concentration Inequalities*, volume Lecture Notes in Artificial Intelligence 3176, pages 208–240. Springer, Heidelberg, Germany, 2004.
- [17] M. Bowling, D. Wilkinson, and A. Ghodsi. Subjective mapping. In *Twenty-First National Conference on Artificial Intelligence*, Boston, Massachusetts, USA, July 2006. AAAI Press.
- [18] M. Bowling, D. Wilkinson, A. Ghodsi, and A. Milstein. Subjective localization with action respecting embedding. In *Proceedings of the International Symposium of Robotics Research*, 2005.
- [19] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30, 1998.
- [20] 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.

- [21] J.A. Castellanos, R. Martinez-Cantin, J.D. Tardos, and J. Neira. Robocentric map joining: Improving the consistency of EKF-SLAM. *Robotics and Autonomous Systems*, 55(1):21–29, 2007.
- [22] A. Checheta, C. Guestrin, JC Platt, D. Koller, Y. Singer, and S. Roweis. Efficient principled learning of thin junction trees. *Advances in Neural Information Processing Systems*, 20:273–280, 2007.
- [23] H. Choset and K. Nagatani. Topological simultaneous localization and mapping (SLAM): Toward exact localization without explicit localization. *IEEE Transactions on Robotics and Automation*, 17(2):125 – 137, April 2001.
- [24] C.K. Chow and C.N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, IT-14(3), May 1968.
- [25] O. Chum, M. Perdoch, and J. Matas. Geometric min-Hashing: Finding a (Thick) Needle in a Haystack. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2009.
- [26] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *International Conference on Computer Vision*, 2007.
- [27] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Second Edition*. The MIT Press, September 2001.
- [28] N. Cornelis and L. Van Gool. Fast scale invariant feature detection and matching on programmable graphics hardware. In *CVPR 2008 Workshop CVGPU*, 2008.
- [29] D. Crisan and A. Doucet. A survey of convergence results on particle filtering methods for practitioners. *IEEE Transactions on signal processing*, 50(3):736–746, 2002.
- [30] M. Cummins and P. Newman. Probabilistic appearance based navigation and loop closing. In *Proc. IEEE International Conference on Robotics and Automation (ICRA’07)*, Rome, April 2007.
- [31] M. Cummins and P. Newman. Accelerated appearance-only SLAM. In *Proc. IEEE International Conference on Robotics and Automation (ICRA’08)*, Pasadena, California, April 2008.

- [32] 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.
- [33] 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.
- [34] J. Cussens. Bayes and Pseudo-Bayes Estimates of Conditional Probabilities and Their Reliability. *Lecture Notes in Computer Science*, pages 136–136, 1993.
- [35] S. Dasgupta and P.M. Long. Performance guarantees for hierarchical clustering. *Journal of Computer and System Sciences*, 70(4):555–569, 2005.
- [36] A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *International Conference on Computer Vision*, Nice France, October 2003.
- [37] F. Dellaert. Square root SAM: Simultaneous location and mapping via square root information smoothing. In *Proceedings of Robotics: Science and Systems*, Cambridge, USA, June 2005.
- [38] R. Dementiev, L. Kettner, and P. Sanders. STXXL: Standard template library for XXL data sets. *Software: Practice and Experience*, 38(6), 2008.
- [39] R. Dementiev, P. Sanders, D. Schultes, and J. Sibeyn. Engineering an external memory minimum spanning tree algorithm. In *Proceedings 3rd International Conference on Theoretical Computer Science*, 2004.
- [40] 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.
- [41] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: Part I. *IEEE Robotics & Automation Magazine*, 13(2):99–110, 2006.
- [42] E. Eade and T. Drummond. Unified loop closing and recovery for real time monocular slam. In *Proc. 19th British Machine Vision Conference*, 2008.

- [43] G. Elidan and S. Gould. Learning Bounded Treewidth Bayesian Networks. *Journal of Machine Learning Research*, 9:2699–2731, 2008.
- [44] F. Ferreira, V. Santos, and J. Dias. Integration of Multiple Sensors using Binary Features in a Bernoulli Mixture Model. *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 104–109, 2006.
- [45] D. Filliat. A visual bag of words method for interactive qualitative localization and mapping. In *IEEE International Conference on Robotics and Automation*, pages 3921–3926, 2007.
- [46] 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.
- [47] F. Fraundorfer, C. Engels, and D. Nistér. Topological mapping, localization and navigation using image collections. In *International Conference on Intelligent Robots and Systems*, 2007.
- [48] F. Fraundorfer, H. Stewenius, and D. Nister. A binning scheme for fast hard drive based image search. In *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR'07*, pages 1–6, 2007.
- [49] U. Frese, P. Larsson, and T. Duckett. A multilevel relaxation algorithm for simultaneous localisation and mapping. *IEEE Transactions on Robotics*, 21(2):196–207, April 2005.
- [50] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian Network Classifiers. *Machine Learning*, 29(2):131–163, 1997.
- [51] A. Frome, D. Huber, R. Kolluri, T. Bulow, and J. Malik. Recognizing objects in range data using regional point descriptors. In *Proc. European Conf. on Computer Vision*. Springer, 2004.
- [52] T. Goedemé. *Visual navigation*. PhD thesis, Katholieke Universiteit Leuven, December 2006.
- [53] G. Grisetti, S. Grzonka, C. Stachniss, P. Pfaff, and W. Burgard. Efficient estimation of accurate maximum likelihood maps in 3D. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2007.

- [54] G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard. A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. In *Proc. of Robotics: Science and Systems (RSS)*, 2007.
- [55] J. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, pages 318–325, Monterey, California, November 1999.
- [56] C. G. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, pages 147–151, 1988.
- [57] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049, 2000.
- [58] K. Ho and P. Newman. Combining visual and spatial appearance for loop closure detection. *Proceedings of European Conference on Mobile Robotics*, September 2005.
- [59] K. Ho and P. Newman. Multiple map intersection detection using visual appearance. In *International Conference on Computational Intelligence, Robotics and Autonomous Systems*, Singapore, December 2005.
- [60] K.L. Ho and P. Newman. Detecting loop closure with scene sequences. *International Journal of Computer Vision*, 74(3):261–286, September 2007.
- [61] W. Hoeffding. Probability Inequalities for Sums of Bounded Random Variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.
- [62] Klaus-U. Höffgen. Learning and robust learning of product distributions. In *COLT '93: Proceedings of the sixth annual conference on Computational learning theory*, pages 77–83, New York, NY, USA, 1993. ACM.
- [63] 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.

- [64] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometry consistency for large scale image search - extended version. Technical report, INRIA, RR 6709, 2008.
- [65] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. On the burstiness of visual elements. In *IEEE Conference on Computer Vision & Pattern Recognition*, June 2009.
- [66] A. Johnson. *Spin-Images: A Representation for 3-D Surface Matching*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 1997.
- [67] M.I. Jordan, Z. Ghahramani, T.S. Jaakkola, and L.K. Saul. An introduction to variational methods for graphical models. *Mach. Learn.*, 37(2):183–233, 1999.
- [68] S. Julier and J. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls, Orlando, FL*, 1997.
- [69] 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.
- [70] Michael Kaess, Ananth Ranganathan, and Frank Dellaert. Fast incremental square root information smoothing. In *International Joint Conference on Artificial Intelligence*, 2007. To appear.
- [71] 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.
- [72] B. Kröse, N.A. Vlassis, R. Bunschoten, and Y. Motomura. A probabilistic model for appearance-based robot localization. *Image and Vision Computing*, 19(6):381–391, 2001.
- [73] B. Kuipers and Y. Byun. *A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations*, pages 47–63. MIT Press, Cambridge, MA, USA, 1993.
- [74] S. Kullback and R.A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, pages 79–86, 1951.

- [75] J. Kurlbaum and U. Frese. A benchmark data set for data association. Technical Report 017-02/2009, University of Bremen, 2009.
- [76] P. Lamon, I. Nourbakhsh, B. Jensen, and R. Siegwart. Deriving and matching image fingerprint sequences for mobile robot localization. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Seoul, Korea, May 21-26 2001.
- [77] B. Leibe, K. Mikolajczyk, and B. Schiele. Efficient clustering and matching for object class recognition. In *British Machine Vision Conference*, volume 2, 2006.
- [78] J.J. Leonard and P.M. Newman. Consistent, convergent, and constant-time SLAM. In *International Joint Conference on Artificial Intelligence*, Acapulco Mexico, August 2003.
- [79] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1465, 2006.
- [80] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44, 2001.
- [81] A. Levin and R. Szeliski. Visual odometry and map correlation. *IEEE Conference on Computer Vision and Pattern Recognition*, 01:611–618, 2004.
- [82] F. Li and J. Kosecka. Probabilistic location recognition using reduced feature set. In *IEEE International Conference on Robotics and Automation*, Orlando, Florida, 2006.
- [83] D.G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the 7th International Conference on Computer Vision*, pages 1150–1157, Kerkyra, 1999.
- [84] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [85] 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.

- [86] P.C Mahalanobis. On the generalized distance in statistics. In *Proceedings of the National Institute of Sciences of India*, 1936.
- [87] C. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.
- [88] O. Maron and A.W. Moore. Hoeffding races: Accelerating model selection search for classification and function approximation. In Jack D. Cowan, Gerald Tesauro, and Joshua Alspecter, editors, *Advances in Neural Information Processing Systems*, volume 6, pages 59–66. Morgan Kaufmann Publishers, Inc., 1994.
- [89] J. Matas and O. Chum. Randomized RANSAC with sequential probability ratio test. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2005.
- [90] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of British Machine Vision Conference*, 2002.
- [91] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid. A constant time efficient stereo slam system. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2009.
- [92] 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.
- [93] M. Meilă and T. Jaakkola. Tractable Bayesian learning of tree belief networks. *Statistics and Computing*, 16(1):77–92, 2006.
- [94] M. Meilă and M.I. Jordan. Learning with mixtures of trees. *The Journal of Machine Learning Research*, 1:1–48, 2001.
- [95] C. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 18-20 June 2003.
- [96] C. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, (1):63–86, 2004.

- [97] K. Mikolajczyk, B. Leibe, B. Schiele, M.I. Syst, and G. Darmstadt. Local features for object class recognition. In *Tenth IEEE International Conference on Computer Vision, 2005. ICCV 2005*, volume 2, 2005.
- [98] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1):43–72, 2005.
- [99] 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.
- [100] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, Edmonton, Canada, 2002. AAAI.
- [101] Michael Montemerlo, Sebastian Thrun, Daphne Koller, and Ben Wegbreit. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *International Joint Conferences on Artificial Intelligence*, pages 1151–1156, 2003.
- [102] A. Moore. A tutorial on kd-trees. Technical Report 209, University of Cambridge, 1991.
- [103] A.I. Mourikis and S.I. Roumeliotis. A multi-state constraint Kalman filter for vision-aided inertial navigation. In *2007 IEEE International Conference on Robotics and Automation*, pages 3565–3572, 2007.
- [104] O.M. Mozos, C. Stachniss, and W. Burgard. Supervised Learning of Places from Range Data using AdaBoost. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1730–1735, 2005.
- [105] M. Muja and D. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Applications*, 2009.
- [106] J. Neira and J. D. Tardós. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17(6):890–897, 2001.

- [107] P. Newman, M. Chandran-Ramesh, D. Cole, M. Cummins, A. Harrison, I. Posner, and D. Schroeter. Describing, navigating and recognising urban spaces - building an end-to-end SLAM system. In *Proc. of the Int. Symposium of Robotics Research (ISRR)*, Hiroshima, Japan, November 2007.
- [108] P. Newman and K. Ho. SLAM - Loop Closing with Visually Salient Features. *IEEE International Conference on Robotics and Automation*, 18-22 April 2005.
- [109] P. Newman, G. Sibley, M. Cummins, M. Smith, A. Harrison, C. Mei, and I. Reid. Toward life long mapping. In *14th International Symposium on Robotics Research*, 2009.
- [110] 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.
- [111] P. M. Newman, D. M. Cole, and K. Ho. Outdoor SLAM using visual appearance and laser ranging. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Orlando Florida USA, May 2006.
- [112] K. Ni, A. Kannan, A. Criminisi, and J. Winn. Epitomic location recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*, pages 1–8, 2008.
- [113] D. Nistér. Preemptive RANSAC for live structure and motion estimation. *Machine Vision and Applications*, 16(5):321–329, 2005.
- [114] D. Nistér and H. Stewenius. Scalable recognition with a vocabulary tree. In *Conf. Computer Vision and Pattern Recognition*, volume 2, pages 2161–2168, 2006.
- [115] E. Olson. *Robust and Efficient Robotic Mapping*. PhD thesis, Massachusetts Institute of Technology, June 2008.
- [116] E. Olson, J. Leonard, and S. Teller. Fast iterative alignment of pose graphs with poor initial estimates. In *Proceedings of International Conference on Robotics and Automation*, Orlando, Florida, USA, May 2006.

- [117] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies, 1998.
- [118] M.A. Paskin. Thin junction tree filters for simultaneous localization and mapping. Computer Science Division Technical Report CSD-02-1198, University of California, Berkeley, September 2002.
- [119] M.A. Paskin. Thin junction tree filters for simultaneous localization and mapping. In Georg Gottlob and Toby Walsh, editors, *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03)*, pages 1157–1164, San Francisco, CA, 2003. Morgan Kaufmann Publishers.
- [120] L.M. Paz, J.D. Tardos, and J. Neira. Divide and Conquer: EKF SLAM in $O(n)$. *IEEE Transactions on Robotics*, 24(5):1107–1120, 2008.
- [121] M. Perdoch, O. Chum, and J. Matas. Efficient Representation of Local Geometry for Large Scale Object Retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2009.
- [122] 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.
- [123] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008*, pages 1–8, 2008.
- [124] I. Posner, M. Cummins, and P. Newman. Fast probabilistic labeling of city maps. In *Proc. Robotics: Science and Systems*, Zurich, June 2008.
- [125] I. Posner, M. Cummins, and P. Newman. A generative framework for fast urban labeling using spatial and temporal context. *Autonomous Robots*, 26(2-3):153–170, April 2009.

- [126] F.T. Ramos, B. Upcroft, S. Kumar, and H.F. Durrant-Whyte. A Bayesian approach for place recognition. In *IJCAI Workshop on reasoning with Uncertainty in Robotics*, July 2005.
- [127] A. Ranganathan, E. Menegatti, and F. Dellaert. Bayesian inference in the space of topological maps. *IEEE Transactions on Robotics*, 22(1):92–107, 2006.
- [128] Ananth Ranganathan. *Probabilistic Topological Maps*. PhD thesis, Georgia Institute of Technology, 2008.
- [129] Ananth Ranganathan and Frank Dellaert. A Rao-Blackwellized particle filter for topological mapping. In *Proceedings of International Conference on Robotics and Automation*, Orlando, Florida, USA, May 2006.
- [130] M. Ranzato, Y.L. Boureau, and Y. LeCun. Sparse feature learning for deep belief networks. In *Advances in neural information processing systems*, 2007.
- [131] D.B. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, 1979.
- [132] E. Remolina and B. Kuipers. Towards a general theory of topological maps. *Artificial Intelligence*, 152:47–104, 2004.
- [133] J. Roure. Incremental learning of tree augmented naive Bayes classifiers. In *Ibero-American Conference on Artificial Intelligence (IBERAMIA)*, 2002.
- [134] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Proc. 3D Digital Imaging and Modeling*, pages 145–152, 2001.
- [135] G. Schindler, M. Brown, and R. Szeliski. City-Scale Location Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, 2007.
- [136] C. Schmid. Constructing models for content-based image retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2001.

- [137] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 19(5):530–535, 1997.
- [138] G. Sibley. Relative bundle adjustment. Technical Report 2307/09, University of Oxford, 2009.
- [139] G. Sibley, C. Mei, I. Reid, and P. Newman. Adaptive Relative Bundle Adjustment. In *Robotics Science and Systems*, August 2009.
- [140] C. Silpa-Anan and R. Hartley. Localisation using an image-map. In *Australian Conference on Robotics and Automation*, 2004.
- [141] C. Silpa-Anan and R. Hartley. Visual localization and loop-back detection with a high resolution omnidirectional camera. In *Workshop on Omnidirectional Vision*, 2005.
- [142] C. Silpa-Anan and R. Hartley. Optimised kd-trees for fast image descriptor matching. In *Computer Vision and Pattern Recognition*, 2008.
- [143] Robert Sim and Gregory Dudek. Mobile robot localization from learned landmarks. *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, 1998.
- [144] 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.
- [145] M. Smith, I. Baldwin, W. Churchill, R. Paul, and P. Newman. The new college vision and laser data set. *International Journal for Robotics Research (IJRR)*, 28(5):595–599, May 2009.
- [146] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. *Autonomous Robot Vehicles*, pages 167–193, 1990.
- [147] P. Spira and A. Pan. On finding and updating spanning trees and shortest paths. *SIAM Journal on Computing*, 4:375, 1975.
- [148] N. Srebro. Maximum likelihood bounded tree-width markov networks. In *Proceedings of the 17th Annual Conference on Uncertainty in Artificial Intelligence (UAI-01)*, pages 504–51, San Francisco, CA, 2001. Morgan Kaufmann.

- [149] M.J. Swain and D.H. Ballard. Color indexing. *International journal of computer vision*, 7(1):11–32, 1991.
- [150] M. Teyssier and D. Koller. Ordering-based search: A simple and effective algorithm for learning Bayesian networks. In *Proceedings of the Twenty-first Conference on Uncertainty in AI (UAI)*, pages 584–590, 2005.
- [151] A. Torralba, R. Fergus, and Y. Weiss. Small codes and large image databases for recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008*, pages 1–8, 2008.
- [152] A. Torralba, K.P. Murphy, W.T. Freeman, and M.A. Rubin. Context-based vision system for place and object recognition. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 273, Washington, DC, USA, 2003.
- [153] L. Torresani, M. Szummer, and A. Fitzgibbon. Learning query-dependent prefilters for scalable image retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.
- [154] M. Turk and A. Pentland. Face recognition using eigenfaces. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 591, 1991.
- [155] T. Tuytelaars and C. Schmid. Vector quantizing feature space with a regular lattice. In *IEEE 11th International Conference on Computer Vision, 2007. ICCV 2007*, pages 1–8, 2007.
- [156] 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.
- [157] C. Valgren and A. Lilienthal. SIFT, SURF and Seasons: Long-term outdoor localization using local features. In *Proc. of 3rd European Conference on Mobile Robots, Freiburg, Germany*, 2007.
- [158] A. Wald. *Sequential Analysis*. Dover Publications, New York, 1947.

- [159] M. Walter, R. Eustice, and J. Leonard. A provably consistent method for imposing sparsity in feature-based SLAM information filters. In *Proc. of the Int. Symposium of Robotics Research (ISRR)*, 2005.
- [160] Junqiu Wang, Hongbin Zha, and Roberto Cipolla. Vision-based global localization using a visual vocabulary. In *IEEE International Conference on Robotics and Automation*, 2005.
- [161] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *Advances in Neural Information Processing Systems*, volume 21, pages 1753–1760, 2009.
- [162] D.B. West et al. *Introduction to graph theory*. Prentice Hall, NJ, 2001.
- [163] 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 Systems*, 2008.
- [164] B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid, and J. Tardós. A comparison of loop closing techniques in monocular SLAM. *Robotics and Autonomous Systems*, 2009.
- [165] S. Winder and M. Brown. Learning local image descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR'07*, pages 1–8, 2007.
- [166] S. Winder, G. Hua, and M. Brown. Picking the best daisy. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR09)*, Miami, June 2009.
- [167] J. Wolf, W. Burgard, and H. Burkhardt. Robust vision-based localization by combining an image-retrieval system with Monte Carlo localization. *IEEE Transactions on Robotics*, 21(2):208–216, 2005.
- [168] Z. Wu, QF Ke, M. Isard, and J. Sun. Bundling features for large-scale partial-duplicate web image search. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.
- [169] 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.

- [170] Z. Zivkovic, B. Bakker, and B. Kröse. Hierarchical map building using visual landmarks and geometric constraints. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 7–12, 2005.