

# Continuous Appearance-based Trajectory SLAM

Will Maddern, *Student Member, IEEE*, Michael Milford, *Member, IEEE*, and Gordon Wyeth, *Member, IEEE*

**Abstract**— This paper describes a novel probabilistic approach to incorporating odometric information into appearance-based SLAM systems, without performing metric map construction or calculating relative feature geometry. The proposed system, dubbed Continuous Appearance-based Trajectory SLAM (CAT-SLAM), represents location as a probability distribution along a trajectory, and represents appearance continuously over the trajectory rather than at discrete locations. The distribution is evaluated using a Rao-Blackwellised particle filter, which weights particles based on local appearance and odometric similarity and explicitly models both the likelihood of revisiting previous locations and visiting new locations. A modified resampling scheme counters particle deprivation and allows loop closure updates to be performed in constant time regardless of map size. We compare the performance of CAT-SLAM to FAB-MAP (an appearance-only SLAM algorithm) in an outdoor environment, demonstrating a threefold increase in the number of correct loop closures detected by CAT-SLAM.

## I. INTRODUCTION

VISUAL appearance-based localization is increasingly used for loop closure detection in metric SLAM systems. Since it relies only upon the visual similarity between images from two locations, it can perform loop closure regardless of accumulated metric error (a major cause of failure for metric SLAM systems [1]). So-called ‘appearance-based SLAM’ systems represent the environment as a series of images from discrete locations, and typically calculate image similarity based on extracted SIFT [2] descriptors.

The largest successful appearance-based SLAM experiment to date used FAB-MAP [3], and detected loop closures on a 1000km road network [4]. Development of appearance-based SLAM systems has focused on increasing the number of previously visited locations that are recognized (high recall) while maintaining low numbers of false positives (high precision). Since false positive loop closures cause corruption in most mapping systems, 100% precision is a common requirement for appearance-based loop closure detection [3].

Manuscript received September 15, 2010. This work was supported in part by the Australian Research Council and National Health and Medical Research Council under a Thinking Systems Initiative, and the Australian Federal Government under an Australian Postgraduate Award.

W. P. Maddern, M. J. Milford and G. F. Wyeth are with the School of Engineering Systems, Faculty of Built Environment and Engineering, Queensland University of Technology, Brisbane QLD 4001 Australia (e-mail: {w.maddern, michael.milford, gordon.wyeth}@qut.edu.au).

Attempts to improve the precision-recall performance of appearance-based SLAM algorithms typically require additional information not provided by descriptor-based image similarity alone; [4, 5] uses RANSAC to compare feature geometry, while [6-8] uses additional laser or stereo image sensors for 3D geometric verification. These methods still rely on matching two distinct locations using appearance alone – they discard the motion information between locations (provided by vehicle odometry), and the sequence in which the locations were visited.

This paper presents Continuous Appearance-based Trajectory SLAM (CAT-SLAM), a probabilistic approach to appearance-based loop closure detection incorporating odometric information. CAT-SLAM represents the map as a continuous trajectory which traverses all previously visited locations, and appearance is represented continuously along the trajectory, rather than at discrete points. Loop closure hypotheses are developed over a number of updates using a Rao-Blackwellised particle filter, which weights particles based on trajectory-constrained metric motion information and appearance-based observation likelihoods.

We evaluate the loop closure performance of CAT-SLAM in comparison to FAB-MAP using the New College dataset [9], previously used for various FAB-MAP experiments [6, 7]. In this environment CAT-SLAM demonstrates a threefold improvement in location recall over FAB-MAP with zero false positives.

## II. BACKGROUND

Comparatively few appearance-based SLAM systems that make use of odometric information (but do not perform metric SLAM) have been developed. [10] combines a graph relaxation algorithm for map-building with appearance-only image matching, but relies on visual matches alone for loop closure. [11] uses a POMDP to reason about likely loop closures based on sequences of location ‘fingerprints’, but has only been demonstrated for small indoor environments. Promising results for using odometry to perform ‘pose filtering’ are shown in RatSLAM [12], which uses a biologically-inspired approach to combine appearance and metric information. Combinations of FAB-MAP with RatSLAM demonstrated long-term mapping with no false positive loop closures [13, 14].

The following section describes the essential components of two SLAM systems from which components of CAT-SLAM are derived: FastSLAM and FAB-MAP. FastSLAM

[15] uses a Rao-Blackwellised particle filter and various schemes for particle resampling to perform efficient geometric SLAM. By assuming the map stored by each particle is correct, observations become conditionally independent. The joint state is represented by  $N$  particles, each with pose history  $\mathbf{X}_{0:k}$ , weight  $w$  and distribution as follows:

$$\{w_k^{(i)}, \mathbf{x}_{0:k}^{(i)}, P(\mathbf{m} | \mathbf{X}_{0:k}^{(i)}, \mathbf{Z}_{0:k})\}_i^N \quad (1)$$

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

$$\mathbf{x}_k^{(i)} \sim P(\mathbf{x}_k | \mathbf{x}_{k-1}^{(i)}, \mathbf{u}_k) \quad (2)$$

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

$$w_k^{(i)} = w_{k-1}^{(i)} \frac{P(\mathbf{z}_k | \mathbf{X}_{0:k}^{(i)}, \mathbf{Z}_{0:k-1})P(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)}, \mathbf{u}_k)}{\pi(\mathbf{x}_k^{(i)} | \mathbf{X}_{0:k-1}^{(i)}, \mathbf{Z}_{0:k}, \mathbf{u}_k)} \quad (3)$$

The particles are then resampled with replacement after normalization, where the probability of selection is proportional to the weight  $w$ . While this allows 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 [15]. Extensions have been made to the FastSLAM algorithm; however, even these state-of-the-art implementations of FastSLAM rely on accurate odometric information to close large loops [1].

FAB-MAP [3] forsakes geometric map building and instead focuses on performing SLAM in appearance space. Each unique location  $L_k$  is represented by a set of probabilities that each object  $e_i$  (that creates observation  $z_i$ ) is present in the scene.

$$\{P(e_i = 1 | L_k), \dots, P(e_{|v|} = 1 | L_k)\} \quad (4)$$

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

$$P(L_i | \mathcal{Z}^k) = \frac{P(\mathbf{Z}_k | L_i, \mathcal{Z}^{k-1})P(L_i | \mathcal{Z}^{k-1})}{P(\mathbf{Z}_k | \mathcal{Z}^{k-1})} \quad (5)$$

where  $\mathcal{Z}^k$  is a collection of previous observations up to time  $k$ ,  $P(\mathbf{Z}_k | L_i, \mathcal{Z}^{k-1})$  is assumed to be independent from all past observations and is calculated using a Chow Liu approximation [16]. Observation likelihoods are determined using the Chow Liu tree as follows:

$$P(\mathbf{Z}_k | L_i) \approx P(z_r | L_i) \prod_{q=1}^{|v|} P(z_q | z_{p_q}, L_i) \quad (6)$$

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  $P(L_i | \mathcal{Z}^{k-1})$  is estimated using a naïve motion model. The denominator of equation 5 incorporates the probability of matching to all possible locations; to estimate if a new observation comes from a previously unvisited location the model needs to consider mapped and unmapped locations as follows:

$$P(\mathbf{Z}_k | \mathcal{Z}^{k-1}) = \sum_{m \in M} P(\mathbf{Z}_k | L_m)P(L_m | \mathcal{Z}^{k-1}) + \sum_{n \in M} P(\mathbf{Z}_k | L_n)P(L_n | \mathcal{Z}^{k-1}) \quad (7)$$

where  $M$  is the set of mapped locations. Since the second term cannot be evaluated directly (as it would require knowledge of all unknown locations), a mean field or sampling estimation must be used [3].

### III. CAT-SLAM

In this section we outline our proposed appearance-based SLAM system. CAT-SLAM is derived from a ‘trajectory-based’ interpretation of the SLAM problem. It combines aspects of the geometric motion model of FastSLAM with the appearance-based observation model of FAB-MAP. As with FastSLAM, poses  $\mathbf{x}_i$  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 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 arbitrary location  $\mathbf{x}_k$  on the trajectory between two previously visited discrete locations.

The history of states is represented by a continuous trajectory  $T$ , which intersects all previously visited locations  $\mathbf{X}_{0:k}$ :

$$\mathbf{X}_{0:k} \in T \Rightarrow \mathbf{x}(t) \in T, \quad 0 \leq t \leq k \quad (8)$$

The full history of states is recovered using the continuous state  $\mathbf{x}(t)$  with continuous index  $t$ . The particular form of the trajectory  $T$  is defined by the continuous motion model of the vehicle; the simplest case of a linearly interpolated trajectory is illustrated as follows:

$$\mathbf{x}(t) = (\lceil t \rceil - t)\mathbf{X}_{\lceil t \rceil} + (t - \lfloor t \rfloor)\mathbf{X}_{\lfloor t \rfloor} \quad (9)$$

As with appearance-based systems, the map is formed by the history of states as follows:

$$\mathbf{m} = \mathbf{X}_{0:k} \Rightarrow \mathbf{m} = \mathbf{x}(0 \leq t \leq k) \quad (10)$$

The map update is performed by correcting the history of states  $\mathbf{X}_{0:k}$  when a loop closure is detected using a graph relaxation algorithm if required. The location distribution conditioned on the continuous trajectory  $T$  is therefore:

$$P(\mathbf{x}_k \in T | \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{x}_0) \quad (11)$$

where  $\mathbf{Z}_{0:k}$  is the full history of observations and  $\mathbf{U}_{0:k}$  the history of control inputs. The distribution is approximated using  $N$  particles, each with weight  $w$ , position on the trajectory  $\mathbf{x}_k$ , continuous trajectory index  $t$  and binary particle direction  $d$ :

$$\{w_k^{(i)}, \mathbf{x}_k^{(i)}, t^{(i)}, d^{(i)}\}_i^N \quad (12)$$

Figure 1 illustrates the 4 stage update process of the CAT-SLAM particle filter:

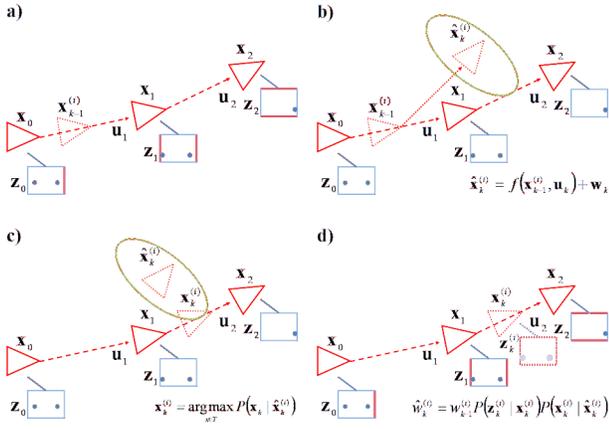


Fig. 1 – Update process of CAT-SLAM particles. a) Particles  $\mathbf{x}_{k-1}^{(i)}$  are constrained to the trajectory between previously visited locations  $\mathbf{x}_{0:k}$ . b) Proposed particle locations  $\hat{\mathbf{x}}_k^{(i)}$  are sampled from the motion model with control input  $\mathbf{u}_k$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

### A. Trajectory-based Pose Filtering

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

$$\mathbf{x}_k^{(i)} \sim P(\mathbf{x}_k \in T | \mathbf{x}_{k-1}^{(i)}, \mathbf{u}_k) \quad (13)$$

This method permits the use of a nonlinear motion model but ensures all particles remain constrained to the trajectory of previously visited locations. The particle update is performed by first generating a proposed pose  $\hat{\mathbf{x}}_k$  using the nonlinear vehicle model  $f$  given control input  $\mathbf{u}_k$  with additive Gaussian noise  $\mathbf{w}_k$ , based on direction  $d$ :

$$\hat{\mathbf{x}}_k^{(i)} = \begin{cases} f(\mathbf{x}_{k-1}^{(i)}, \mathbf{u}_k) + \mathbf{w}_k, & d = 1 \\ f(\mathbf{x}_{k-1}^{(i)}, -\mathbf{u}_k) + \mathbf{w}_k, & d = 0 \end{cases} \quad (14)$$

This allows particles to propagate in both forward and reverse directions along the trajectory. The proposed state covariance is generated by linearizing the motion model at the proposed state location with noise covariance  $\mathbf{Q}_k$ :

$$\Sigma_k^{(i)} = J_k^{(i)} \mathbf{Q}_k J_k^{(i)T}, \quad J_k^{(i)} = \frac{\partial \mathbf{x}(\mathbf{x}_{k-1}^{(i)}, \mathbf{u}_k)}{\partial \mathbf{u}} \quad (15)$$

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

$$P(\mathbf{x} | \hat{\mathbf{x}}_k) = \frac{1}{2\pi\sqrt{|\Sigma_k|}} \exp\left[-\frac{1}{2}(\mathbf{x} - \hat{\mathbf{x}}_k)\Sigma_k^{-1}(\mathbf{x} - \hat{\mathbf{x}}_k)^T\right] \quad (16)$$

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:

$$t^{(i)} = \arg \max_{t \leq k} P(\mathbf{x}(t) | \hat{\mathbf{x}}_k^{(i)}) \quad (17)$$

From this index the pose of the particle is set to the

maximum likelihood pose on the trajectory relative to the current pose:

$$\mathbf{x}_k^{(i)} = \mathbf{x}(t^{(i)}) \quad (18)$$

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

### B. Continuous Appearance Model

The location representation for each particle is extended from equation 4 to represent appearance continuously between discrete observations as follows:

$$\{P(e_i = 1 | \mathbf{x}(t^{(i)})), \dots, P(e_{|v|} = 1 | \mathbf{x}(t^{(i)}))\} \quad (19)$$

The method of generating these interpolated appearance representations is dependent on both the continuous vehicle motion model and the camera model. For the linear case of equation 9 the continuous representation of appearance is generated by interpolating between two successive discrete observations:

$$P(e_i = 1 | \mathbf{x}(t)) = (\lceil t \rceil - t)P(e_i = 1 | \mathbf{Z}_{\lceil t \rceil}) + (t - \lfloor t \rfloor)P(e_i = 1 | \mathbf{Z}_{\lfloor t \rfloor}) \quad (20)$$

The set of objects  $e_i$  that form the appearance representation must be derived from training data in a similar environment to the test environment [3].

### C. Particle Weighting and Resampling

The importance weighting of the particles combines the observation likelihood of FAB-MAP using the continuous representation of appearance with the motion prior of FastSLAM conditioned on the trajectory. The proposed weighting of each particle is as follows:

$$\hat{w}_k^{(i)} = w_{k-1}^{(i)} P(\mathbf{z}_k | \mathbf{x}_k^{(i)}) P(\mathbf{x}_k^{(i)} \in T | \mathbf{x}_{k-1}^{(i)}, \mathbf{u}_k) \quad (21)$$

The observation likelihood makes use of the Chow Liu distribution from equation 6 at location  $t$  on the trajectory:

$$P(\mathbf{z}_k | \mathbf{x}_k^{(i)}) = P(z_r | \mathbf{x}(t^{(i)})) \prod_{q=1}^{|v|} P(z_q | z_{p_q}, \mathbf{x}(t^{(i)})) \quad (22)$$

The leftmost part of equation 22 is calculated as follows:

$$P(z_q | z_{p_q}, \mathbf{x}(t^{(i)})) = \sum_{s \in \{0,1\}} P(z_q | e_q = s, z_{p_q}) P(e_i = s | \mathbf{x}(t^{(i)})) \quad (23)$$

where  $P(z_q | e_q = s, z_{p_q})$  is the detector probability and  $P(e_i = s | \mathbf{x}(t^{(i)}))$  the continuous appearance representation defined in equation 20. The motion prior is the maximum likelihood point of the motion distribution along the trajectory as found in equation 17:

$$P(\mathbf{x}_k^{(i)} \in T | \mathbf{x}_{k-1}^{(i)}, \mathbf{u}_k) = P(\mathbf{x}_k^{(i)} | \hat{\mathbf{x}}_k^{(i)}) \quad (24)$$

To represent the likelihood of a location not on the trajectory, an additional particle  $u$  representing an ‘unknown’ state is weighted as follows:

$$\hat{w}_k^u = \frac{1}{N} P(\mathbf{z}_k | \mathbf{x}_k^u) P(\mathbf{x}_k^u | \mathbf{u}_k) \quad (25)$$

These two distributions can be approximated using

information from training data as follows:

$$P(\mathbf{z}_k | \mathbf{x}_k^u)P(\mathbf{x}_k^u | \mathbf{u}_k) \approx P(\mathbf{z}_k | \mathbf{z}_{\text{avg}})P(\mathbf{u}_{\text{avg}} | \mathbf{u}_k) \quad (26)$$

where  $\mathbf{z}_{\text{avg}}$  represents an ‘average’ observation and  $\mathbf{u}_{\text{avg}}$  an ‘average’ control input, found by averaging all observations and controls in the training data set or by using the sampling method in [3]. Without this ‘unknown’ state the particle distribution represents pure localization; the probability of a state not on the trajectory is otherwise assumed to be zero. The proposed weight of each particle is normalized as follows:

$$w_k^{(i)} = \frac{\hat{w}_k^{(i)}}{\sum_j \hat{w}_k^{(j)} + \hat{w}_k^u} \quad (27)$$

The particles are resampled when the effective sample size (ESS) [17] falls below a predefined threshold. The ESS is computed as follows:

$$\text{ESS} = \frac{N}{1 + \frac{1}{N} \sum_j^N [Nw_k^{(j)} - 1]^2} \quad (28)$$

Particles are selected with probability proportional to their weight  $w_k$  using the Select with Replacement method [17]. Any particles selected to replace the ‘unknown’ particle are sampled to a uniform random location on the trajectory, which serves to counteract the effects of particle deprivation:

$$t^{(i)} \sim U(0, k), \mathbf{x}_k^{(i)} = \mathbf{x}(t^{(i)}), d^{(i)} \sim \text{round}(U(0,1)) \quad (29)$$

The value of the distribution at particle location  $\mathbf{x}_k$  is determined using a spatially selective method:

$$P(\mathbf{x}_k^{(i)}) = \frac{\sum_j^N h(i, j)}{1 + w_k^u} \quad (30)$$

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

$$h(i, j) = \begin{cases} w_k^{(j)}, & |\mathbf{x}_k^{(j)} - \mathbf{x}_k^{(i)}| \leq r \\ 0, & \text{otherwise} \end{cases} \quad (31)$$

The distribution will only reach a probability of 1 at a location if all particles are within predefined distance  $r$  of that location (causing the numerator to sum to 1), and the ‘unknown’ location weight is equal to 0. Evaluating the distribution can be performed in constant time proportional to the number of particles regardless of the number of previously visited locations.

#### IV. EXPERIMENTAL METHOD

##### A. Dataset

The dataset used for this evaluation of CAT-SLAM against FAB-MAP is presented in [9]. It comprises over 7000 panoramic images from a Point Grey Ladybug2 camera with accompanying wheel odometry (from shaft encoders on the Segway RMP) and GPS data logged at 5Hz. The route taken is a 2.5km tour of the grounds of New College, Oxford, with multiple traversals of each location in both forward and reverse directions.

Ground truth is provided by GPS locations; however, the signal is degraded in many locations throughout the dataset

(particularly through a tunnel between courtyards). Approximately 45% of the panoramic images have an associated valid GPS position; data for the precision recall curves are based only on these images for which ground truth is available.

##### B. Algorithm Details

The FAB-MAP implementation used for comparison is derived from [3]. Enhancements presented in [4] primarily reduce computation time and increase scalability, and are not required for the comparatively small dataset used for this experiment. The geometric post-verification presented in [4] is not used for either algorithm.

Training data for the codebook and Chow Liu tree were provided by a downsampled 1000 image version of the main dataset with repeated sections removed. The codebook was generated using modified sequential clustering [18] yielding 6856 visual words. The ‘average’ observation  $\mathbf{z}_{\text{avg}}$  was generated using the mean field approximation in [3]. Table 1 presents a summary of the constants used in both algorithms.

## V. RESULTS

##### A. Precision-Recall Curve

The primary performance metric is the precision-recall curve. Expected matches are defined as previously visited GPS locations within 7.5m of the current location; a true positive results if the maximum likelihood location is above the hypothesis threshold and the estimated GPS location is within 7.5m of the current location. For use in loop closure detection for metric SLAM, the desired performance is high recall at 100% precision.

TABLE 1

SUMMARY OF CONSTANTS USED IN ALGORITHMS FOR EXPERIMENTS.	
<b>FAB-MAP</b>	
$p(z_i = 1   e_i = 0)$	0
$p(z_i = 0   e_i = 1)$	0.61
$p(L_{\text{new}}   Z^{k-1})$	0.9
<b>CAT-SLAM</b>	
$p(z_i = 1   e_i = 0)$	0
$p(z_i = 0   e_i = 1)$	0.61
Translation Uncertainty $\sigma_y$	0.05 meters
Rotation Uncertainty $\sigma_\theta$	0.05 radians
Number of Particles $N$	2000
ESS Threshold	0.25
Distribution Radius $r$	2.5 meters

The precision-recall curve for both FAB-MAP and CAT-SLAM is shown in Figure 2. The result for FAB-MAP is consistent with that presented in [7] for the same dataset, achieving 16% recall at 99.5% precision and 12% recall at 100% precision. CAT-SLAM reports 38% recall at 100% precision, more than 3 times that of FAB-MAP. For loop closure detection in metric SLAM this is a clearly superior result. The trajectory-matching nature of CAT-SLAM

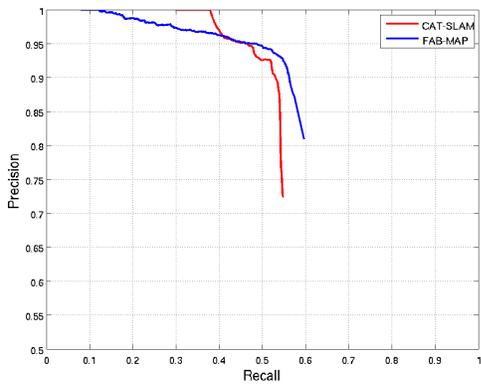


Fig. 2 – Precision-recall curve for FAB-MAP and CAT-SLAM on the New College dataset.

provides increased false positive rejection whilst increasing recall over a sequence of matching observations.

Below 95% precision FAB-MAP provides superior recall rates to CAT-SLAM. Since FAB-MAP can form a location hypothesis from a single frame, it can often recognize isolated loop closures (such as approaching a previously visited location from a new direction) where CAT-SLAM cannot. By requiring a sequence of supporting visual information over a number of updates, CAT-SLAM trades isolated loop closure detection for increased false positive rejection.

### B. Frame Recall Sequence

Figure 3 shows the frame recall sequence graphs for both FAB-MAP and CAT-SLAM. When a location is revisited multiple times, illustrated in the inset at 1000 frames, FAB-MAP recalls frames from all previous traversals, whereas CAT-SLAM matches only to a single previous visit. Maintaining multiple partial location hypotheses simultaneously reduces FAB-MAP’s ability to match a single location with certainty.

The insets at 4500 frames in Figure 3 illustrate the process of matching along a long sequence of revisited locations. CAT-SLAM maintains a strong location hypothesis throughout the path, while FAB-MAP, which uses only a naïve motion model, does not maintain strong matches across the full loop closure sequence.

### C. Loop Closure Detection

Figure 4 shows loop closures detected by both systems at 100% precision projected onto the GPS ground truth (at locations where GPS signals were valid). At this precision, FAB-MAP recalls only a small fraction of possible loop closures; large visually indistinct areas around (120, 20) are not recognized even when revisited twice. The inset in Figure 4 a) reveals inconsistent matching even in visually distinct locations. The advantages of performing trajectory-based matching in CAT-SLAM are particularly evident in Figure 4 b). Parts of the trajectory that are not visually distinct in isolation are correctly localized given a sufficient number of partial matches in the correct order over a period

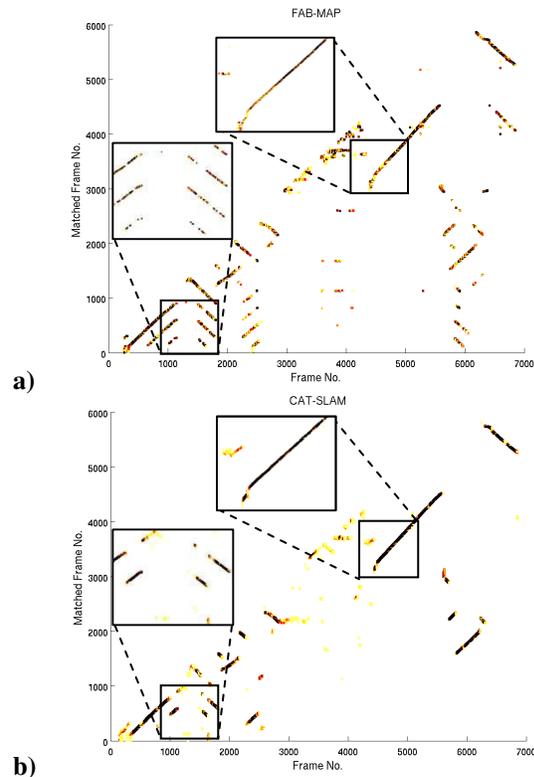


Fig. 3 – Frame recall sequence graphs for a) FAB-MAP and b) CAT-SLAM. Darker colors indicate higher likelihoods. Insets illustrate sequential frame recall performance.

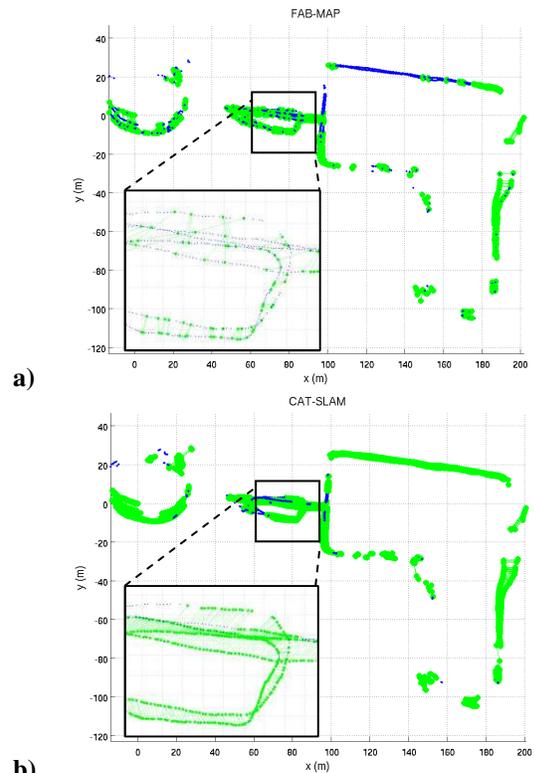


Fig. 4 – Loop closures projected on GPS ground truth for a) FAB-MAP and b) CAT-SLAM. Lighter green points indicate true positives and darker blue points false negatives. Insets illustrate sequential loop closure performance.

of time. The inset illustrates the sequential loop closures in detail; in contrast to FAB-MAP, almost every location is correctly matched to a previously visited location in the correct order.

## VI. DISCUSSION

Appearance based SLAM systems, such as FAB-MAP, represent the map using the appearance observed at discrete locations. CAT-SLAM models the appearance at all locations along a continuous trajectory, which allows odometric information to be used to improve the recall of loop closure events. By making use of odometric information that appearance-based SLAM systems typically discard, spurious false positives can be rejected, and location hypotheses can be maintained with only partial visual matches. 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 in terms of recall at 100% precision. In this case CAT-SLAM provided three times the recall rate of FAB-MAP at 100% precision.

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 required for FAB-MAP. This greatly increases the recall rates as entire sections of trajectories can be matched.

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; revisiting short sections of a path (such as crossing an intersection from a different approach) may not be detected by CAT-SLAM.

### A. Future Work

We are currently modifying the method to accommodate holonomic vehicles which do not necessarily revisit a previously traversed trajectory with an identical orientation. Explicit decoupling of orientation with trajectory will be required to support holonomic vehicles and similar platforms. We are also working to improve interpolation of appearance along the trajectory using a more sophisticated method that incorporates feature-based optical flow without evaluating 3D feature geometry. The next stage of the project is to generate topological maps that can be used and maintained for autonomous navigation tasks.

## REFERENCES

- [1] S. Thrun and J. Leonard, "Simultaneous Localization and Mapping," in *Springer Handbook of Robotics*, B. Siciliano, Ed., ed: Springer Berlin Heidelberg, 2008.
- [2] D. Lowe, "Object recognition from local scale-invariant features," in *IEEE International Conference on Computer Vision*, Corfu, Greece, 1999, p. 1150.
- [3] M. Cummins and P. Newman, "FAB-MAP: Probabilistic localization and mapping in the space of appearance," *The International Journal of Robotics Research*, vol. 27, p. 647, 2008.
- [4] M. Cummins and P. Newman, "Highly scalable appearance-only SLAM—FAB-MAP 2.0," in *Robotics: Science and Systems Conference*, Seattle, Washington, 2009.
- [5] K. Konolige and M. Agrawal, "FrameSLAM: From bundle adjustment to real-time visual mapping," *IEEE Transactions on Robotics*, vol. 24, pp. 1066-1077, 2008.
- [6] R. Paul and P. Newman, "FAB-MAP 3D: Topological Mapping with Spatial and Visual Appearance," in *IEEE International Conference on Robotics and Automation*, Anchorage, Alaska, 2010, pp. 2649-2656.
- [7] P. Newman, *et al.*, "Navigating, Recognizing and Describing Urban Spaces With Vision and Lasers," *The International Journal of Robotics Research*, vol. 28, p. 1406, 2009.
- [8] K. Konolige, *et al.*, "View-based maps," *The International Journal of Robotics Research*, vol. 29, pp. 1-17, July 2010 2009.
- [9] M. Smith, *et al.*, "The new college vision and laser data set," *The International Journal of Robotics Research*, vol. 28, p. 595, 2009.
- [10] A. Angeli, *et al.*, "Visual topological SLAM and global localization," in *IEEE International Conference on Robotics and Automation*, Kobe, Japan, 2009, pp. 2029-2034.
- [11] A. Tapus and R. Siegwart, "A cognitive modeling of space using fingerprints of places for mobile robot navigation," pp. 1188-1193.
- [12] M. Milford and G. Wyeth, "Mapping a suburb with a single camera using a biologically inspired SLAM system," *IEEE Transactions on Robotics*, vol. 24, pp. 1038-1053, 2008.
- [13] W. Maddern, *et al.*, "Augmenting RatSLAM using FAB-MAP-based Visual Data Association," presented at the Australasian Conference on Robotics and Automation, Sydney, Australia, 2009.
- [14] A. Glover, *et al.*, "FAB-MAP + RatSLAM: Appearance-based SLAM for Multiple Times of Day," presented at the IEEE International Conference of Robotics and Automation, Anchorage, Alaska, 2010.
- [15] M. Montemerlo, *et al.*, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proceedings of the National conference on Artificial Intelligence*, Edmonton, Canada, 2002, pp. 593-598.
- [16] C. Chow and C. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE Transactions on Information Theory*, vol. 14, pp. 462-467, 1968.
- [17] J. Liu, *et al.*, "A theoretical framework for sequential importance sampling and resampling," *Sequential Monte Carlo Methods in Practice*, pp. 225-246, 2001.
- [18] A. Teynor and H. Burkhardt, "Fast Codebook Generation by Sequential Data Analysis for Object Classification," *Advances in Visual Computing*, pp. 610-620, 2007.