Loop Closure Detection on a Suburban Road Network using a Continuous Appearance-based Trajectory

Will Maddern, Michael Milford and Gordon Wyeth
School of Engineering Systems, Faculty of Built Environment and Engineering
Queensland University of Technology
{w.maddern, michael.milford, gordon.wyeth}@qut.edu.au

Abstract

This paper presents a novel technique for performing SLAM along a continuous trajectory of appearance. Derived from components of FastSLAM and FAB-MAP, the new system dubbed Continuous Appearance-based Trajectory SLAM (CAT-SLAM) augments appearance-based place recognition with particle-filter based 'pose filtering' within a probabilistic framework, without calculating global feature geometry or performing 3D map construction. For loop closure detection CAT-SLAM updates in constant time regardless of map size. We evaluate the effectiveness of CAT-SLAM on a 16km outdoor road network and determine its loop closure performance relative to FAB-MAP. CAT-SLAM recognizes 3 times the number of loop closures for the case where no false positives occur, demonstrating its potential use for robust loop closure detection in large environments.

1 Introduction

The future capabilities of mobile robots depend strongly on their abilities to navigate and interact in the real world. A key requirement for navigation is an internal representation of the environment that the robot inhabits. Autonomous robot navigation has been a major topic of robotics research for the past two decades, and is commonly referred to as the Simultaneous Localisation and Mapping (SLAM) problem [Thrun et al., 2008]. However, there is still very little use of SLAM systems outside research institutions, as a number of key problems with current SLAM approaches prevent their widespread use in unconstrained, real-world environments.

The majority of current state-of-the-art SLAM systems are based on a geometric interpretation of the SLAM problem. These geometric SLAM systems employ probabilistic algorithms such as Kalman filters [Dissanayake et al., 2001], Expectation Maximisation [Thrun et al., 2006] and Rao-Blackwellised particle filters [Montemerlo et al., 2002], and are simple to characterise and implement. However, their reliance on geometric consistency causes them to become computationally expensive and fragile when building large maps [Thrun et al., 2008].

To avoid computational and scaling limitations, a number of SLAM approaches forsake geometric accuracy for flexibility to form semi-metric or non-metric 'topological' approaches. Instead of attempting to combine all features from the environment in a single Euclidean space, non-geometric approaches typically form loosely-connected sub-maps [Bosse et al., 2003], reduced topological maps [Konolige et al., 2008], or simply record the trajectory and identify loop closure events [Angeli et al., 2009]. Although the maps generated by these algorithms are not sufficient to create accurate reconstructions of the environment, they provide the robot with the ability to localise and navigate successfully, which is all that is required for autonomous applications.

The most successful appearance-based SLAM algorithm to date is FAB-MAP [Cummins et al., 2008b]. FAB-MAP forsakes map building entirely and instead focuses on visual data association (so-called 'SLAM in appearance space'). A rigorous probabilistic approach to image matching based on a 'visual bag-of-words' model has allowed FAB-MAP to perform localisation on trajectories up to 1000km in length [Cummins et al., 2009].

While attempts have been made to incorporate FAB-MAP into a full SLAM solution, they have all involved additional geometric algorithms or additional sensors [Newman et al., 2009; Paul et al., 2010; Sibley et al., 2010]. These attempts are arguably not full SLAM systems since they do not incorporate a pose 'filter', instead relying only on strong data association.

In this paper we propose a novel interpretation of the SLAM problem, combining the spatial filtering characteristics of traditional geometric SLAM algorithms with the appearance-based place recognition of FAB-MAP. The novel algorithm, dubbed Continuous Appearance-based Trajectory SLAM (CAT-SLAM), conditions the joint distribution of the observation and motion model on a continuous trajectory of previously visited locations. The distribution is evaluated using a Rao-Blackwellised particle filter, which represents location hypotheses as particles constrained to the trajectory. We evaluate CAT-SLAM in a large outdoor environment, with results compared directly to those obtained with FAB-MAP.

2 Background

The probabilistic foundation of the Simultaneous Localisation and Mapping problem is defined as follows. Given a sequence of motion $\mathbf{U}_{0:k}$ and a sequence of observations $\mathbf{Z}_{0:k}$ of features $\mathbf{m}$, a history of states $\mathbf{X}_{0:k}$ can be derived. For the online SLAM problem, the state is derived by computing the following probability distribution:

$$P\left(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{x}_0\right) \tag{1}$$

The crucial observation first presented in [Smith et al., 1990] is that the state vector $\mathbf{x}$ and map $\mathbf{m}$ are not independent; errors in motion estimation are coupled with errors in observation, and as such the full joint posterior must be solved recursively. This can be performed with the use of two additional distributions; the motion model and the observation model, which describe the effect of motion and feature observation information on the joint posterior. The motion model describes the likelihood of a particular vehicle state given the current state and motion information:

$$P\left(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{u}_k\right) \tag{2}$$

The observation model describes the likelihood of a particular observation given the current state and map:

$$P\left(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{m}\right) \tag{3}$$

The joint posterior can now be updated in a standard predict-correct recursive Bayes form using these two models. This probabilistic definition primarily describes ‘pose filtering’; the process of combining uncertain observation and motion information to form an optimal estimate of the vehicle state. The definition does not constrain the solution to a particular type of map, nor does it provide any information on how to perform data association. The remainder of this section will describe 2 common approaches to the SLAM problem: geometric SLAM using particle filters, and appearance-based SLAM using FAB-MAP.

2.1 Particle Filter SLAM

The majority of Kalman filter and Rao-Blackwellised particle filter approaches to the SLAM problem use a geometric interpretation of the observation and motion model, shown in Figure 1. A series of metric measurements $\mathbf{z}_i$ are taken from locations $\mathbf{x}_i$ to features $\mathbf{m}_i$, typically in the form of range, bearing or a combination. The location of the features $\mathbf{m}_i$ with respect to the previously visited discrete locations $\mathbf{x}_i$ can then be determined in continuous geometric space. Additionally, the expected observation for locations between previously visited states (labelled $\mathbf{x}_k$) can be determined using relative geometry, as can the expected observation for any arbitrary location in space (labelled $\mathbf{x}_j$).

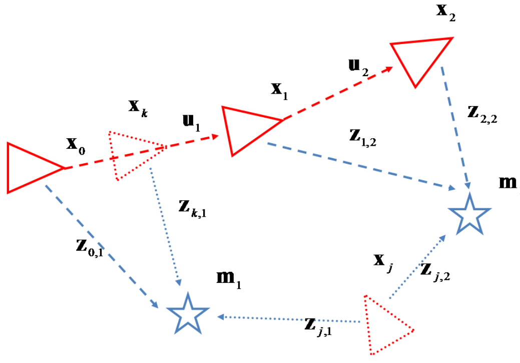

Figure 1 – Geometric SLAM interpretation. A continuous observation and motion model is defined by successive observations of feature geometry.

A popular SLAM algorithm that makes use of the geometric solution to the SLAM problem is FastSLAM, developed in [Montemerlo et al., 2002], which uses a Rao-Blackwellised particle filter and various schemes for particle resampling. By storing many different location and map hypotheses as individual particles and assigning weights to those particles based on how well they match observations, FastSLAM avoids both the linearization and computational complexity issues of EKF SLAM. The chief innovation in Rao-Blackwellisation is decoupling the process noise from the observation noise. By assuming the map stored by each particle is correct, observations become conditionally independent. The distribution is partitioned as follows:

$$\begin{aligned} &P\left(\mathbf{x}_{0:k}, \mathbf{m} \mid \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{x}_0\right) \\ &= P\left(\mathbf{m} \mid \mathbf{x}_{0:k}, \mathbf{Z}_{0:k}\right) P\left(\mathbf{x}_{0:k} \mid \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{x}_0\right) \end{aligned} \tag{4}$$

The joint state is represented by N particles, each with pose history $\mathbf{X}$, weight w and distribution as follows:

$$\left\{ w_k^{(i)}, \mathbf{X}_{0:k}^{(i)}, P\left(\mathbf{m} \mid \mathbf{X}_{0:k}^{(i)}, \mathbf{Z}_{0:k}\right) \right\}_i^N \tag{5}$$

The motion-update of FastSLAM is performed by directly sampling from the distribution for each particle:

$$\mathbf{x}_k^{(i)} \sim P\left(\mathbf{x}_k \mid \mathbf{x}_{k-1}^{(i)}, \mathbf{u}_k\right) \tag{6}$$

Each particle is then assigned a weight based on the importance function:

$$w_k^{(i)} = w_{k-1}^{(i)} \frac{P\left(\mathbf{z}_k \mid \mathbf{X}_{0:k}^{(i)}, \mathbf{Z}_{0:k-1}\right) P\left(\mathbf{x}_k^{(i)} \mid \mathbf{x}_{k-1}^{(i)}, \mathbf{u}_k\right)}{\pi\left(\mathbf{x}_k^{(i)} \mid \mathbf{X}_{0:k-1}^{(i)}, \mathbf{Z}_{0:k}, \mathbf{u}_k\right)} \tag{7}$$

All weights are normalised to sum to 1. The particles are then resampled with replacement, where the probability of selection is proportional to the weight w. All remaining particles are then updated using the EKF (or a variant such as the UKF). While this is effective in allowing FastSLAM to store multiple hypotheses and switch between them as required, it can suffer from “particle deprivation” if there are no particles near the correct hypothesis [Van der Merwe et al., 2001].

Many extensions have been made to the FastSLAM algorithm: FastSLAM 2.0 [Montemerlo et al., 2003], which includes the current observation in the proposal distribution for locally optimal sampling; GridSLAM [Hähnel et al., 2003], which extends the environment representation to an occupancy grid reducing the complications of data association in feature-based representations; and Distributed Particle SLAM (DP-SLAM) [Eliazar et al., 2003], which further reduces the computational complexity of FastSLAM by storing the

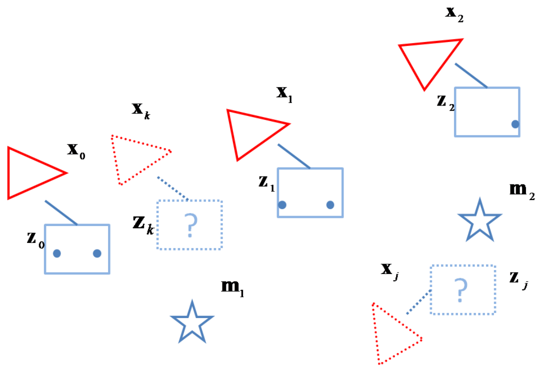

Figure 2 – Appearance-based SLAM interpretation. Expected observations are only available at discrete locations where an observation was previously made. Motion information is not used, allowing loop closures of unlimited size.

particles in an ancestry tree and recording map divergences rather than storing an entire map for each particle.

2.2 Appearance-based Place Recognition

Another major approach to SLAM that has gained popularity in recent years is appearance-based or appearance-only SLAM. It is primarily used for detecting loop closures in large unknown environments, which it performs by determining whether the current location matches any previously visited locations or is sufficiently different as to be classified as a new location.

Figure 2 illustrates the appearance-based approach to the SLAM observation and motion model. Each state $\mathbf{x}_i$ has an associated observation $\mathbf{z}_i$ which stores which features mi are visible from that location. The map is represented by the history of states $\mathbf{X}_{0:k}$. However, motion information is typically discarded, since there is no method of generating the expected appearance neither between locations (labelled $\mathbf{x}_k$) nor at arbitrary locations (labelled $\mathbf{x}_j$). Appearance-based SLAM systems can therefore close loops of any size, regardless of accumulated odometry error, but rely entirely on the data association between the current observation and a previous observation.

The current state-of-the-art appearance-based SLAM system is FAB-MAP [Cummins et al., 2008b], which uses a Chow-Liu dependency tree and recursive Bayes estimation within a rigid probabilistic framework to provide robust loop closure detection.

Each image is converted into the visual bag-of-words representation described in [Sivic et al., 2003]. It is therefore necessary to create a database of common features from a set of training data in a similar environment to the test environment prior to performing localisation [Cummins et al., 2007]. Every feature extracted from the image is converted to the closest visual word, reducing each image to a binary vector of which words are present in the image.

$$Z_k = \{z_1, \ldots, z_{|v|}\} \qquad (8)$$

Each unique location L_k is represented by the probability that the object e_i (that creates observation z_i) is present in the scene.

$$\{P(e_i = 1 \mid L_k), \ldots, P(e_{|v|} = 1 \mid L_k)\} \qquad (9)$$

The probability of a new image coming from the same location as a previous image is estimated using recursive Bayes:

$$P(L_i \mid \mathcal{Z}^k) = \frac{P(Z_k \mid L_i, \mathcal{Z}^{k-1}) P(L_i \mid \mathcal{Z}^{k-1})}{P(Z_k \mid \mathcal{Z}^{k-1})} \qquad (10)$$

where $\mathcal{Z}^k$ is a collection of previous observations up to time k. $P(Z_k \mid L_i, Z^{k-1})$ is assumed to be independent from all past observations and is calculated using a Chow Liu approximation [Chow et al., 1968]. The Chow Liu tree is constructed once as an offline process based on training data. Observation likelihoods are determined using the Chow Liu tree as follows:

$$P(Z_k \mid L_i) \approx P(z_r \mid L_i) \prod_{q=1}^{|v|} P(z_q \mid z_{p_q}, L_i) \qquad (11)$$

where r is the root node of the Chow Liu tree and p_q is the parent of node q. The prior probability of matching a location $P(L_i \mid \mathcal{Z}^{k-1})$ is estimated using a naïve motion model, where the probability of a new place $P(L_{new} \mid Z^{k-1})$ is set to a constant if the current hypothesised location is within 1 frame of the matched location. In practice this has only a slight effect on the final result [Cummins et al., 2008b].

The denominator of equation 10 incorporates the probability of matching to a new location in addition to localisation to a previously visited place. To estimate if a new observation comes from a previously unvisited location the model needs to consider all locations, not just visited locations. This can be split into mapped and unmapped locations:

$$\begin{aligned} P(Z_k \mid \mathcal{Z}^{k-1}) = & \sum_{m \in M} P(Z_k \mid L_m) P(L_m \mid \mathcal{Z}^{k-1}) \\ + & \sum_{n \in M} P(Z_k \mid L_n) P(L_n \mid \mathcal{Z}^{k-1}) \end{aligned} \qquad (12)$$

where M is the set of mapped locations. Since the second term cannot be evaluated directly (as it would require information on all unknown locations), an estimation must be used. A random selection of scenes from training data is used to evaluate the unmapped location according to:

$$\begin{aligned} & \sum_{n \in M} P(Z_k \mid L_n) P(L_n \mid \mathcal{Z}^{k-1}) \\ & \approx P(L_{new} \mid \mathcal{Z}^{k-1}) \sum_{u=1}^{n_s} \frac{P(Z_k \mid L_u)}{n_s} \end{aligned} \qquad (13)$$

where L_u is a sampled location and n_s is the total number of samples. The sampling technique generally provides superior results to the mean field approximation [Cummins et al., 2008b].

A number of enhancements have been made to the original FAB-MAP algorithm, to both reduce the computational cost of storing large environments and to increase the matching speed of the system. The implementation in [Cummins et al., 2008a] presented a probabilistic bail-out condition based on the Bennett Inequality [Boucheron et al., 2004], to rank features based on their information content and to discard unlikely matches without performing the full recursive Bayes calculation. To further reduce the amount of computation required, an inverted index lookup scheme was implemented in FAB-MAP 2.0 [Cummins et al., 2009], which allows fully sparse evaluation.. An additional RANSAC stage was added in FAB-MAP 2.0 to provide a geometric post-verification of image matches.

A number of attempts to incorporate FAB-MAP into a full mapping system have been made, where it has been used as a first stage to detect loop closure. However, these attempts then rely on geometric matching techniques using either laser scanners [Paul et al., 2010] or stereo cameras [Newman et al., 2009], which do not incorporate odometric information in the manner of a pose filter, and as such still rely entirely on the strength of data association between two discrete locations.

3 Trajectory-based Pose Filtering

The proposed SLAM system outlined in this section is derived from a ‘trajectory-based’ interpretation of the SLAM problem. This interpretation lies between the two major SLAM paradigms presented in the previous section; it combines aspects of the geometric motion model of FastSLAM with the appearance-based observation model of FAB-MAP.

A diagram of the trajectory-based interpretation is presented in Figure 3. As with FastSLAM, states $\mathbf{x}_i$ are linked by odometry information $\mathbf{u}_i$; however, observations $\mathbf{z}_i$ are formed by appearance representations rather than metric distances. The observation model is formed by a continuous trajectory-based appearance model, which calculates the expected appearance along the trajectory between two nodes. This model allows the calculation of the expected observation $\mathbf{z}_k$ from location $\mathbf{x}_k$ on the trajectory between two previously visited locations. However, unlike the geometric observation model, it does not allow the calculation of the expected observation $\mathbf{z}_j$ at an arbitrary location $\mathbf{x}_j$. This limits the system to localising only to exact trajectories it has previously traversed; however, the utility of other appearance-based SLAM methods indicate that this capability is not required for all applications.

The observation model can take any form, but is only required to determine the existence or non-existence of visible features along the continuous trajectory between two sequential observations. As such, methods that do not require feature correspondence or geometry are preferred.

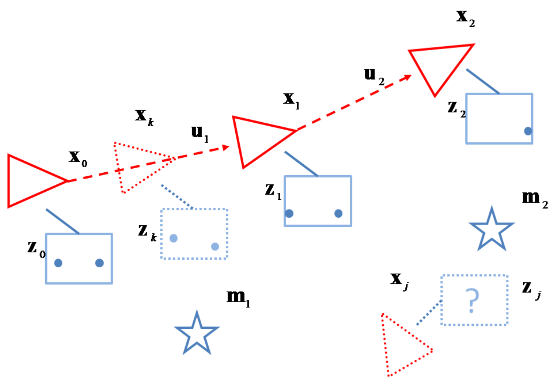

Figure 3 – Trajectory-based SLAM interpretation. A continuous trajectory-based observation model allows the expected appearance to be calculated at any point along a previously visited trajectory. Motion information permits the use of pose filtering without restricting loop closure size.

The history of poses is represented by a continuous trajectory T, which intersects all previous discrete poses $\mathbf{X}_{0:k}$:

$$\mathbf{X}_{0:k} \in T \Rightarrow \mathbf{x}(t) \in T, \quad 0 \leq t \leq k \tag{14}$$

The full history of poses is recovered using the continuous pose $\mathbf{x}(t)$ with continuous index t, where $\mathbf{x}(k) = \mathbf{x}_k$. For localisation in three degrees of freedom the discrete and continuous pose are represented as follows:

$$\mathbf{x}_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix}, \quad \mathbf{x}(t) = \begin{bmatrix} x(t) \\ y(t) \\ \theta(t) \end{bmatrix} \tag{15}$$

The particular form of the trajectory T is defined by the continuous motion model of the vehicle; the simplest case of a linearly interpolated motion model is illustrated as follows:

$$\mathbf{x}(t) = (\lceil t \rceil - t)\mathbf{X}_{\lfloor t \rfloor} + (t - \lfloor t \rfloor)\mathbf{X}_{\lceil t \rceil} \tag{16}$$

As in topological SLAM solutions, the map is formed by the continuous history of poses as follows:

$$\mathbf{m} = \mathbf{X}_{0:k} \Rightarrow \mathbf{m} = \mathbf{x}(0 \leq t \leq k) \tag{17}$$

The map update is performed by correcting the history of poses $\mathbf{X}_{0:k}$ when a loop closure is detected. However, for this implementation the purpose is to determine the characteristics of loop closure, and as such global map correction is not required. The SLAM distribution when conditioned on the continuous trajectory T is modified from equation 1 as follows:

$$P(\mathbf{x}_k \in T \mid \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{x}_0) \tag{18}$$

For this implementation, the distribution above is evaluated using a Rao-Blackwellised particle filter. The distribution is approximated using N particles, each with weight w, position on the trajectory $\mathbf{x}_k$, and continuous trajectory index t:

$$\left\{ w_k^{(i)}, \mathbf{x}_k^{(i)}, t^{(i)} \right\}_i^N \tag{19}$$

The following sections detail the components of the Rao-Blackwellised particle filter required to solve the joint distribution along the continuous trajectory.

3.1 Trajectory-based Sampling

The proposal distribution for the trajectory-based particle filter is given by the vehicle motion conditioned on the trajectory T:

$$\mathbf{x}_k^{(i)} \sim P(\mathbf{x}_k \in T \mid \mathbf{x}_{k-1}^{(i)}, \mathbf{u}_k) \tag{20}$$

This method allows a nonlinear vehicle motion model to be used (as with EKF- or FastSLAM), but ensures all particles remain constrained to the trajectory of previously visited locations. The particle update is performed by first generating a proposed pose $\hat{\mathbf{x}}_k$ from the nonlinear vehicle model f with additive Gaussian noise $\mathbf{w}_k$:

$$\hat{\mathbf{x}}_k^{(i)} = f(\mathbf{x}_{k-1}^{(i)}, \mathbf{u}_k) + \mathbf{w}_k \tag{21}$$

For a vehicle moving in three degrees of freedom the motion model is as follows:

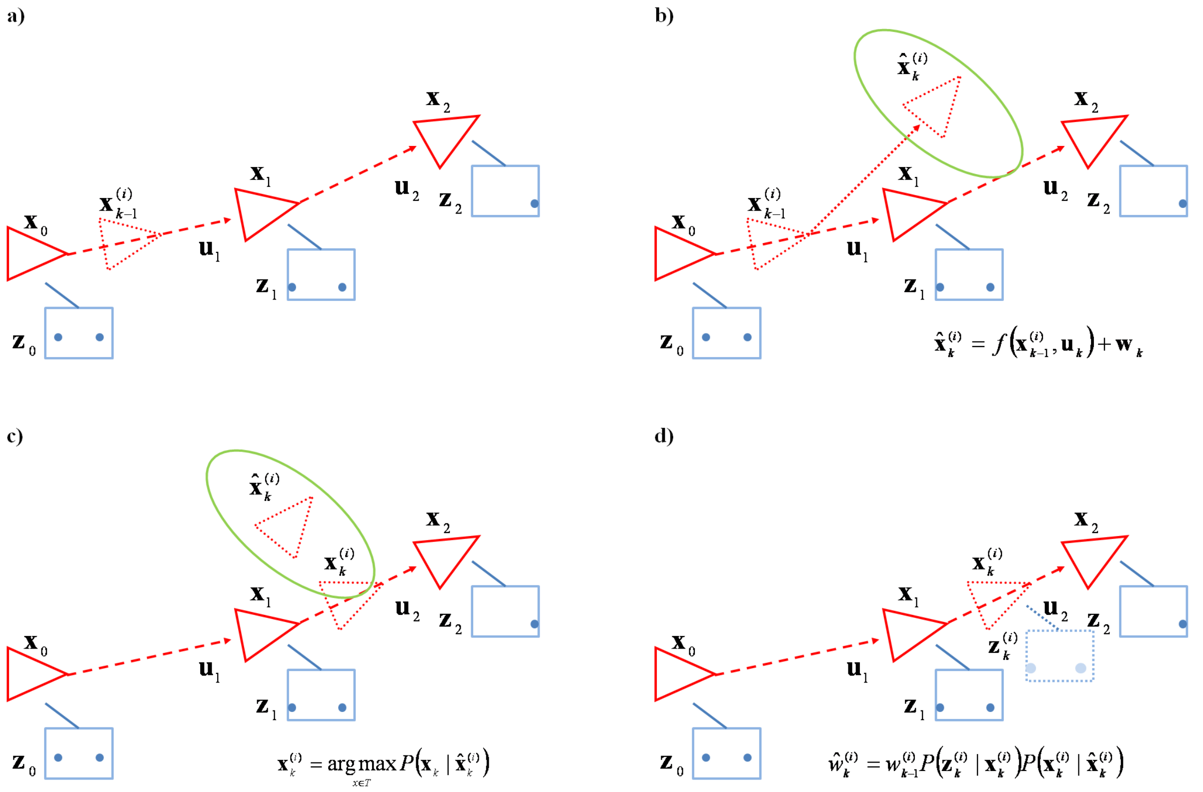

Figure 4 – Update process of CAT-SLAM particles. a) Particles $\mathbf{x}_{k-1}^{(i)}$ are constrained to the trajectory between previously visited locations $\mathbf{x}_{0:k}$. b) Proposed particle locations $\hat{\mathbf{x}}_k^{(i)}$ are sampled from the motion model with control input $\mathbf{u}_k$. c) The updated position on the trajectory $\mathbf{x}_k^{(i)}$ is found at maximum likelihood location of distribution $P\left(\mathbf{x}_k \mid \hat{\mathbf{x}}_k^{(i)}\right)$. d) The particle weight is updated using the motion likelihood and observation likelihood $P\left(\mathbf{z}_k^{(i)} \mid \mathbf{x}_k^{(i)}\right)$, where $\mathbf{z}_k$ is generated using a continuous appearance model.

$$f\left(\mathbf{x}_k, \mathbf{u}_k\right) = \begin{bmatrix} x_k + \Delta x_k \cos\left(\theta_k + \Delta\theta_k\right) - \Delta y_k \sin\left(\theta_k + \Delta\theta_k\right) \\ y_k + \Delta x_k \sin\left(\theta_k + \Delta\theta_k\right) + \Delta y_k \cos\left(\theta_k + \Delta\theta_k\right) \\ \theta_k + \Delta\theta_k \end{bmatrix} \tag{22}$$

where $\mathbf{u}_k = \left[\Delta x_k \quad \Delta y_k \quad \Delta\theta_k\right]^T$. The proposed covariance is generated by linearising the motion model at the proposed location with noise covariance $\mathbf{Q}_k$:

$$\mathbf{\Sigma}_k^{(i)} = J_k^{(i)} \mathbf{Q}_k J_k^{(i)\mathrm{T}}, \quad J_k^{(i)} = \frac{\partial f\left(\mathbf{x}_{k-1}^{(i)}, \mathbf{u}_k\right)}{\partial \mathbf{u}} \tag{23}$$

From this, a distribution over all possible poses can be represented using the standard multivariate Gaussian:

$$P\left(\mathbf{x} \mid \hat{\mathbf{x}}_k\right) = \frac{1}{2\pi\sqrt{\left|\mathbf{\Sigma}_k\right|}} \exp\left[\left(\mathbf{x} - \hat{\mathbf{x}}_k\right) \mathbf{\Sigma}_k^{-1} \left(\mathbf{x} - \hat{\mathbf{x}}_k\right)^{\mathrm{T}}\right] \tag{24}$$

The location of the particle on the trajectory is found by searching the trajectory for the continuous index t for which the above distribution is maximized:

$$t^{(i)} = \arg\max_{t \le k} P\left(\mathbf{x}(t) \mid \hat{\mathbf{x}}_k^{(i)}\right) \tag{25}$$

From this index the pose of the particle is set to the most likely pose on the trajectory:

$$\mathbf{x}_k^{(i)} = \mathbf{x}\left(t^{(i)}\right) \tag{26}$$

The maximum motion likelihood $P\left(\mathbf{x} \mid \hat{\mathbf{x}}_k\right)$ is stored for later use in particle importance weighting.

3.2 Continuous Appearance Representation

In order to generate the observation model of equation 3 for a continuous trajectory based system, a continuous representation of appearance is required. Conventional geometric SLAM systems such as MonoSLAM [Davison et al., 2007] perform this using 3D feature locations in a fixed co-ordinate frame; however as stated above the purpose of this algorithm is not to locate features in 3D space. The location representation for each particle is derived from that presented in equation 9, but extended to represent appearance between discrete observations as follows:

$$\left\{P\left(e_i = 1 \mid \mathbf{x}\left(t^{(i)}\right)\right), \ldots, P\left(e_{|v|} = 1 \mid \mathbf{x}\left(t^{(i)}\right)\right)\right\} \tag{27}$$

The method of generating these interpolated appearance representations is dependent on both the continuous vehicle motion model and the camera model. However, for the simple linear case of equation 15, the continuous representation of appearance can be generated similarly as follows, by interpolating between two successive discrete observations:

$$\begin{aligned} P\left(e_i = 1 \mid \mathbf{x}(t)\right) = &\left(\lceil t \rceil - t\right) P\left(e_i = 1 \mid \mathbf{Z}_{\lfloor t \rfloor}\right) \\ &+ \left(t - \lfloor t \rfloor\right) P\left(e_i = 1 \mid \mathbf{Z}_{\lceil t \rceil}\right) \end{aligned} \tag{28}$$

As with FAB-MAP, the set of visual words v that form the observation and appearance representation must be derived from training data in a similar environment to the test environment.

3.3 Importance Weighting

The importance weighting of the particles is drawn from the numerators of equation 7 and 10; it combines the observation likelihood of FAB-MAP using the continuous representation of appearance with the motion prior of FastSLAM conditioned on the trajectory. By only evaluating the motion and observation model once per particle, updating the weights can be performed in constant time proportional to the number of particles regardless of the number of previously visited locations. The proposed weighting of each particle is as follows:

$$\hat{w}_k^{(i)} = w_{k-1}^{(i)} P\left(\mathbf{z}_k \mid \mathbf{x}_k^{(i)}\right) P\left(\mathbf{x}_k^{(i)} \in T \mid \mathbf{x}_{k-1}^{(i)}, \mathbf{u}_k\right) \qquad (29)$$

The observation likelihood makes use of the Chow Liu distribution as in equation 11 at location *t* on the trajectory:

$$P\left(\mathbf{z}_k \mid \mathbf{x}_k^{(i)}\right) = P\left(z_r \mid \mathbf{x}\left(t^{(i)}\right)\right) \prod_{q=1}^{|v|} P\left(z_q \mid z_{p_q}, \mathbf{x}\left(t^{(i)}\right)\right) \qquad (30)$$

The leftmost part of equation 28 is calculated as follows:

$$\begin{aligned} & P\left(z_q \mid z_{p_q}, \mathbf{x}\left(t^{(i)}\right)\right) \\ &= \sum_{s \in \{0,1\}} P\left(z_q \mid e_q = s, z_{p_q}\right) P\left(e_i = s \mid \mathbf{x}\left(t^{(i)}\right)\right) \end{aligned} \qquad (31)$$

where $P\left(z_q \mid e_q = s, z_{p_q}\right)$ is the detector probability and $P\left(e_i = s \mid \mathbf{x}\left(t^{(i)}\right)\right)$ is the continuous appearance representation defined in equation 26. The motion prior is the maximum likelihood point of the motion distribution along the trajectory as found in equation 23:

$$P\left(\mathbf{x}_k^{(i)} \in T \mid \mathbf{x}_{k-1}^{(i)}, \mathbf{u}_k\right) = P\left(\mathbf{x}_k^{(i)} \mid \hat{\mathbf{x}}_k^{(i)}\right) \qquad (32)$$

To represent the likelihood of a location not on the trajectory (similar to the denominator term in equation 12), an additional particle representing an ‘unknown’ pose is required. The weight of this particle is calculated as follows:

$$\hat{w}_k^u = \tfrac{1}{N} P\left(\mathbf{z}_k \mid \mathbf{x}_k^u\right) P\left(\mathbf{x}_k^u \mid \mathbf{u}_k\right) \qquad (33)$$

Note that this particle does not update using its previous weight but is re-normalised at each step; this represents the uniform likelihood of departing from a previously visited section of the trajectory at any point in time. The two distributions can be approximated using information from training data as follows:

$$P\left(\mathbf{z}_k \mid \mathbf{x}_k^u\right) P\left(\mathbf{x}_k^u \mid \mathbf{u}_k\right) \approx P\left(\mathbf{z}_k \mid \mathbf{z}_{\text{avg}}\right) P\left(\mathbf{u}_{\text{avg}} \mid \mathbf{u}_k\right) \qquad (34)$$

Where $\mathbf{z}_{\text{avg}}$ represents an ‘average’ observation and $\mathbf{u}_{\text{avg}}$ an ‘average’ control input. These can be found by simply averaging all observations and controls in the training data set, or by using the random sampling method in equation 13. Without this ‘unknown’ pose particle, the particle distribution represents pure localization, since the probability of a pose not on the trajectory is assumed to be zero.

3.4 Particle Resampling

The proposed weight of each particle is normalised, such that the sum of all weighs of particles on the trajectory plus the ‘unknown’ particle weight is equal to 1.

$$w_k^{(i)} = \frac{\hat{w}_k^{(i)}}{\sum_j^N \hat{w}_k^{(j)} + \hat{w}_k^u} \qquad (35)$$

The particles are resampled when the effective sample size (ESS) falls below a predefined threshold [Liu et al., 2001]. The ESS is computed as follows:

$$ESS = \frac{N}{1 + \frac{1}{N} \sum_j^{N,u} \left[N w_k^{(j)} - 1\right]^2} \qquad (36)$$

Particles are selected with probability proportional to their weight w_k using the Select with Replacement method [Liu et al., 2001]. Any particles selected to replace the ‘unknown’ particle are sampled to a uniform random location on the trajectory as follows:

$$t^{(i)} \sim U(0,k), \mathbf{x}_k^{(i)} = \mathbf{x}\left(t^{(i)}\right) \qquad (37)$$

This serves to counteract the effects of particle deprivation, since the proportion of particles sampled to diverse locations on the trajectory increases (thereby increasing the probability of detecting loop closure) as the ‘unknown place’ likelihood increases.

3.5 Trajectory Distribution Calculation

To determine the most likely location hypothesis from the distribution of particles a spatially selective method is used, equivalent to integrating the probability distribution over a short distance along the trajectory. The value of the distribution at particle location $\mathbf{x}_k$ is as follows:

$$P\left(\mathbf{x}_k^{(i)}\right) = \frac{\sum_j^N h(i,j)}{1 + w_k^u} \qquad (38)$$

The spatially selective function $h(i, j)$ is defined as follows:

$$h(i,j) = \begin{cases} w_k^{(j)} & \left|\mathbf{x}_k^{(j)} - \mathbf{x}_k^{(i)}\right| \le d \\ 0 & \text{otherwise} \end{cases} \qquad (39)$$

The distribution will therefore only reach a probability of 1 at any location if all particles are within distance d of the hypothesized location (causing the numerator to sum to 1), and the ‘unknown’ location weight is equal to 0.

4 Experimental Procedure

The following section details the steps taken to evaluate the proposed CAT-SLAM algorithm. Since the primary focus of this algorithm is improving loop closure performance and not map construction, the experiment will be focused on comparing it to FAB-MAP in a large outdoor environment.

4.1 Experimental Setup

This experiment uses a dataset previously gathered for the full-day mapping experiment presented in [Glover et al., 2010]. The dataset consists of video captured at 15 frames per second from a forward-facing Logitech QuickCam Pro 9000 webcam mounted on the roof of a car, as well as GPS data gathered at 1Hz for ground truth. The images gathered by the camera have a resolution of 640x480 pixels, representing a field of view of 62 degrees horizontal by 46 degrees vertical. The GPS lock remained consistent throughout the entire route.

The route taken by the car is a 16km tour of a surburban road network, pictured in Figure 5, with multiple repeated loops and wide variation in the types of roads traversed; from wide 4-lane main roads to single lane roads bordered by dense foliage. The dataset was gathered at midday to reduce the likelihood of image saturation due to direct sunlight. No modification of the environment or interruption of normal traffic conditions was performed.

4.2 Algorithm Details

The codebook and Chow Liu tree used for both FAB-MAP and CAT-SLAM are derived from the experiments presented in [Milford et al., 2008a]. SURF features were extracted from 7000 non-overlapping images sampled from a larger dataset of the same suburb, resulting in a codebook containing 5730 visual words. The average observation was derived from this same dataset using the mean field approximation.

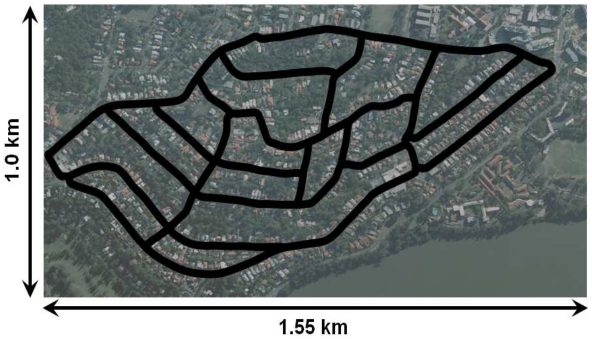

Figure 5 – Test environment consisting of a 16km road network covering approximately 1.5 square kilometres.

Table 1 – Summary of algorithm parameters.

FAB-MAP		
Detector model	$P(z{=}1 \mid e{=}0)$	0
	$P(z{=}0 \mid e{=}1)$	0.61
New place likelihood	$P(L_{\text{new}} \mid Z^{k\text{-}1})$	0.9
CAT-SLAM		
Detector model	$P(z{=}1 \mid e{=}0)$	0
	$P(z{=}0 \mid e{=}1)$	0.61
Motion uncertainty	σ_y	0.05 metres
	σ_θ	0.05 radians
Number of Particles	N	2000
Minimum ESS threshold	ESS	0.25
Local distribution size	d	2.5 metres

Odometry information was generated from visual information using the method presented in [Milford et al., 2008b]. While this semi-metric method is not as precise as feature-based approaches, it does not require the calculation of feature correspondence or geometry and produces repeatable results in successive traversals of the same location.

The list of constants used in both algorithms is presented in Table 1. For evaluation, the maximum distance between the expected and actual GPS location for a true positive loop closure was set to 20m, to reflect the large scale of the dataset.

5 Results

This section describes the mapping results of both FAB-MAP and CAT-SLAM on the 16km suburban road network. The primary performance metric is the precision-recall curve, where precision is defined as the number of correct matches divided by the total number of matches, and recall as the number of correct matches divided by the total number of expected matches:

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}}, \ \text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} \qquad (40)$$

where TP are true positives, FP false positives and FN false negatives. Expected correct matches are defined as previously visited GPS locations within 20m distance and 10 degrees heading to the current location, to avoid unreasonable loop closure expectations (such as when crossing intersections from different approaches).

To use either of these systems to detect loop closure for a semi-metric or metric SLAM system a precision of 100% is required, since false positive matches can cause catastrophic failure during mapping for geometric systems [Thrun et al., 2008]. In this respect, the desired outcome is a high recall rate at 100% precision. However, analysis of the false positives reported by both systems is important to determine the likely failure modes in other environments.

5.1 Precision-Recall

Figure 6 shows the precision-recall curve for both FAB-MAP and CAT-SLAM on the full road network.

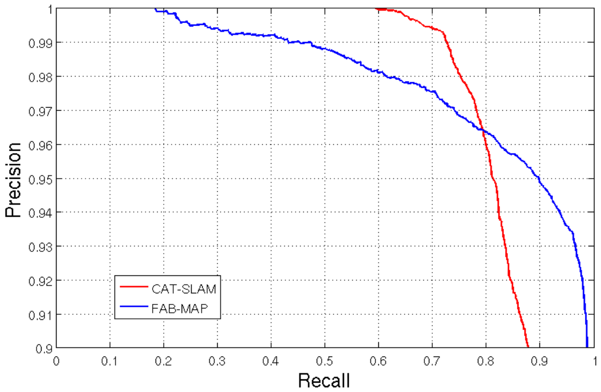

Figure 6 – Precision-recall curve for CAT-SLAM and FAB-MAP for the 16km trajectory. At 100% precision, CAT-SLAM recalls 59% of the correct matches, where FAB-MAP only recalls 19%.

Note the scale on the y-axis; both algorithms report over 90% precision over more than 80% of the recall range. However, FAB-MAP 1.0 achieves only 19% recall at 100% precision, where CAT-SLAM recalls over 59% of locations successfully. The poor performance of FAB-MAP in this environment compared to previous FAB-MAP experiments can be attributed to the use of a single forward-facing camera, as opposed to multiple wide-angle [Cummins et al., 2008b] or omnidirectional [Newman et al., 2009] cameras. The geometric post-verification in [Cummins et al., 2009] was not used.

Below 96% precision, FAB-MAP provides superior recall rates to CAT-SLAM. Since FAB-MAP compares the current location appearance to all previous locations where CAT-SLAM only compares to previous locations where particles exist, it is possible that if the particle diversity is sufficiently low occasional loop closures will not be detected. This does not appear to have a significant effect on the results, as even at 90% precision CAT-SLAM provides almost 90% recall.

5.2 Loop Closure Performance

Figure 7 shows the loop closures projected on to the GPS positions recorded on the route for four separate matching cases. Figures 7a) and 7b) show the loop closures detected by FAB-MAP and CAT-SLAM for the threshold that provides maximum recall at 100% precision. The larger number of green true positive loop closures illustrate that CAT-SLAM correctly relocalises more often than FAB-MAP. Additionally, the true positive loop closures are approximately evenly distributed across the area and not concentrated in particular locations of unique visual appearance.

The relative distributions of true positive and false negative loop closures are of note: FAB-MAP tends to switch between true positive and false negative matches within the space of a few metres, where CAT-SLAM typically has unbroken sequences of 20m or more. This reflects the sequential matching nature of CAT-SLAM; it requires a number of correct visual and odometric matches before a sufficiently dominant hypothesis is formed. However, once such a hypothesis is dominant, it is maintained until particles not within the local distribution distance *d* are sufficiently supported by novel visual and odometric information to suggest an alternate hypothesis.

Figures 7c) and 7d) show the loop closures detected at 95% precision. While both algorithms present an equal fraction of false positives, FAB-MAP appears to distribute them more evenly across the dataset. The trajectory-based matching characteristics of CAT-SLAM cause it to maintain even false positives (provided there is sufficient false visual and odometric information). Interestingly, all sections reported as false positives by CAT-SLAM occur on straight sections of road; in these cases the motion model provides sufficient weighting to remove loop closure candidates that are geometrically dissimilar along the trajectory.

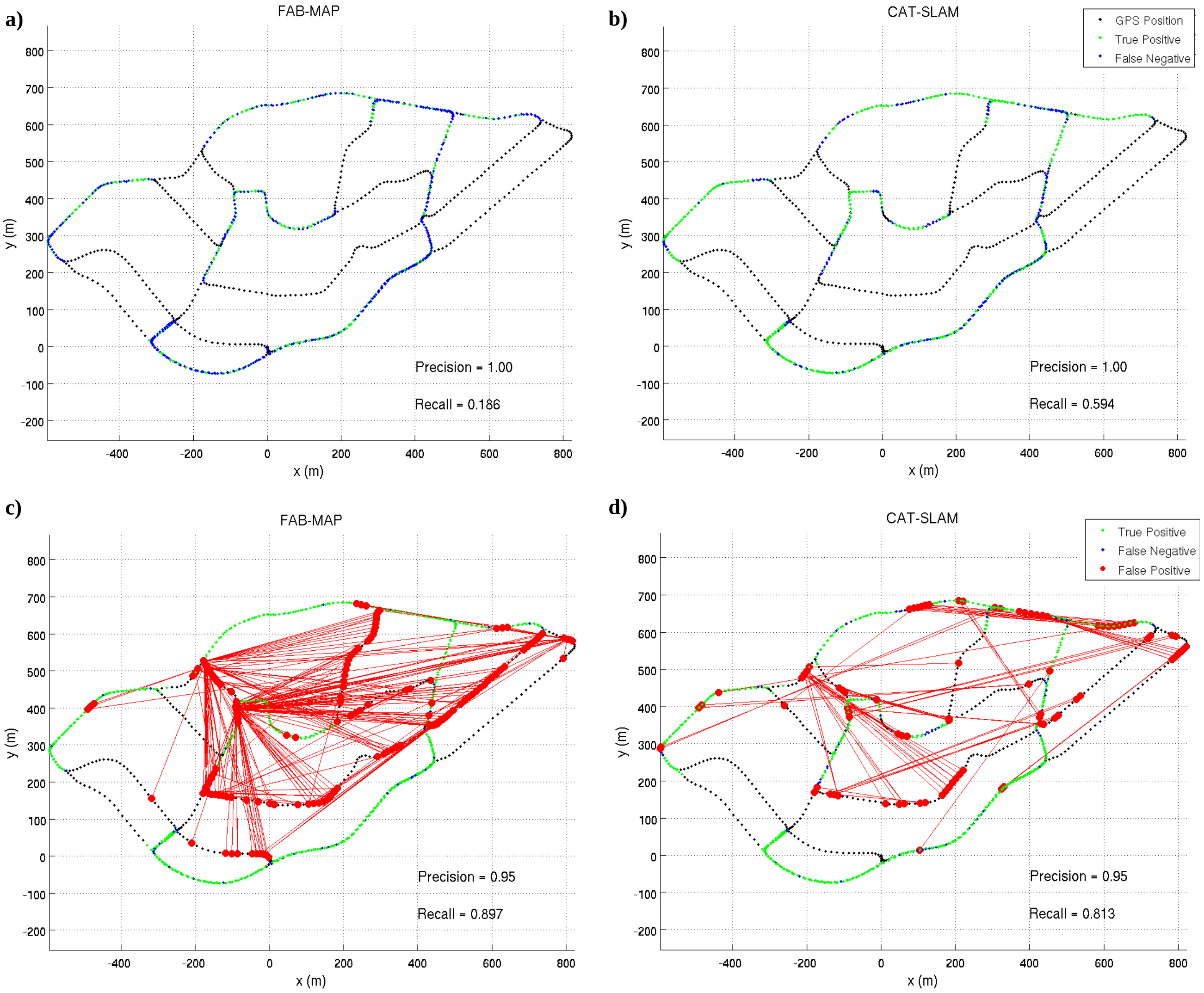

Figure 7 – Loop closures projected on GPS ground truth. a) and b) show loop closures at 100% precision, c) and d) show loop closures at 95% precision. Lighter green points indicate true positives, darker blue points indicate false negatives, red points with loop closure lines show false positives.

5.3 Computational Complexity

Currently CAT-SLAM requires approximately 1ms per particle update on a single core 3GHz Core 2 processor, comparable to early FAB-MAP implementations [Cummins et al., 2007]. The majority of this time is spent evaluating the observation likelihood. CAT-SLAM is unsuitable for real-time application in its current form; however, since each particle is independent, the algorithm is highly parallelisable. However, every past observation must be stored, so the appearance-based map grows in memory requirements in proportion to the number of observations.

6 Discussion

The results of the mapping experiment demonstrated that the combination of both appearance and motion information in CAT-SLAM provides a clear advantage over appearance-based SLAM systems that rely on visual data alone for applications that require 100% precision loop closure. The improvement over FAB-MAP is twofold; first, the addition of a pose filter allows spurious false positives to be rejected, and it allows a location hypothesis to be maintained with only partial visual matches.

Since CAT-SLAM is built upon the same underlying appearance-based matching system as FAB-MAP, its performance at identifying an initial loop closure is approximately equal. Due to the trajectory following properties of the particles, CAT-SLAM can maintain a hypothesis across a number of frames when supporting visual information above the hypothesis threshold is not available for all frames, as is the case with FAB-MAP. This greatly increases the recall rates as entire sections of trajectories can be matched, rather than simply individual frames.

However, the requirement for a sequence of familiar visual and odometric information reduces the speed at which CAT-SLAM is able to generate a new location hypothesis. While FAB-MAP can localize using only a single frame, CAT-SLAM requires a number of particle update (and possibly resample) stages; revisting short sections of a path (such as crossing an intersection from a different approach) may not be detected by CAT-SLAM.

The computational advantages of a fixed number of particles representing a distribution could allow CAT-SLAM to scale to much larger environments than other appearance-based SLAM systems, provided sufficient particle diversity is maintained.

6.1 Future Work

There are a number of improvements that can be made in many aspects of the current CAT-SLAM implementation to improve its performance.

Currently the linear approximations in equations 16 and 28 cater to the vehicle model used. However, for holonomic vehicles which do not necessarily revisit a previously traversed trajectory with an identical orientation, these approximations will not suffice. Explicit decoupling of orientation with trajectory will be required to support holonomic vehicles and similar platforms.

The linear interpolation of appearance in equation 28 does not capture the true nature of changing appearance with respect to motion. A more sophisticated method that incorporates feature-based optical flow without evaluating 3D feature geometry is currently in development.

The principles behind the Bennet bound and inverted index of later FAB-MAP implementations [Cummins et al., 2008a; Cummins et al., 2009] are equally applicable to CAT-SLAM. This would significantly reduce the computation time taken to update the particle weights, since the observation likelihoods could be determined using a batch update method. As mentioned, the algorithm is highly parallelisable, and could provide real-time performance on modern parallel or GPGPU processors without additional computational enhancements.

References

[Angeli et al., 2009] Angeli, A., Doncieux, S., Meyer, J. and Filliat, D. 2009, "Visual topological SLAM and global localization", *In: IEEE International Conference on Robotics and Automation*, Kobe, Japan, 2029-2034.

[Bosse et al., 2003] Bosse, M., Newman, P., Leonard, J., Soika, M., Feiten, W. and Teller, S. 2003, "An atlas framework for scalable mapping", *In: IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, 1899-1906.

[Boucheron et al., 2004] Boucheron, S., Lugosi, G. and Bousquet, O. 2004, "Concentration inequalities", *Advanced Lectures on Machine Learning*, 208-240.

[Chow et al., 1968] Chow, C. and Liu, C. 1968, "Approximating discrete probability distributions with dependence trees", *IEEE Transactions on Information Theory*, 14**,** 462-467.

[Cummins et al., 2007] Cummins, M. and Newman, P. 2007, "Probabilistic appearance based navigation and loop closing", *In: IEEE International Conference on Robotics and Automation*, Rome, Italy, 2042-2048.

[Cummins et al., 2008a] Cummins, M. and Newman, P. 2008a, "Accelerated appearance-only SLAM", *In: IEEE International Conference on Robotics and Automation*, Pasadena, California.

[Cummins et al., 2008b] Cummins, M. and Newman, P. 2008b, "FAB-MAP: Probabilistic localization and mapping in the space of appearance", *The International Journal of Robotics Research*, 27**,** 647.

[Cummins et al., 2009] Cummins, M. and Newman, P. 2009, "Highly scalable appearance-only SLAM–FAB-MAP 2.0", *In: Robotics: Science and Systems Conference*, Seattle, Washington.

[Davison et al., 2007] Davison, A., Reid, I., Molton, N. and Stasse, O. 2007, "MonoSLAM: Real-time single camera SLAM", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29**,** 1052-1067.

[Dissanayake et al., 2001] Dissanayake, M., Newman, P., Clark, S., Durrant-Whyte, H. and Csorba, M. 2001, "A solution to the simultaneous localization and map building (SLAM) problem", *IEEE Transactions on robotics and automation*, 17**,** 229-241.

[Durrant-Whyte et al., 2006] Durrant-Whyte, H. and Bailey, T. 2006, "Simultaneous localization and mapping: part I", *IEEE Robotics & Automation Magazine*, 13**,** 99-110.

[Eliazar et al., 2003] Eliazar, A. and Parr, R. 2003, "DP-SLAM: Fast, robust simultaneous localization and mapping without predetermined landmarks", *In: International Joint Conference on Artificial Intelligence*, Acapulco, Mexico, 1135-1142.

[Glover et al., 2010] Glover, A., Maddern, W., Milford, M.

and Wyeth, G. 2010. FAB-MAP + RatSLAM: Appearance-based SLAM for Multiple Times of Day. *IEEE International Conference of Robotics and Automation.* Anchorage, Alaska.

[Hähnel et al., 2003] Hähnel, D., Fox, D., Burgard, W. and Thrun, S. 2003, "A highly efficient FastSLAM algorithm for generating cyclic maps of large-scale environments from raw laser range measurements", *In: IEEE International Conference on Intelligent Robots and Systems*, Las Vegas, NV, 206–211.

[Konolige et al., 2008] Konolige, K. and M. Agrawal (2008). "FrameSLAM: From bundle adjustment to real-time visual mapping." IEEE Transactions on Robotics 24(5): 1066-1077.

[Liu et al., 2001] Liu, J., Chen, R. and Logvinenko, T. 2001, "A theoretical framework for sequential importance sampling and resampling", *Sequential Monte Carlo Methods in Practice,* 225–246.

[Milford et al., 2008a] Milford, M. and Wyeth, G. 2008a, "Mapping a suburb with a single camera using a biologically inspired SLAM system", *IEEE Transactions on Robotics*, 24**,** 1038-1053.

[Milford et al., 2008b] Milford, M. and Wyeth, G. 2008b, "Single Camera Vision-Only SLAM on a Suburban Road Network", *In: IEEE International Conference on Robotics and Automation*, Pasadena, California.

[Montemerlo et al., 2002] Montemerlo, M., Thrun, S., Koller, D. and Wegbreit, B. 2002, "FastSLAM: A factored solution to the simultaneous localization and mapping problem", *In: Proceedings of the National conference on Artificial Intelligence*, Edmonton, Canada, 593-598.

[Montemerlo et al., 2003] Montemerlo, M., Thrun, S., Koller, D. and Wegbreit, B. 2003, "FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges", *In: International Joint Conference on Artificial Intelligence*, Acapulco, Mexico, 1151-1156.

[Newman et al., 2009] Newman, P., Sibley, G., Smith, M., Cummins, M., Harrison, A., Mei, C., Posner, I., Shade, R., Schroeter, D. and Murphy, L. 2009, "Navigating, Recognizing and Describing Urban Spaces With Vision and Lasers", *The International Journal of Robotics Research,* 28**,** 1406.

[Paul et al., 2010] Paul, R. and Newman, P. 2010, "FAB-MAP 3D: Topological Mapping with Spatial and Visual Appearance", *In: IEEE International Conference on Robotics and Automation*, Anchorage, Alaska, 2649-2656.

[Sibley et al., 2010] Sibley, G., Mei, C., Reid, I. and Newman, P. 2010, "Vast-scale Outdoor Navigation Using Adaptive Relative Bundle Adjustment", *The International Journal of Robotics Research,* (in press).

[Sivic et al., 2003] Sivic, J. and Zisserman, A. 2003, "Video Google: A text retrieval approach to object matching in videos", *In: IEEE International Conference on Computer Vision*, Nice, France, 1470-1477.

[Smith et al., 1990] Smith, R., Self, M. and Cheeseman, P. 1990, "Estimating uncertain spatial relationships in robotics", *Autonomous robot vehicles,* 1**,** 167-193.

[Thrun et al., 2008] Thrun, S. and Leonard, J. 2008. Simultaneous Localization and Mapping. *In:* SICILIANO, B. (ed.) *Springer Handbook of Robotics.* Springer Berlin Heidelberg.

[Thrun et al., 2006] Thrun, S. and Montemerlo, M. 2006, "The graph SLAM algorithm with applications to large-scale mapping of urban structures", *The International Journal of Robotics Research,* 25**,** 403.

[Van der Merwe et al., 2001] Van der Merwe, R., Doucet, A., De Freitas, N. and Wan, E. 2001, "The unscented particle filter", *Advances in neural information processing systems,* 584-590.