

# Towards Persistent Localization and Mapping with a Continuous Appearance-based Topology

Will Maddern, Michael Milford and Gordon Wyeth  
School of Electrical Engineering and Computer Science  
Queensland University of Technology, Brisbane, Australia  
{w.maddern, michael.milford, gordon.wyeth}@qut.edu.au

**Abstract**— Appearance-based localization can provide loop closure detection at vast scales regardless of accumulated metric error. However, the computation time and memory requirements of current appearance-based methods scale not only with the size of the environment but also with the operation time of the platform. Additionally, repeated visits to locations will develop multiple competing representations, which will reduce recall performance over time. These properties impose severe restrictions on long-term autonomy for mobile robots, as loop closure performance will inevitably degrade with increased operation time. In this paper we present a graphical extension to CAT-SLAM, a particle filter-based algorithm for appearance-based localization and mapping, to provide constant computation and memory requirements over time and minimal degradation of recall performance during repeated visits to locations. We demonstrate loop closure detection in a large urban environment with capped computation time and memory requirements and performance exceeding previous appearance-based methods by a factor of 2. We discuss the limitations of the algorithm with respect to environment size, appearance change over time and applications in topological planning and navigation for long-term robot operation.

**Keywords**— Localization, Mapping, SLAM, Vision

## I. INTRODUCTION

Appearance based mapping provides the means to create useful topological and metric maps on resource limited robots by using appearance signatures to identify places, rather than relying on accurate metric sensing. Appearance based methods are popular as a method of detecting loop closure in range based metric maps [1, 2], and also for generating complete topological maps that can be used for path planning and navigation [3]. Appearance may refer more broadly to a robot’s sensor signatures [4], but most often refers to a snapshot image of a location from a camera mounted on the robot.

The computer vision community has provided much of the initial impetus in the advent of appearance based SLAM. Advances in image retrieval techniques, such as visual bag-of-words [5], have produced impressive results, but there is an opportunity to take advantage of the robotic context of the appearance based mapping problem. Robots on the move typically have readily available odometric information, or can

easily derive odometry from visual information. The incorporation of a motion model into the image retrieval process has the potential to greatly enhance the performance of an appearance based system. In CAT-SLAM [6, 7], the local movement information of the robot is fused with appearance information using a particle filter to dramatically improve the recall of location over the use of appearance information alone.

In this paper, we address the problem of using appearance based methods when revisiting the same location multiple times. Typical appearance-based methods have linear growth in the number of representations as locations are re-visited over and over. Consequently, both computation time and memory usage have unbounded growth over time. This problem is compounded with a corresponding fall in recall performance as multiple representations of a single place compete to be the “correct” or best representation.

The new method, called CAT-Graph, uses a combination of visual appearance and local odometry data as in CAT-SLAM, but fuses multiple visits to the same location into a topological graph-based representation. Localization is performed by propagating particles along the edges in the graph using local motion information and updating particle weights based on local appearance information. Mapping is performed by adding new motion and observation information to the graph as the robot visits new locations. Locations that are revisited are explicitly connected to the most recent location, and particles can traverse these connections to simultaneously evaluate multiple representations of a single place. The number of nodes in the graph is limited by a pruning scheme that maintains the map at constant memory requirements while minimizing information loss and maintaining performance.

Importantly, the algorithm does not rely on global metric information, or attempt to relax the measurements between nodes for global metric consistency. Instead, the topology maintains relative representations between nodes to provide local metric information to provide improvement in recall performance. We illustrate that global metric accuracy is not required to incorporate appearance and motion into our mapping and localization algorithm.

The paper proceeds with a review of recent work in appearance-based mapping and localization systems, before presenting the details of the CAT-Graph algorithm. The performance of the algorithm is demonstrated using the well-known New College dataset [8], illustrating marked

This research was supported by an Australian Research Council Special Research Initiative on Thinking Systems, TS0669699, and an Australian Postgraduate Award.

improvements in recall performance with capped memory and computation costs.

## II. RELATED WORK

Graphical representations in metric SLAM have been studied extensively for many years [9], and pose-graph optimization remains an active area of research [10, 11]. GraphSLAM [12] and other well-known topological SLAM methods form a pose graph of observations connected by edges constrained using relative motion information. However, the goal of these graphical methods is to create an optimal metric map in a global Euclidean frame. The use of relative representations without global embedding has been explored in the VO community [13], where only local metric accuracy is required. Notably, [14] combines local relative geometry without a global frame with topological appearance-based localization using FAB-MAP.

A number of recent algorithms in the field of probabilistic topological mapping approach loop closure and map construction as two parts of the same problem. The approach of [15] finds the optimal set of local metric and appearance information in the current map that best matches the current set of observations and local motion. [16] describes a system where both local metric maps and topological position are used to determine the current location within the hybrid map. A general formulation of this approach is presented in [17] using a Rao-Blackwellised particle filter across a probabilistic topology. However, these approaches have only been demonstrated in small environments.

Appearance-based localization systems do not typically address data association when revisiting locations multiple times, instead creating multiple representations for each location. [18, 19] describes a data association procedure upon loop closure detection, but the later approach adopted in [20] simply adds multiple representations. Large scale appearance-based localization is typically only demonstrated on trajectories that only feature one loop closure for each location [21], which does not address the persistence problem. Attempts to improve the recall performance of appearance-based SLAM algorithms such as FAB-MAP typically require additional information not provided by descriptor-based image similarity alone; [1] uses RANSAC to compare feature geometry, [6] requires vehicle odometry information and [22] uses additional laser or stereo image sensors for 3D geometric verification.

Constant computation time and memory requirements for mapping systems have been addressed most extensively in the metric mapping domain. Submaps have been used to achieve constant time map convergence in the approach by [23], with the assumption of repeated robot visits to all parts of the environment. Dynamic submaps have also been used to achieve a constant map update time in the approach by [24]. Occupancy grid mapping approaches typically scale linearly in terms of memory requirements with the area of the map: [25] builds on the occupancy grid approach by forming multiple occupancy maps in parallel, each representing a different timescale, and demonstrated it over a period of five weeks.

Relatively little work on constant memory or computation time mapping has occurred in the appearance-based mapping space. [26] describes a short/long term memory approach to limiting visual bag-of-words location growth, and [27] outlines

a clustering-based approach to identify unnecessary views and remove them from the map. [28] presented an information-based approach to node pruning, which removes nodes based on local visual saliency relative to its neighbors. This method provided constant-memory loop closure detection using CAT-SLAM when operating in a fixed size environment, but did not address frequent location revisiting.

In this paper we will explore the challenges of achieving constant computation time and memory requirement mapping and localization during repeated revisits in a fixed size environment using an appearance based system.

## III. ALGORITHM DETAILS

The proposed algorithm outlined in this section extends the linear ‘trajectory-based’ representation of [7] to a generalised graph-based representation. The steps of the algorithm for each new update of control input  $\mathbf{u}_k$  and visual bag-of-words observation  $\mathbf{z}_k$  are as follows:

1. Add  $\mathbf{u}_k$  and  $\mathbf{z}_k$  to the graph  $G$  as node  $N_k$ .
2. Update all particle locations in the graph using control input  $\mathbf{u}_k$ ; match to best existing location in the graph.
3. Update all particle weights using observation  $\mathbf{z}_k$ ; match to expected appearance at particle location.
4. Normalise particle weights along with an ‘unvisited location’ weight to represent the likelihood of a new place.
5. Determine if the particles indicate a loop closure has occurred; if so, create a link from  $N_k$  to the particle location.
6. Resample the particles if necessary. Distribute ‘unvisited location’ particles to random locations in the graph (to combat particle deprivation).
7. If the number of nodes exceeds the maximum, determine node information content and remove the least informative node.

The graph  $G$  defines a connected manifold which is locally Euclidian but not embedded in a global Euclidean frame. Nodes  $N_i$  in the graph are associated with observations  $\mathbf{z}_i$ , which take the form of a visual bag-of-words representation of features visible at location  $i$ . Edges  $e_{ij}$  connect node  $N_i$  to  $N_j$  and are associated with the control input  $\mathbf{u}_{ij}$  (and associated covariance matrix  $\Sigma_{ij}$ ) experienced at location  $i$ . Localization is performed using a set of  $n_p$  particles  $p_k^{(i)}$  which represent the likelihood of revisiting the previously visited location at fraction  $\alpha^{(i)}$  (between 0 and 1) along associated edge  $e_{ij}^{(i)}$  at time  $k$ . Each particle is also associated with a Boolean direction variable  $d^{(i)}$  along with a weight  $w_k^{(i)}$ .

### A. Local Graph Construction

To perform local operations on the graph we adopt a relative representation, which constructs the local neighbourhood of the graph from the reference frame of particle  $p_k^{(i)}$  to a maximum radius of  $r$ . A standard non-linear motion model  $f(\mathbf{x}, \mathbf{u})$  generates 3DOF Euclidean changes in local pose  $\mathbf{x}$  from control input  $\mathbf{u}$ , defined as follows:

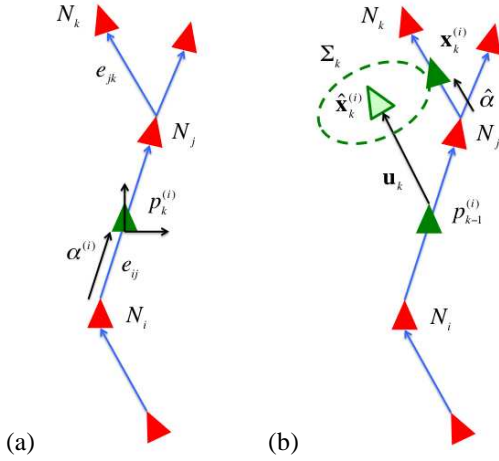


Figure 1 – Local graph and particle update diagram. (a) illustrates the local graph of particle  $p_k^{(i)}$ . The graph is constructed from the reference frame of the particle to a fixed radius  $r$ . (b) illustrates the local particle update procedure for particle  $p_k^{(i)}$ . The particle is locally propagated by control input  $\mathbf{u}_k$  (plus additive Gaussian noise  $\mathbf{w}_k$ ) to generate proposed local position  $\hat{\mathbf{x}}_k^{(i)}$ . Each edge is then tested to find the location  $\mathbf{x}_k^{(i)}$  which minimizes the Mahalanobis distance to the proposed particle location  $\hat{\mathbf{x}}_k^{(i)}$  using covariance matrix  $\Sigma_k$ . The updated particle location is set to the most likely edge  $e_{jk}$  at fraction  $\hat{\alpha}$ .

$$\mathbf{x} = [x \ y \ \theta]^T, \mathbf{u}_k = [\Delta x \ \Delta y \ \Delta \theta]^T \quad (1)$$

The first step in constructing the local graph is to determine the local position  $\mathbf{x}_i^L$  of node  $N_i$  relative to particle  $p_k^{(i)}$ . From there a breadth-first search (with fringe  $F$ ) is performed to find all nodes  $N_n$  (and associated local positions  $\mathbf{x}_n^L$ ) and edges  $e_{nm}$  and  $e_{mn}$  within a distance  $r$  of particle  $p_k^{(i)}$ , outlined in Algorithm 1. Note that the first node and associated edge to fall outside distance  $r$  is still added to the list – this ensures all edges even partly within the local graph radius are included in  $E^L$ . The direction variable  $d^{(i)}$  reverses the initial position  $\mathbf{x}_i^L$  to represent a particle in a reverse orientation along edge  $e_{ij}$ . A sample local graph is illustrated in Fig. 1 (a).

### B. Particle Position Update

To follow a localization hypothesis, each particle is propagated along edges in the graph according to the current control input  $\mathbf{u}_k$ . The proposed local change in pose  $\hat{\mathbf{x}}_k^{(i)}$  for each particle is generated by adding Gaussian noise  $\mathbf{w}_k$  with covariance  $\Sigma_k$  to the position generated by the non-linear motion model  $f(\mathbf{0}, \mathbf{u}_k)$ . However, the proposed current location will rarely correspond to a location along an existing edge in the local graph, therefore particles are assigned to the most likely location along an edge in the local graph. Formally, we seek a location  $\hat{\mathbf{x}}_k^L$  at fraction  $\hat{\alpha}$  along edge  $e_{ij}$  in  $E^L$  which minimizes the Mahalanobis distance (using covariance matrix  $\Sigma_k$ ) to the proposed particle location  $\hat{\mathbf{x}}_k^{(i)}$ . We solve for the minimum by differentiating the Mahalanobis distance along an edge  $e_{ij}$  with respect to fraction  $\hat{\alpha}$  and solving for  $\partial/\partial \hat{\alpha} = 0$ : if  $\hat{\alpha}$  is between 0 and 1 a local minimum lies along edge  $e_{ij}$ . The weight of each particle is updated based on the Gaussian likelihood function  $P(\hat{\mathbf{x}}_k^L | \hat{\mathbf{x}}_k^{(i)}, \Sigma_k)$ . This process is illustrated in Fig. 1 (b) and outlined in Algorithm 2. To reduce computational complexity the graph is only built to the size required to include any edge near the most likely particle

---

### Algorithm 1 Local Graph Construction

---

**function** localGraph( $p_k^{(i)}, r$ ):

$N^L$  is a list of nodes and local positions

$E^L$  is a list of edges

$\mathbf{x}_i^L := -\alpha^{(i)} f(\mathbf{0}, \mathbf{u}_{ij})$  or  $\mathbf{x}_i^L := -\alpha^{(i)} f([0 \ 0 \ \pi], \mathbf{u}_{ij})$  if  $d^{(i)} = 1$

**push**  $\{N_i, \mathbf{x}_i^L\}$  to fringe  $F$

**until** fringe  $F$  is empty

**pop**  $\{N_n, \mathbf{x}_n^L\}$  from fringe  $F$

**add**  $\{N_n, \mathbf{x}_n^L\}$  to  $N^L$  (if  $N_n$  not already present)

**for every edge**  $e_{nm}$  and  $e_{mn}$  linked to  $N_n$  **do**

$\mathbf{x}_m^L := f(\mathbf{x}_n^L, \mathbf{u}_{nm})$  or  $\mathbf{x}_m^L := f(\mathbf{x}_n^L, -\mathbf{u}_{nm})$

**push**  $e_{nm}$  or  $e_{mn}$  to  $E^L$

**if**  $|\mathbf{x}_m^L| < r$  **then**

**push**  $\{N_m, \mathbf{x}_m^L\}$  to fringe  $F$

**else**

**add**  $\{N_m, \mathbf{x}_m^L\}$  to  $N^L$

**end if**

**end for**

**end until**

**return**  $N^L, E^L$

---

location  $\hat{\mathbf{x}}_k^{(i)}$ . The local graph radius is set to the length of the proposed pose change  $|\hat{\mathbf{x}}_k^{(i)}|$  plus a multiple  $s$  of the maximum eigenvalue  $\lambda_1$  of covariance matrix  $\Sigma_k$  (typically set to 3 standard deviations or more). This represents a worst-case scenario for the difference between the proposed pose change  $\hat{\mathbf{x}}_k^{(i)}$  and nearest edge location  $\hat{\mathbf{x}}_k^L$ .

### C. Appearance-based Observation Update

The observation update weights particles based on the likelihood of the current observation given the expected local appearance. The likelihood is calculated by comparing the current visual bag-of-words  $\mathbf{z}_k$  to the appearance generated by interpolating between observations  $\mathbf{z}_i$  and  $\mathbf{z}_j$ , given particle  $p_k^{(i)}$

---

### Algorithm 2 Particle Position Update

---

**function** updateParticlePosition( $p_{k-1}^{(i)}, \mathbf{u}_k, \Sigma_k$ ):

$\hat{\mathbf{x}}_k^{(i)} := f(\mathbf{0}, \mathbf{u}_k) + N(0, \Sigma_k)$

$N^L, E^L := \text{localGraph}(p_{k-1}^{(i)}, |\hat{\mathbf{x}}_k^{(i)}| + s\lambda_1(\Sigma_k))$

**for every edge**  $e_{ij}$  in  $E^L$  from  $\mathbf{x}_i^L$  to  $\mathbf{x}_j^L$  in  $N^L$  **do**

**solve**  $\frac{\partial}{\partial \hat{\alpha}} [\mathbf{x}_i^L + \hat{\alpha}(\mathbf{x}_j^L - \mathbf{x}_i^L)]^T \Sigma_k^{-1} [\mathbf{x}_i^L + \hat{\alpha}(\mathbf{x}_j^L - \mathbf{x}_i^L)] = 0$  for  $\hat{\alpha}$

**if**  $0 \leq \hat{\alpha} \leq 1$  **then**

$\hat{\mathbf{x}}_k^L := \mathbf{x}_i^L + \hat{\alpha}(\mathbf{x}_j^L - \mathbf{x}_i^L)$

$P(\hat{\mathbf{x}}_k^L | \hat{\mathbf{x}}_k^{(i)}, \Sigma_k) := \exp\left(-\frac{1}{2} [\hat{\mathbf{x}}_k^L - \hat{\mathbf{x}}_k^{(i)}]^T \Sigma_k^{-1} [\hat{\mathbf{x}}_k^L - \hat{\mathbf{x}}_k^{(i)}]\right)$

        store  $\hat{\alpha}, i, j$  for maximum  $P$

**end if**

**end for**

$\alpha_k^{(i)} := \max_p \hat{\alpha}, \quad e_k^{(i)} := \max_p e_{ij}, \quad \hat{w}_k^{(i)} := w_{k-1}^{(i)} \max_p P(\hat{\mathbf{x}}_k^L | \hat{\mathbf{x}}_k^{(i)}, \Sigma_k)$

---

is at fraction  $\alpha^{(i)}$  on edge  $e_{ij}$  at time  $k$ . The weight of each particle is updated as follows:

$$\hat{w}_k^{(i)} := \hat{w}_k^{(i)} P(\mathbf{z}_k | \mathbf{z}_p, \mathbf{z}_j, \alpha^{(i)}) \quad (3)$$

The form of this likelihood function is derived from FAB-MAP and illustrated in full in [7].

#### D. New Place Detection

Since the particle set  $p_k$  is constrained to exist only along edges  $e$  in the graph, they can only represent location hypotheses for previously visited locations. To determine if the current set of observation and motion information indicates the vehicle is in a previously unvisited location we sample from an ‘unvisited’ location  $\bar{G}$  not on the graph  $G$ .

$$P(\bar{G} | \mathbf{z}_k, \mathbf{u}_k) = P(\mathbf{z}_k | \bar{G}) P(\bar{G} | \mathbf{u}_k) \quad (4)$$

The observation and motion distributions for an unvisited location  $\bar{G}$  can be approximated using information from training data as follows:

$$P(\mathbf{z}_k | \bar{G}) P(\bar{G} | \mathbf{u}_k) \approx P(\mathbf{z}_k | \mathbf{z}_{\text{avg}}) P(\mathbf{u}_{\text{avg}} | \mathbf{u}_k) \quad (5)$$

$\mathbf{z}_{\text{avg}}$  represents an ‘average’ observation and  $\mathbf{u}_{\text{avg}}$  an ‘average’ control input. These are determined using the mean field approximation or a sampling method, both presented in [8]. The proposed weighting assigned to a location not on the graph is given as follows:

$$\hat{w}_k^{\text{new}} = \frac{1}{n} P(\mathbf{z}_k | \mathbf{z}_{\text{avg}}) P(\mathbf{u}_{\text{avg}} | \mathbf{u}_k) \quad (6)$$

The new location weight is denoted by  $w^{\text{new}}$ . Note that it is not recursively updated; this represents a uniform likelihood of departing the graph at any point in time.

#### E. Resampling and Loop Closure Detection

Particle resampling is performed using the Select with Replacement method as in [7]. Any particles selected to replace the new location weight are sampled to a random edge on the graph (with a random direction). This serves to counteract the effects of particle deprivation since the proportion of particles sampled to random locations on the graph increases as the new place likelihood increases, thereby increasing the probability of detecting loop closures without requiring evaluation of every previously visited location.

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 local area in the graph. For every particle  $p_k^{(i)}$ , the location hypothesis  $P(p_k^{(i)})$  is equal to the sum of the weights of all particles within distance  $d_h$  of the current particle within its local graph. The value of  $d_h$  is selected based on the desired resolution of loop closure detection, and as such the location hypothesis is not subject to arbitrary location discretization due to local visual saliency. If the maximum location hypothesis exceeds a threshold  $T$ , a new graph edge is added between the

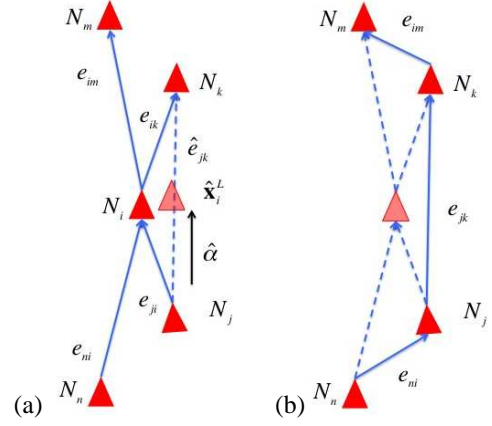


Figure 2 – Node information content and node pruning diagram. (a) illustrates the calculation of most likely local position  $\hat{\mathbf{x}}_i^L$  at fraction  $\hat{\alpha}$  on the hypothetical edge  $e_{jk}$  between node  $N_j$  and  $N_k$ . This location is used to generate an expected appearance which is compared to observation  $\mathbf{z}_i$  at node  $N_i$  to determine the information content  $I_i$ . (b) illustrates the connectivity of the local graph after node  $N_i$  has been removed.

---

#### Algorithm 3 Loop Closure Detection

---

```

for every particle  $p_k^{(i)}$  do
   $N^L, E^L := \text{localGraph}(p_k^{(i)}, d_h)$ 
  for every particle  $p_k^{(j)}$  on every edge  $e_{ij}$  in  $E^L$  do
     $P(p_k^{(i)}) := \sum w_k^{(j)}$ 
  end for
end for
if  $\max P(p_k^{(i)}) > T$  then
  create edge  $e_{ki}$  using  $\mathbf{u}_k - \alpha_k^{(i)} \mathbf{u}_{ij}, \Sigma_k + \Sigma_{ij}$ 
end if

```

---

current and matched location. This process is outlined in Algorithm 3.

The addition of a new edge on a loop closure event is crucial for increasing recall on repeated traverses of a route; it allows particles to simultaneously evaluate multiple representations of a location while recognising that all representations correspond to the same physical location (since the local graph construction will connect both locations).

#### F. Local Information-based Pruning

To limit the map to a fixed maximum size we extend the information-based trajectory pruning approach of [28] to a graphical representation. The pruning stage is performed before each particle update process. For each new node added to the graph, if the total number of nodes in the graph exceeds a preset number, the node with the lowest information content relative to its neighbors is removed and replaced with a direct link between its neighbors.

To find the information content of a node  $N_i$  relative to its neighbors  $N_j$  and  $N_k$  we compare the observation  $\mathbf{z}_i$  to that generated by interpolating neighboring observations  $\mathbf{z}_j$  and  $\mathbf{z}_k$  along a proposed edge  $e_{jk}$  that bypasses node  $N_i$ . If the proposed edge  $e_{jk}$  produces an adequate representation for observation  $\mathbf{z}_i$ , then node  $N_i$  can be removed with minimal loss of information. This process is illustrated Fig. 2 (a) and outlined in Algorithm

---

**Algorithm 4** Node Information Content

---

**for** every node  $N_i$  **do**  
  **for** every edge pair  $e_{ji}$  from  $N_j$  and  $e_{ik}$  to  $N_k$  **do**  
     $\mathbf{x}_j^L := f(\mathbf{0}, -\mathbf{u}_{ji})$ ,  $\mathbf{x}_k^L := f(\mathbf{0}, \mathbf{u}_{ik})$ ,  $\Sigma_{jk} := \Sigma_{ji} + \Sigma_{ik}$   
  **solve**  $\frac{\partial}{\partial \hat{\mathbf{x}}_i^L} [\mathbf{x}_j^L + \hat{\mathbf{x}}_i^L(\mathbf{x}_k^L - \mathbf{x}_j^L)]^T \Sigma_{jk}^{-1} [\mathbf{x}_j^L + \hat{\mathbf{x}}_i^L(\mathbf{x}_k^L - \mathbf{x}_j^L)] = 0$   
     $\hat{\mathbf{x}}_i^L := \mathbf{x}_j^L + \hat{\mathbf{x}}_i^L(\mathbf{x}_k^L - \mathbf{x}_j^L)$   
     $P(\hat{\mathbf{x}}_i^L | \mathbf{x}_j^L, \mathbf{x}_k^L) := \exp\left(-\frac{1}{2} [\hat{\mathbf{x}}_i^L]^T \Sigma_{jk} [\hat{\mathbf{x}}_i^L]\right)$   
     $I_i = -\log P(\mathbf{z}_i | \mathbf{z}_j, \mathbf{z}_k, \hat{\mathbf{x}}_i^L) P(\hat{\mathbf{x}}_i^L | \mathbf{x}_j^L, \mathbf{x}_k^L)$   
  **end for**  
**end for**

---

---

**Algorithm 5** Node Pruning

---

**if**  $k > n_n$  **then**  
  find  $N_i$  with minimum  $I_i$  between nodes  $N_j$  and  $N_k$   
  create edge  $e_{jk}$  between  $N_j$  and  $N_k$  using  $\mathbf{u}_{ji} + \mathbf{u}_{ik}$ ,  $\Sigma_{ji} + \Sigma_{ik}$   
  **for** every edge  $e_{ni}$  from  $N_n$  to  $N_i$  where  $n \neq j$  **do**  
    create edge  $e_{nj}$  using  $\mathbf{u}_{ji} - \mathbf{u}_{ni}$ ,  $\Sigma_{ji} + \Sigma_{ni}$   
  **for** every edge  $e_{im}$  from  $N_i$  to  $N_m$  where  $m \neq k$  **do**  
    create edge  $e_{km}$  using  $\mathbf{u}_{im} - \mathbf{u}_{ik}$ ,  $\Sigma_{im} + \Sigma_{ik}$   
  **end for**  
  delete node  $N_i$  and observation  $\mathbf{z}_i$ , edges  $e_{ji}$  and  $e_{ik}$   
**end if**

---

4. The information content  $I_i$  of node  $N_i$ , is defined as the negative log-likelihood of the odometric and appearance-based match between node  $N_i$  and the proposed location  $\hat{\mathbf{x}}_i^L$  along the proposed edge between  $N_j$  and  $N_k$ . Unlike in [28], nodes may have multiple neighbors due to explicit loop closure events, and therefore the information content of all proposed edges between neighbors must be evaluated to find the bypassing edge with highest information content.

To avoid unbounded growth in storage requirements, nodes are removed from the graph once the total number of nodes exceeds a threshold  $n_n$ . The node  $N_i$  with minimum information content  $I_i$  is deleted from the graph, and the proposed edge  $e_{jk}$  between neighbors  $N_j$  and  $N_k$  is added. All other nodes connected to  $N_i$  are re-routed to neighbors  $N_j$  and  $N_k$  to preserve the connectivity of the graph. Particles previously on edges connected to  $N_i$  are relocated to the new edges. This process is illustrated in Fig. 2 (b) and outlined in Algorithm 5.

#### IV. EXPERIMENTAL SETUP

To facilitate a direct comparison to FAB-MAP and previous implementations of CAT-SLAM, we use the same evaluation dataset as that presented in [6]. To differentiate the algorithms we refer to the novel representation presented in this paper as a Continuous Appearance-based Topological Graph, or CAT-Graph.

##### A. Testing Environment

The urban dataset used for this evaluation is presented in [8]. It consists of 8127 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, pictured in Fig. 3, with multiple traversals of each location in both forward and reverse directions (a total of 5 traversals of the quadrangle area). Ground truth is nominally provided by GPS locations; however, the signal is degraded in many locations throughout the urban dataset (particularly through a tunnel between courtyards). Approximately 45% of the panoramic images have an associated valid GPS position; recall data for the precision recall curves is instead based on a hand-corrected trajectory from wheel odometry and manual loop closures, which provides a topologically correct location for every frame.

##### B. Algorithm Parameters

As FAB-MAP, CAT-SLAM and CAT-Graph only require appearance information, no camera calibration or image registration is required. Feature descriptors are extracted using SURF [29], and a fast approximate nearest neighbour algorithm [30] was used to find the corresponding visual word for each descriptor.

The FAB-MAP implementation used for comparison is openFABMAP [31], which produces results comparable to those presented in [2]. Parameters for the detector functions of FAB-MAP were taken from [19]. The parameters and results for CAT-SLAM on the same dataset are taken from [6]. The

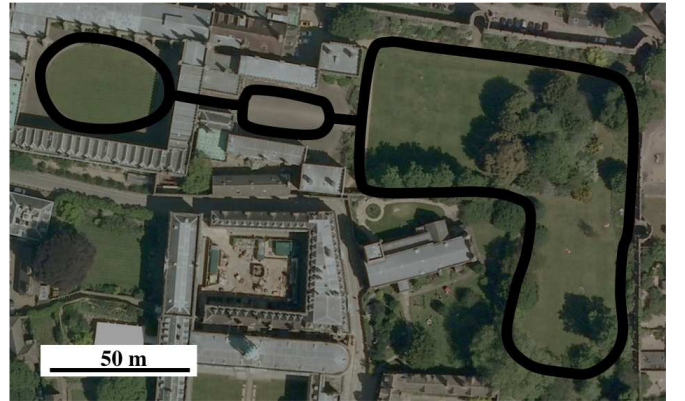


Figure 3 – Aerial view of test environment with route overlaid.

TABLE 1 – ALGORITHM PARAMETERS

<i>openFABMAP</i>	
$p(z_i = 1   e_i = 0)$	0
$p(z_i = 0   e_i = 1)$	0.61
$p(L_{new}   Z^{t-1})$	0.9
<i>CAT-SLAM</i>	
$p(z_i = 1   e_i = 0)$	0
$p(z_i = 0   e_i = 1)$	0.61
Number of Particles $N$	2000
ESS Threshold	0.25
Distribution Radius $r$	2.5m
<i>CAT-Graph</i>	
$p(z_i = 1   e_i = 0)$	0
$p(z_i = 0   e_i = 1)$	0.61
Number of Particles $n_p$	1000
Number of Nodes $n_n$	2000
Particle Update Graph Size $s$	3 $\sigma$
ESS Threshold	0.25
Distribution Radius $d_i$	2.5m
Hypothesis Threshold $T$	0.9



codebook was generated using modified sequential clustering [32] yielding 5000 visual words. Parameters for the three algorithms are presented in Table 1.

## V. RESULTS

### A. Precision-Recall Performance

To assess the performance of the CAT-Graph algorithm in comparison to openFABMAP and CAT-SLAM, we examine the precision-recall curves they produce for the test environment. Expected matches are defined as previously visited locations within 7.5m of the current location. The desired performance is high recall at 100% precision.

Fig. 4 presents the precision-recall curves produced by openFABMAP, CAT-SLAM and two variants of the CAT-Graph algorithm; one using 1000 particles and one with 1000 particles and a limit of 2000 nodes. Both CAT-Graph variants provide almost double the recall of CAT-SLAM and 7 times the recall of openFABMAP at 100% precision, but do not differ significantly from each other despite the difference in memory scaling. This demonstrates the effectiveness of the information-based node-pruning algorithm; for this environment, localization performance is not significantly affected despite the removal of approximately 5000 observations from the graph.

### B. Loop Closure Distribution

To assess the effects of using loop closure events to inform the topological graph construction (and therefore to examine the improvement gained over other appearance-based SLAM algorithms which do not explicitly perform data association) we examine the loop closure distribution for all four algorithms. Fig. 6 shows the loop closures detected at 100% precision projected onto the hand-corrected ground truth.

Both variants of CAT-Graph detect a significant number more loop closures than openFABMAP and CAT-SLAM, and the distribution of loop closures are uniform (they are not concentrated at any particular location but rather spread across the environment).

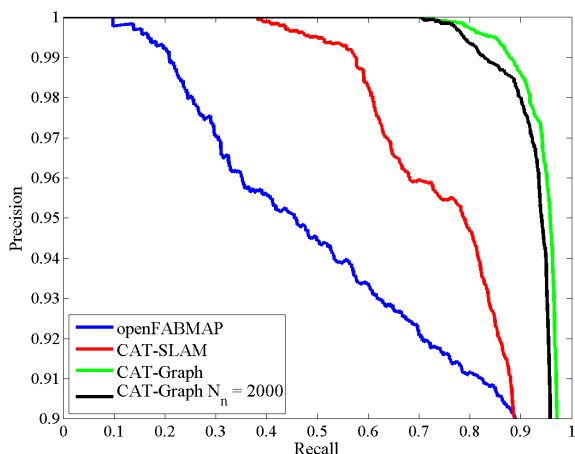


Figure 4 – Precision-Recall curve for four algorithm variants on the New College dataset. The two CAT-Graph variants provide greatly increased recall performance over CAT-SLAM and openFABMAP despite the differences in computational and memory scaling.

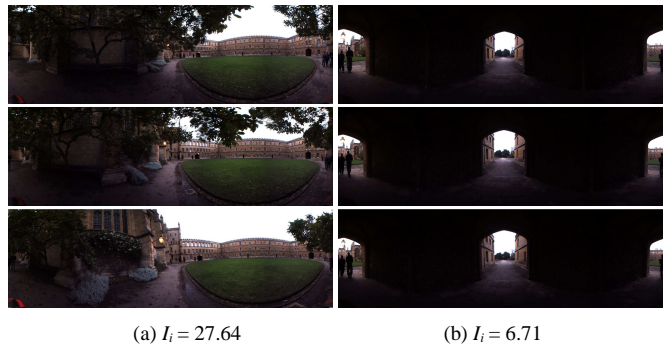


Figure 5 – Information content illustration. (a) illustrates a sequence of nodes with high information content. In this case, close proximity to buildings and trees cause a high degree of difference between sequential frames, yielding high relative information. (b) illustrates a sequence through a tunnel, with very little visual change between successive frames and therefore a low information content.

### C. Node Information Content

Fig. 5 illustrates a pair of frame sequences at different locations in the environment. The first sequence involves the robot moving under an overhanging tree and passing a distinctive building, while the second involves the robot travelling through a dark, featureless tunnel. As shown below the sequences, the information content for the central frame in the second sequence is significantly lower than that calculated for the central frame in the first sequence.

### D. Computational and Memory Scaling

Fig. 7 presents the computation and storage requirements for openFABMAP, CAT-SLAM and the two CAT-Graph variants. The computation time does not include feature extraction and visual word classification (on average 800ms per frame), as these will be identical for all three algorithms. The difference in computational scaling is clear, with openFABMAP reduced to update rates below 1Hz by the end of the dataset due to the linear increase in computational requirements. The CAT-Graph configuration with 2000 nodes requires the greatest amount of time per update (as the information content of each node is assessed), but still remains approximately constant over the length of the dataset. Spikes in the computation graph are due to the evaluation of nodes with multiple neighbors, as the information content between every combination of neighbors is assessed.

The storage requirements scale linearly with operation time for all but the CAT-Graph configuration with limited nodes. All algorithms require the codebook and Chow-Liu tree from training data, which contribute to the initial ~4MB at the start of the dataset. Storage requirements initially increase linearly for both algorithms; CAT-SLAM requires additional storage for location and odometry information for the continuous trajectory as well as particle locations, weights and directions. However, once the number of locations reaches 2000, CAT-Graph with limited nodes does not require more storage, whereas openFABMAP and CAT-SLAM memory requirements continue to increase linearly with the number of frames.

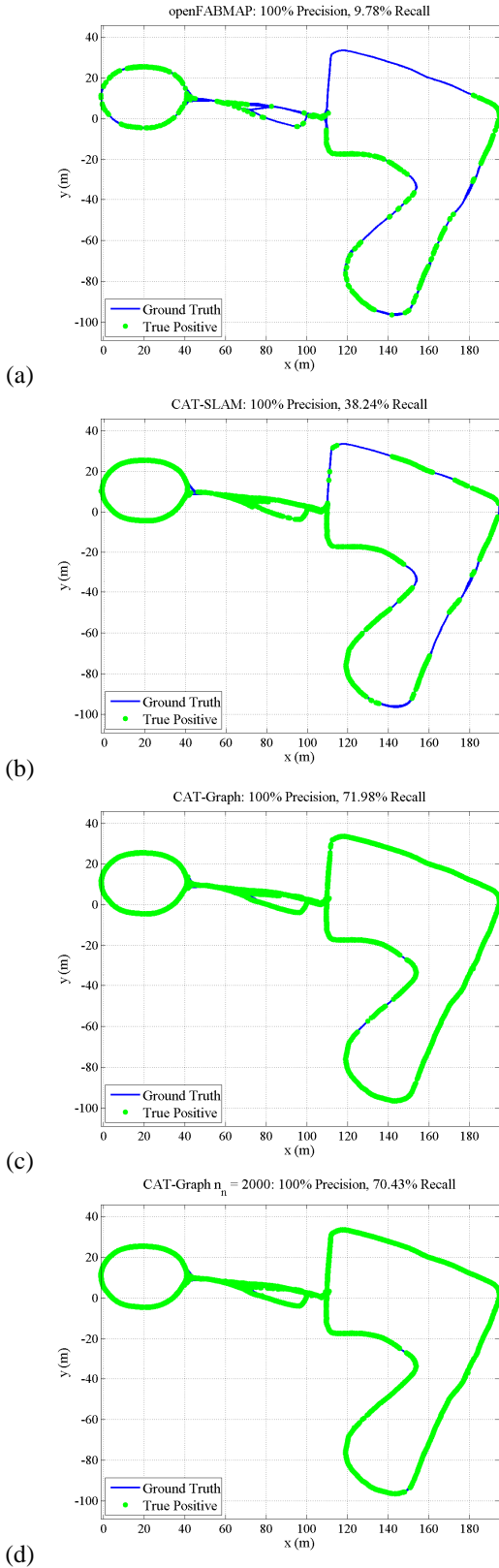


Figure 6 – Loop closure distribution at 100% precision overlaid on hand-corrected ground truth based on wheel odometry information. CAT-Graph detects consistently more loop closures than both openFABMAP and CAT-SLAM in all areas of the environment.

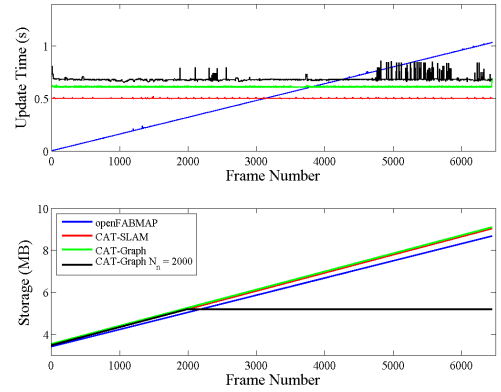


Figure 7 - Computation time and memory requirements for openFABMAP, CAT-SLAM and CAT-Graph. All CAT-SLAM and CAT-Graph variants provide constant computational time scaling, and the CAT-Graph variant with limited nodes provides constant memory scaling over time.

Loop closure events in CAT-Graph also increase the number of edges in the map and therefore memory requirements. However, the worst case memory requirement for storage of edges is when an edge exists for every node pair, making it  $O(N^2)$  in number of nodes; the worst case edge storage requirements do not grow over time if the number of nodes is fixed.

## VI. DISCUSSION

The use of a topological graph and local metric representations along with loop-closure-informed graph links provide CAT-Graph with significant performance increases over CAT-SLAM and openFABMAP. In this section we discuss the key insights gained from this work and directions for future work.

### A. Improved Loop Closure Through Hypothesis Aggregation

The ability to combine multiple hypotheses for multiple representations of the same location is crucial for long-term operation in a fixed-size environment. CAT-Graph consistently detects more loop closures over successive revisits to the same location, whereas CAT-SLAM develops multiple separate representations of the same trajectory but does not identify that all correspond to the same location. Unlike CAT-SLAM, where location hypotheses compete upon revisiting a location, by creating explicit links between locations when loop closures are detected, multiple hypotheses from multiple representations of the same location support, rather than compete with, each other.

### B. Saliency-based Map Pruning

The node pruning scheme enables localization and mapping in a fixed size environment without memory requirements increasing linearly with operation time. Nodes are removed based on visual saliency, with less distinctive areas of the environment generating fewer nodes. The selection of the maximum number of nodes  $n_n$  depends on the memory available to the robot and the visual saliency of the environment. Experiments presented in [28] for the CAT-SLAM algorithm illustrate a graceful degradation of recall performance with reduced numbers of nodes and particles; however, determining the absolute minimum number of nodes

sufficient to fully represent a particular fixed-size environment remains an open problem, and the constant-memory approach will fail if the robot continuously explores new locations. When maintaining a constant memory map in a fixed-size environment, removing the least informative nodes ensures future localization performance is minimally affected.

### C. Future Work

Incorporating loop closure events into graph construction provides performance benefits when repeatedly traversing the same location. However, in order for the loop closure events to be detected, the appearance of the environment cannot change significantly between visits. While FAB-MAP provides a level of robustness, matching still fails over large changes in environmental appearance, such as experienced during the course of a day [33]. Explicit modeling of appearance change over time as in [34] could enable persistent localization and mapping over longer time periods.

Along with its mapping and localization capabilities, the graph-based representation of CAT-Graph forms a suitable basis for mobile robot path planning and navigation. A world representation which is globally topological and locally metric, such as that provided by CAT-Graph, is suitable for autonomous robot operations as demonstrated in [3]. To use the global topological plan for robot navigation, it must be integrated with a local planning mechanism. This can be accomplished by generating the local metric graph representation at the current location, and planning a local metric path towards the nodes selected by the global topological planner, such as in [28].

In conclusion, we believe that the use of loop closure events to inform future appearance-based matches, along with a framework for constant computational and memory resource usage, are important steps towards persistent robot autonomy.

### REFERENCES

- [1] K. Konolige et al., "View-based maps." *The International Journal of Robotics Research*, 2009. 29(8): p. 1-17.
- [2] P. Newman et al., "Navigating, Recognizing and Describing Urban Spaces With Vision and Lasers". *The International Journal of Robotics Research*, 2009. 28(11-12): p. 1406.
- [3] M. Milford and G. Wyeth, "Persistent Navigation and Mapping using a Biologically Inspired SLAM System." *The International Journal of Robotics Research*, 2010,29: 1131-1153.
- [4] P.E. Rybski et al., "Appearance-based minimalistic metric SLAM". in *IEEE International Conference on Intelligent Robots and Systems*, Las Vegas, NV, 2003.
- [5] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos." in *IEEE International Conference on Computer Vision*. 2003. Nice, France.
- [6] W. Maddern, M. Milford, and G. Wyeth, "Continuous Appearance-based Trajectory SLAM". in *IEEE International Conference on Robots and Automation*. 2011. Shanghai, China.
- [7] W. Maddern, M. Milford and G. Wyeth, "CAT-SLAM: Probabilistic localization and mapping using a continuous appearance-based trajectory," *The International Journal of Robotics Research*, 31(4): 429-451, 2012.
- [8] M. Smith et al., "The new college vision and laser data set." *The International Journal of Robotics Research*, 2009. 28(5): p. 595.
- [9] S. Thrun and J. Leonard, "Simultaneous Localization and Mapping", in *Springer Handbook of Robotics*, B. Siciliano, Editor. 2008, Springer Berlin Heidelberg.
- [10] M. Kaess et al., "iSAM2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering" in *IEEE International Conference on Robots and Automation*. 2011. Shanghai, China.
- [11] R. Kummerle et al., "g2o: A general framework for graph optimization" in *IEEE International Conference on Robots and Automation*. 2011. Shanghai, China.
- [12] S. Thrun and M. Montemerlo, "The graph SLAM algorithm with applications to large-scale mapping of urban structures". *The International Journal of Robotics Research*, 2006. 25(5-6): p. 403.
- [13] D. Nister, O. Naroditsky, and J. Bergen. "Visual odometry" 2004: IEEE.
- [14] G. Sibley et al., "Vast-scale Outdoor Navigation Using Adaptive Relative Bundle Adjustment." *The International Journal of Robotics Research*, 2010:
- [15] E. Olson, "Robust and efficient robotic mapping" PhD Thesis. 2008, Massachusetts Institute of Technology.
- [16] J. Blanco, J. Fernandez-Madrigo, and J. Gonzalez, "Toward a Unified Bayesian Approach to Hybrid Metric-Topological SLAM". *IEEE Transactions on Robotics*, 2008. 24(2): p. 259-270.
- [17] A. Ranganathan and F. Dellaert, "Online Probabilistic Topological Mapping". *The International Journal of Robotics Research*, 2011.
- [18] M. Cummins and P. Newman. "Probabilistic appearance based navigation and loop closing." in *IEEE International Conference on Robotics and Automation*. 2007. Rome, Italy.
- [19] M. Cummins and P. Newman, "FAB-MAP: Probabilistic localization and mapping in the space of appearance." *The International Journal of Robotics Research*, 2008. 27(6): p. 647.
- [20] M. Cummins and P. Newman, "Appearance-only SLAM at large scale with FAB-MAP 2.0." *The International Journal of Robotics Research*, 2010.
- [21] M. Cummins and P. Newman. "Highly scalable appearance-only SLAM-FAB-MAP 2.0." in *Robotics: Science and Systems Conference*. 2009. Seattle, Washington.
- [22] R. Paul and P. Newman. "FAB-MAP 3D: Topological Mapping with Spatial and Visual Appearance." in *IEEE International Conference on Robotics and Automation*. 2010. Anchorage, Alaska.
- [23] J. Leonard and P. Newman. "Consistent, convergent, and constant-time SLAM." in *ICJAI*, Acapulco, Mexico, 2003.
- [24] S. Thrun et al. "SLAM updates require constant time." in *Workshop on the Algorithmic Foundations of Robotics*, 2002.
- [25] P. Biber and T. Duckett, "Experimental analysis of sample-based maps for long-term SLAM." *The International Journal of Robotics Research*, 2009. 28(1): p. 20-33.
- [26] M. Labbe and F. Michaud, "Memory management for real-time appearance-based loop closure detection." in *IEEE Conference on Intelligent Robots and Systems*, San Francisco, CA, 2011.
- [27] K. Konolige and J. Bowman, "Towards lifelong visual maps," in *IEEE Conference on Intelligent Robots and Systems*, St. Louis, MO, 2009.
- [28] W. Maddern, M. Milford and G. Wyeth, "Capping computation time and storage requirements for appearance-based localization with CAT-SLAM," in *IEEE International Conference on Robotics and Automation*, Minneapolis, MN, 2012.
- [29] H. Bay, T. Tuytelaars, and L. Van Gool. "Surf: Speeded up robust features." in *European Conference on Computer Vision*. 2006. Graz, Austria: Springer.
- [30] S. Arya et al., "An optimal algorithm for approximate nearest neighbor searching fixed dimensions." *Journal of the ACM*, 1998. 45(6): p. 891-923.
- [31] A. Glover et al. "OpenFABMAP: An Open Source Toolbox for Appearance-based Loop Closure Detection." in *IEEE International Conference on Robotics and Automation*. 2012. St Paul, United States: IEEE.
- [32] A. Teynor and H. Burkhardt, "Fast Codebook Generation by Sequential Data Analysis for Object Classification". *Advances in Visual Computing*, 2007: p. 610-620.
- [33] A. Glover et al., "FAB-MAP + RatSLAM: Appearance-based SLAM for Multiple Times of Day", in *IEEE International Conference of Robotics and Automation*. 2010: Anchorage, Alaska.
- [34] F. Dayoub and T. Duckett. "An adaptive appearance-based map for long-term topological localization of mobile robots." *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008.