A Unified Bayesian Framework for Global Localization and SLAM in Hybrid Metric/Topological Maps

Stephen Tully, George Kantor, Howie Choset

October 9, 2011

Abstract

We present a unified filtering framework for hybrid metric/topological robot global localization and SLAM. At a high level, our method relies on a topological graph representation whose vertices define uniquely identifiable places in the environment and whose edges define feasible paths between them. At a low level, our method generalizes to any detailed metric submapping technique. The filtering framework we present is designed for multi-hypothesis estimation in order to account for ambiguity when closing loops and to account for uniform uncertainty when initializing pose estimates. Our implementation tests multiple topological hypotheses through the incremental construction of a hypothesis forest with each leaf representing a possible graph/state pair at the current time step. Instead of using a heuristic approach to accept or reject hypotheses, we propose a novel Bayesian method that computes the posterior probability of each hypothesis. In addition, for every topological hypothesis, a metric estimate is maintained with a local Kalman filter. Careful pruning of the hypothesis forest keeps the growing number of hypotheses under control while a garbage-collector hypothesis is used as a catch-all for pruned hypotheses. This enables the filter to recover from unmodeled disturbances such as the kidnapped robot problem.

1 Introduction

Simultaneous localization and mapping (SLAM) is the task of mapping an environment with a mobile robot while simultaneously localizing the robot in the constructed map. Existing solutions typically rely on one of three map representations: obstacle-based grid maps or particle maps (Burgard et al. 1996, 1998; Fox et al. 1999b; Grisetti et al. 2005; Eliazar and Parr 2003), feature-based metric maps (Smith et al. 1990; Dissanayake et al. 2001; Durran-Whyte and Bailey 2006; Bailey and Durran-Whyte 2006; Montiel et al. 2006; Civera et al. 2007; Tully et al. 2008, 2010; Montemerlo et al. 2002, 2003), and topological maps (Choset and Burdick 1995; Nourbakhsh et al. 1995; Simmons and Koenig 1995; Cassandra et al. 1996; Choset and Nagatani 2001; Dudek et al. 1996; Tomatis et al. 2002; Ranganathan et al. 2006; Savelli and Kuipers 2004; De et al. 2008; Tully et al. 2009; Kuipers and Byun 1991; Tully et al. 2007; Lisien et al. 2005; Blanco et al. 2008, 2007; Kuipers et al. 2004; Angeli et al. 2008; Kouzoubov and Austin 2004). Arguably, the most efficient of the three is topological mapping, as it concisely represents an environment as a graph, whose vertices are interesting "places" and whose edges represent the paths between them (Kuipers and Byun 1991). Unfortunately, adopting a topological representation often comes at the expense of environmental detail.

Recently, hybrid methods have been proposed that combine the efficiency and conciseness of a topological implementation with the detail of metric mapping (Lisien et al. 2005; Tully et al. 2007; Blanco et al. 2008, 2007; Kuipers et al. 2004; Angeli et al. 2008; Kouzoubov and Austin 2004; Thrun and Montemerlo 2006). We believe there are three classes of such methods that are commonly used within the context of localization and mapping: the first includes algorithms that essentially lay down topological vertices intermittently along the robot path as the robot perceives the surrounding environment (Thrun and Montemerlo 2006; Angeli et al. 2008). For each vertex, sensed information is stored that can be used to close loops when, for example, a visual fingerprint is observed for a second time (Angeli et al. 2008). We should note that, despite its use of graphical vertices for pose inference, this first class also falls into the category of purely metric methods due to its complete reliance on metric positioning.

The second class of hybrid topological methods includes algorithms that subdivide an environment into submaps whose connectivity are represented by a higher level topological graph but whose placement in the environment are not precisely defined (Blanco et al. 2008, 2007; Kouzoubov and Austin 2004). The submaps are typically defined with their own coordinate frame, but the origins of these coordinate frames do not have any physical meaning. For this class, the pose of the robot is determined by estimating the submap in which the robot resides along with the robot's pose within that submap. Also, due to the fact that vertices lack physical meaning, the local transformations between submaps must also be continuously estimated.

The third class (Lisien et al. 2005; Tully et al. 2007; Werner et al. 2009a,b; Kuipers et al. 2004), to which we believe our work is most closely related, associates vertices in the topological graph with true anchored points in the environment with physical meaning that the robot can precisely navigate to in order to deterministically transition to a new subspace in the global map. Thus, the representative topology is unique for the environment, is repeatable upon subsequent experiments, and is, most importantly, stable upon revisiting vertices via sensor-based control. For this reason, true high level estimation can be performed at the topological level without a dependence on metric positioning.

In our previous work (Tully et al. 2007, 2009), we investigated just two components of the work that we are presenting in this paper. In (Tully et al.

2007), we focused on hybrid filtering for metric/topological localization. This original work did not solve the full SLAM problem, only intermittently updated the robot's pose estimate, and could not robustly recover from unmodeled disturbances. Likewise in (Tully et al. 2009), we investigated SLAM as a multi-hypothesis topological loop closing problem. Unfortunately, this previous approach was not designed with a unified framework that could also localize the robot, did not simultaneously solve the metric component of hybrid estimation, and was also not robust to unmodeled disturbances.

In this paper, we seek to extend our previous work in each of the aspects that our previous papers were lacking. We present a unified filtering framework for performing global localization and SLAM in hybrid metric/topological maps. Our approach factors the filtering problem into a discrete estimation problem (estimating a topological graph/state pair) and a continuous problem (estimating the relative metric pose of the robot along with the set of local metric maps attached to the topology).

For the discrete component of our factored estimation problem, one challenge is the so-called loop-closing problem in which the filter must detect when a robot has returned to a previously visited vertex in the topological graph. This can be especially difficult for a map with *perceptual aliasing*, where multiple "places" are indistinguishable to the robot. To solve this problem, the robot must reason about the connectivity of the graph via the sequence of observations it obtains during an experiment. To solve this problem, we introduce a novel algorithm for the incremental construction of a *hypothesis forest* that is used to evaluate possible topological hypotheses. We then infer the most likely topological graph and robot state by embedding a novel recursive Bayesian update procedure into the hypothesis expansion process.

The contributions of the work presented in this paper are: 1) the introduction of a novel hypothesis forest expansion algorithm specific to edge-ordered graphs for proposing possible topological map/state pairs, 2) a new algorithm for recursively computing the posterior probability of a topological hypothesis given a sequence of sensor measurements and a novel prior that favors simplified maps, 3) the design of conservative pruning rules that reduce the number of hypotheses in the forest, and 4) the use of a garbage-collector hypothesis that is a catch-all for evaluating the likelihood of pruned hypotheses.

2 Related Work

When addressing global localization and simultaneous localization and mapping (SLAM), every approach must maintain and update a representation of the robot map/pose probability distribution. For both global localization and SLAM, the existing approaches can be classified into either metric or topological methods.

2.1 Metric Localization

In (Burgard et al. 1996, 1998; Fox et al. 1999b), Markov localization is used to determine a robot's pose spatially in the environment. The method first discretizes the environment with some desired resolution, then assigns a probability to every state. Thus, the robot is localized in a fine grid. The number of states S resulting from this discretization depends on the desired resolution and the environment scale, and processing this representation has an $O(S)$ complexity (Fox et al. 1999b). This method does not scale well for a fine resolution in a large environment.

Monte Carlo localization (Fox et al. 1999a; Dellaert et al. 1999; Thrun et al. 2000) is similar in that it represents the probability distribution over robot poses with a set of weighted samples of the state space, called particles. The update of M particles can be implemented with complexity $O(M)$ (Thrun et al. 2000). However, representing the initial uniform probability distribution requires a sampling of particles over the entire state space. Since the convergence of this method depends on initializing at least one particle near the true pose, the number of particles increases with environment size. An example that demonstrates the inefficiency of this approach is shown in Fig. 1-(a).

Multi-hypothesis Kalman filtering is another popular method for metric localization that invokes a separate Kalman filter for each pose that could potentially explain the initial sensor measurement (Jensfelt and Kristensen 2001; Roumeliotis and Bekey 2000). Even this technique can become computationally expensive in the case of large environments with a commonly repeated feature. Also, compared to topological localization, which we will discuss shortly, multi-hypothesis Kalman filtering will typically require more hypotheses for a given environment (see Fig. 1-(b) vs. Fig. 1-(c)).

2.2 Topological and Hybrid Localization

Topological methods discretize the environment to a minimal number of abstract vertices in a graph. Thus, topological state space representations generally scale well with environment size. A common method is to use topological features or fingerprints to heuristically localize a robot to a vertex in the graph (Tully et al. 2007; Choset and Nagatani 2001). In topologically rich environments, though, similarity in topological vertex characteristics is common, and additional information is necessary to resolve these ambiguities. The choice of detailed information attached to the topology varies by implementation, as does the representation of the robot pose probability distribution. Some researchers apply Markov methods (Nourbakhsh et al. 1995; Simmons and Koenig 1995; Cassandra et al. 1996), while others apply *graph-matching* (Choset and Nagatani 2001; Kuipers and Byun 1991; Dudek et al. 1996), which uses observations of neighboring vertices to prune a set of candidate locations. The complexity of topological methods varies with the implementation.

In (Tully et al. 2007), we previously investigated hybrid filtering for metric/topological localization. Unlike the work we are presenting in this paper,

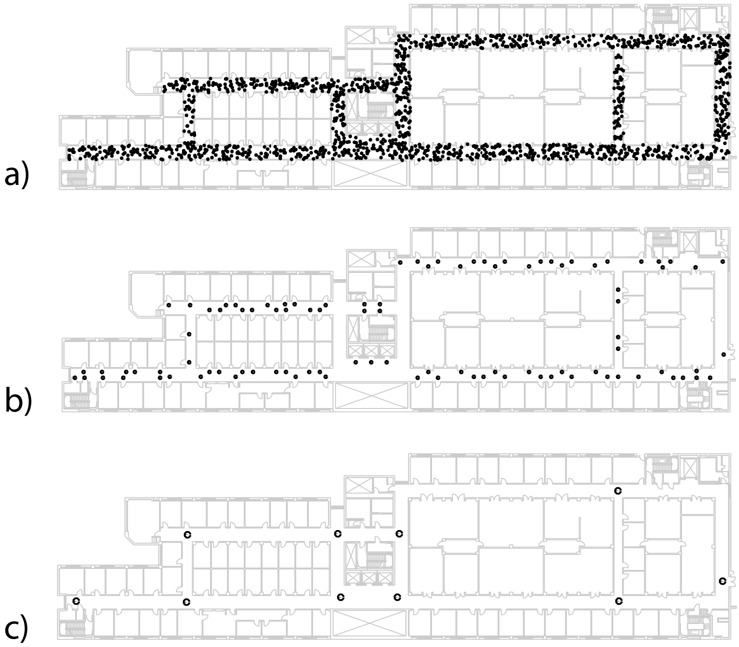


Figure 1: The required number of instantiated hypotheses for (a) particle filter localization, (b) multi-hypothesis Kalman filtering, and (c) our hybrid filtering approach.

our previous work was only designed to update localization hypotheses when the robot arrived at a new topological vertex after an edge traversal instead of updating continuously while the robot obtains sensor measurements. Thus, the algorithm that we presented earlier was essentially forced to pause and wait until the robot completed an edge traversal before it could apply the sensor observations that were obtained along the GVG edge. This delay meant that, unlike our new approach, the robot could not localize as soon as should have been possible given the sensory information. This initial work was also not generalized for SLAM, could not recover from pruning errors, and improperly relied on a heuristic to detect any occurrence of the kidnapped robot problem.

2.3 Metric SLAM

The seminal work by Smith et al. (1990) introduced stochastic mapping to the robotics community, which led to an intense research effort in SLAM. To this day, the field is continually refining and improving SLAM algorithms to achieve maximal accuracy while minimizing computation. One of the most popular SLAM approaches uses an extended Kalman filter (EKF) to jointly estimate the probability distribution over robot poses and feature positions. The typical measurements used for metric mapping are either range/bearing measurements (Dissanayake et al. 2001; Durran-Whyte and Bailey 2006; Bailey and Durran-Whyte 2006) or bearing-only measurements (Montiel et al. 2006; Civera et al. 2007; Tully et al. 2008, 2010).

Another metric SLAM algorithm is FastSLAM (Montemerlo et al. 2002, 2003), which factors the SLAM problem into a feature estimation problem that can be implemented with a bank of independent EKFs and a localization problem that can be estimated with a particle filter. This method is efficient due to the independence of feature estimation but can have issues with particle diversity and closing large loops.

A newer innovation is GraphSLAM (Thrun and Montemerlo 2006), where the SLAM posterior is formulated as a graph network that includes historical poses of the robot throughout an experiment. The graph is reduced using a variable elimination method and solved using conventional optimization techniques. One advantage of GraphSLAM is that features are independently estimated, allowing for efficient mapping in environments with many features. The downside of GraphSLAM is the complexity of performing global data association to close large loops in the presence of positional drift.

2.4 Topological and Hybrid SLAM

Topological maps are also commonly used for SLAM due to their efficiency. But many existing topological mapping methods commit to a loop closure after observing a similar fingerprint or structural characteristic to that of a vertex already in the map. For example, Choset and Nagatani (2001) use the degree and equidistance measures at the vertices of a Voronoi diagram to determine if the robot has returned to a previously visited vertex. Likewise, Tomatis et al. (2002) observe when the probability distribution over robot positions splits into two peaks, suggesting a loop. In both cases, the algorithm is susceptible to the perceptual aliasing problem, in which many locations are ambiguous.

Recently, multi-hypothesis techniques have emerged to better estimate topological hypotheses with ambiguous loop closure: Ranganathan et al. (2006); Ranganathan and Dellaert (2011) perform topological SLAM with a Bayesian inference method that involves sampling to generate map hypotheses, Savelli and Kuipers (2004) test a tree of topological hypotheses for planarity to ensure that the topological maps are consistent, and De et al. (2008) demonstrate an EM technique for generating plausible and efficient topological maps that can explain the data (albeit limited to one-cycle and two cycle graphs). Also, in (Dudek et al. 1993; Marinakis and Dudek 2010), the authors use hypothesis trees to generate topological hypotheses but do not evaluate the posterior probability of each hypothesis.

In (Tully et al. 2009), we previously introduced a multi-hypothesis topological loop closing method with a recursive Bayesian framework for estimating SLAM hypotheses. This work introduced a tree expansion algorithm and a Bayesian method to evaluate the posterior probability of each hypothesis. Unlike the work presented in this paper, though, our previous work in (Tully et al. 2009) was not yet generalized for localization, did not concern metric estimation,

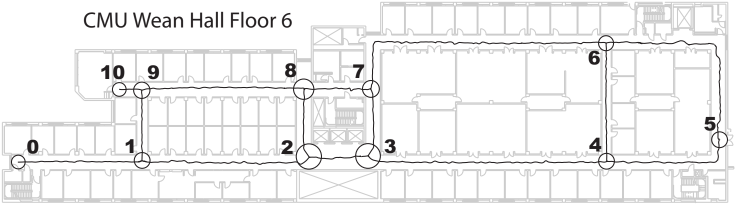


Figure 2: The high-level topological graph for the *hierarchical atlas* is the generalized Voronoi graph (GVG), whose vertices and edges are shown here in an example map.

and was not robust to unmodeled disturbances.

3 Constructing a Hypothesis Forest

In this paper, we are introducing a filtering approach that, when implemented, can be generalized for performing either global localization or SLAM in hybrid metric/topological maps. This is because our formulation computes a *hypothesis forest*, which is a set of multiple trees whose nodes represent possible topological hypotheses. We must maintain multiple trees for this estimation problem because it accounts for uncertainty in the robot's initial state (thus multiple root nodes in the forest) while at the same time allowing for multiple hypotheses to be spawned from a single hypothesis due to ambiguous loop closure.

3.1 Hybrid Metric/Topological Map

Before we describe our recursive algorithm for constructing a hypothesis forest for topological estimation, it is important to define the hybrid map that we are using for this work. Our hybrid map is based on the *hierarchical atlas* (Lisien et al. 2005; Tully et al. 2007), which uses a topology to decompose the space into smaller subspaces in which local metric maps are defined that are tractable by conventional filtering methods. This allows for a high resolution representation of the free space while maintaining low computation and storage costs for localization and mapping.

The topology used by the hierarchical atlas is based on the generalized Voronoi graph (GVG) (Choset and Burdick 1995), whose edges are the set of points equidistant to two obstacles and whose vertices are equidistant to three or more obstacles, see Fig. 2. A principle benefit is that the vertices have a definite location in the free space and the edges define obstacle-free paths between neighboring vertices, *i.e.*, the topology is both abstract and embedded in the free space, and as such can be traversed using sensor-based control. Additionally, a key property of our hybrid map is that metric submaps (e.g., Fig. 3) are inherently loop-free. This means that loops only exist at the topological level[1],

[1] The cardinality of the first fundamental group for both the free space and the topological

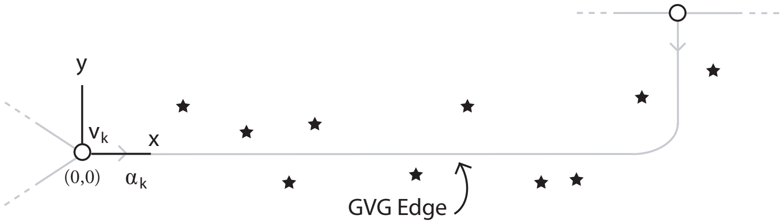


Figure 3: This is a typical submap for the *hierarchical atlas*. The destination vertex and the edge index are used to encode the submap in the robot state. Point features throughout the submap add metric detail to the map.

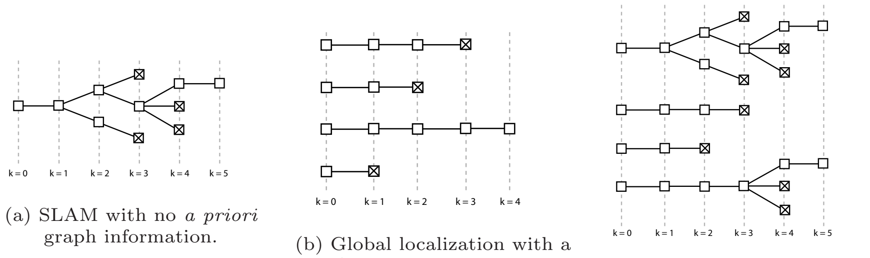
and thus the evaluation of loop closure hypotheses need only be performed on the high level topological graph.

The state of the system for this type of hybrid map can be encoded as $[X_k, S_k, M_k, G_k]$, where X_k represents the relative metric pose of the robot in its local submap, S_k represents the set of local metric maps, M_k designates the topological state of the robot (i.e., in which submap the robot currently resides), and G_k represents the topological graph and its connectivity. While the explicit definition of S_k can vary depending on the metric submapping technique that is adopted, the other terms can be defined as follows,

$$X_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} \quad M_k = \begin{bmatrix} v_k \\ \alpha_k \end{bmatrix} \quad G_k = \begin{bmatrix} N_k \\ L_k \end{bmatrix},$$

where v_k is the GVG vertex the robot departed when entering submap M_k and α_k is the index of the edge the robot is traversing (relative to a reference edge of vertex v_k). The origin of submap M_k coincides with the location of vertex v_k, and the coordinate frame of the submap is defined such that the departing angle of the Voronoi edge aligns with the x-axis. The robot's metric pose in the submap is denoted by x_k, y_k, and θ_k, which are all in the local coordinate frame. The term N_k defines the number of vertices in the topological graph and L_k represents a set of circular neighbor lists (one list per vertex) that define the neighboring vertices for each vertex in the topology, as in (Vijayan and Wigderson 1982). For example, $L_k(v_k)$ stores the vertices in the graph that are neighbors of vertex v_k in the order they occur (counter-clockwise from the first mapped edge). An element of the neighbor list $L_k(v_k, j)$ represents the neighboring vertex of v_k along the j-th edge. Fig. 5 shows two edge-ordered graphs with similar topologies but different edge-orderings. The following variable list will serve as a reference when we formulate our hybrid filtering scheme.

map embedded in the free space is the same.



(a) SLAM with no *a priori* graph information.

(b) Global localization with a fully explored graph.

(c) Partial map localization and SLAM

Figure 4: This figure shows three types of hypothesis forests that could arise using our filtering algorithm. In (a), without any a priori information, we initialize one hypothesis for SLAM and due to multihypothesis loop closure, a tree expands. In (b), with a known graph, we must filter on multiple starting positions. In (c), having a partially explored map and uncertainty in the starting position on the graph requires a hypothesis forest.

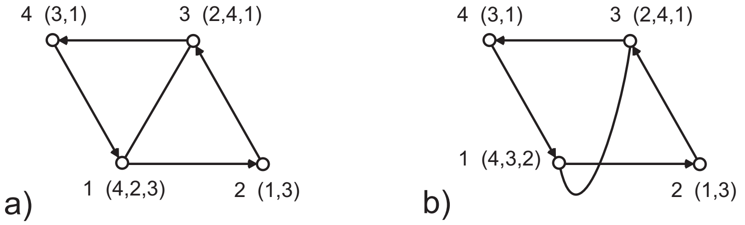


Figure 5: This is an example of two edge-ordered graphs. The values next to each vertex are the vertex indices, while the parenthetical sequences represent the edge-ordered neighbor lists associated to each vertex. The first mapped edge for each vertex is shown with an arrow.

X_k	$\rightarrow$	Local Metric Pose
S_k	$\rightarrow$	Set of Local Metric Maps
M_k	$\rightarrow$	Discrete Robot State
v_k	$\rightarrow$	Departing Vertex Index
α_k	$\rightarrow$	Departing Edge Index
x_k	$\rightarrow$	Relative Metric x Coordinate
y_k	$\rightarrow$	Relative Metric y Coordinate
θ_k	$\rightarrow$	Relative Metric Heading
N_k	$\rightarrow$	Number of Topological Vertices
L_k	$\rightarrow$	Neighbor Lists

For this work, we also consider partially explored maps. In this case, a neighbor list in the graph can contain one or more entries marked as *unexplored*, which means, according to that hypothesis, the robot has not yet traversed the edge associated with that entry of the neighbor list.

3.2 Incremental Construction of a Hypothesis Forest

The first goal is to incrementally build a set of hypotheses that can completely reproduce the possible topological graph/state pairs at every time step k given the topological transitions performed by the robot. We will designate (M_k^h, G_k^h) to be a possible graph/state pair hypothesis at time step k that is indexed by h. To recursively construct the set of hypotheses, we maintain a hypothesis forest where each level of the forest represents a different time step in the experiment. Therefore, a level of the forest is indexed with time step k. The forest structure we maintain is comprised of multiple hypothesis trees, each of which is similar to that in (Dudek et al. 1993; Marinakis and Dudek 2010; Remolina and Kuipers 2004).

The robot begins an experiment at one vertex in the map. The robot has no other information except for the degree of that vertex, δ_0, which equals the number of edges emanating from the vertex. If the topological graph is known *a priori* or is partially explored, then we must initialize multiple root nodes in our hypothesis forest for each possible starting submap state M_k^h in the existing a priori known graph hypothesis G_k^h. For the case of a partially explored map, specifically, we must also add an additional root node that hypothesizes that the robot is starting the experiment in a topological state that is not represented by the partially explored graph.

If the topology is completely unexplored, we initialize a single root node of the hypothesis forest as follows: $h = 0$, $k = 0$, $N_k^h = 1$, $v_k^h = 0$, and $\alpha_k^h = 0$. The circular list for the first vertex, $L_k^h(0)$, is initialized as a list of length δ_0 for which each entry is labeled as *unexplored*. All hypotheses in this single-tree forest are ultimately spawned from this initial root hypothesis.

In Fig. 4, we show three possible examples for how a hypothesis forest can be initialized and expanded as an experiment progresses. In Fig. 4-(a), the robot begins without any *a priori* information and thus there is a single root node initialized in the forest that can spawn multiple hypotheses over time due to multi-hypothesis loop-closure. This equates to pure topological SLAM. In Fig. 4-(b), the robot is given a completely explored map and thus initialization is ambiguous (causing multiple root nodes) but loop-closure no longer spawns multiple hypotheses because all loops are known *a priori*. This equates to a pure global localization problem. Fig. 4-(c) is an example that demonstrates a typical hypothesis forest for a robot performing SLAM while also localizing globally in a partially explored map. This example applies to a robot that is continuing a previous SLAM experiment on a partially explored graph but without a known initial state.

The robot is continuously moving from vertex to vertex in the topology. When the robot arrives at a new vertex at time step k, the robot chooses a motion input u_k in order to transition to another vertex. The motion input is a relative offset from the arrival edge into the vertex β_k, and produces the following departure edge α_k,

$$\alpha_k = (\beta_k + u_k) \bmod \delta_k, \tag{1}$$

where β_k is again the arrival edge when the robot arrived at the new topological vertex. After departing along edge α_k, the robot drives to a new vertex along the GVG edge of the environment and arrives at another vertex in the graph.

We assume that the robot correctly performs the motion input u_k at each time step and therefore leaves the previous vertex via the appropriate departure edge. This has been an accurate assumption experimentally, most likely due to the robust sensor-based control of the robot we use for experiments. Nevertheless, we provide a discussion of how to recover from the failure of this assumption in Sec. 5.

Algorithm 1 Expanding the Hypothesis Tree

```
for all $h \in H_{k-1}$ do
    $[v^h_{k-1},\ \alpha^h_{k-1},\ N^h_{k-1},\ L^h_{k-1}] \leftarrow$ LoadHypothesis($h$)
    if $L^h_{k-1}(v^h_{k-1},\ \alpha^h_{k-1}) = unexplored$ then
        $h' \leftarrow$ CreateChildHypothesis($h$)
        $L^{h'}_k = L^h_{k-1}$
        $v^{h'}_k = N^h_{k-1}$
        $L^{h'}_k(v^{h'}_k,\ 0) = v^h_{k-1}$
        for $e = 1$ to $\delta_k - 1$ do
            $L^{h'}_k(v^{h'}_k,\ e) = unexplored$
        end for
        $L^{h'}_k(v^h_{k-1},\ \alpha^h_{k-1}) = v^{h'}_k$
        $\alpha^{h'}_k = (0 + u_k) \bmod \delta_k$
        AddChild($h', v^{h'}_k,\ 0,\ N^h_{k-1} + 1,\ L^{h'}_k$)
        for $v = 0$ to $N^h_{k-1} - 1$ with $v \neq v^h_{k-1}$ do
            for all $e$ $s.t.$ $L^h_{k-1}(v, e) = unexplored$ do
                $h' \leftarrow$ CreateChildHypothesis($h$)
                $L^{h'}_k = L^h_{k-1}$
                $L^{h'}_k(v, e) = v^h_{k-1}$
                $L^{h'}_k(v^h_{k-1}, \alpha^h_{k-1}) = v$
                $\alpha^{h'}_k = (e + u_k) \bmod \delta_k$
                AddChild($h',\ v,\ \alpha^{h'}_k,\ N^h_{k-1},\ L^{h'}_k$)
            end for
        end for
    else
        $h' \leftarrow$ CreateChildHypothesis($h$)
        $v^{h'}_k = L^h_{k-1}(v^h_{k-1},\ \alpha^h_{k-1})$
        $\beta^{h'}_k = e$ $s.t.$ $L^h_{k-1}(v^{h'}_k,\ e) = v^h_{k-1}$
        $\alpha^{h'}_k = (\beta^{h'}_k + u_k) \bmod \delta_k$
        AddChild($h',\ v^{h'}_k,\ \alpha^{h'}_k, N^h_{k-1},\ L^h_{k-1}$)
    end if
end for
```

When the robot arrives at a new vertex and chooses a new motion input u_k,

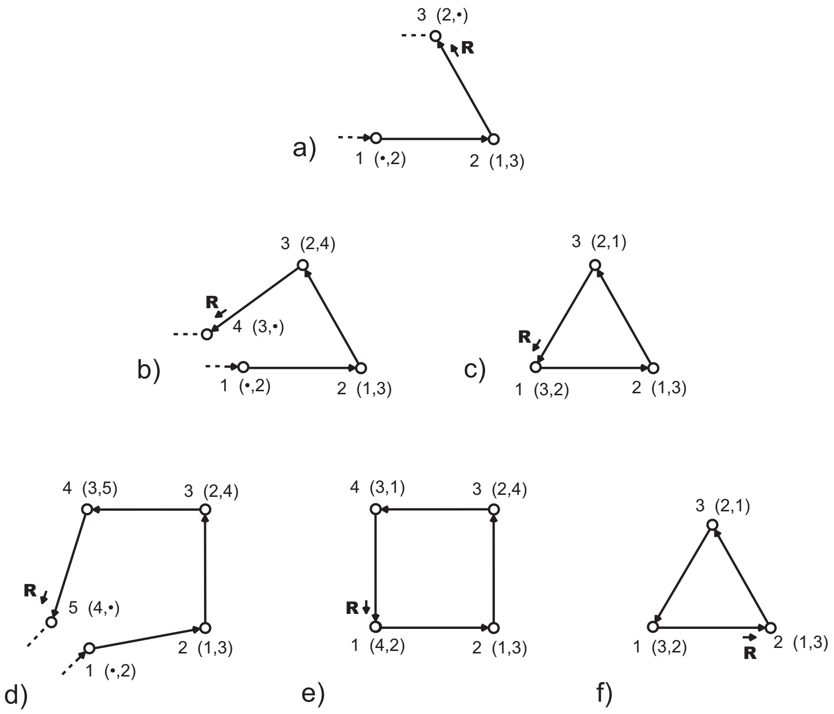


Figure 6: This is an example of expanding the hypothesis tree due to robot motion. Hypothesis (a) spawns (b) and (c) after one edge traversal. After another edge traversal, hypothesis (b) spawns (d) and (e) while hypothesis (c) spawns only (f). The location of **R** in the figure marks the robot's state.

we must update the hypothesis forest by expanding all of the H_{k-1} leaf nodes of the forest (the leaf nodes being the set of hypotheses at time step $k-1$). The new hypotheses that are spawned become the new leaf nodes of the forest for time step k. The algorithm for expanding the tree is outlined in Alg. 1.

Alg. 1 expands all H_{k-1} leaf nodes of the hypothesis forest in the following way. If, according to a hypothesis h, $L^h_{k-1}(v^h_{k-1}, \alpha^h_k)$ (which is the neighbor of the previous vertex v^h_{k-1} that is associated to the departing edge α^h_k) is *not unexplored*, then we copy the hypothesis to a single child hypothesis but move the robot's state to the new vertex and update the departing edge. If $L^h_{k-1}(v^h_{k-1}, \alpha^h_k)$ is *unexplored*, then the algorithm considers several possibilities that would agree with the hypothesis: the first possibility is that the robot has traversed the unexplored edge and has arrived at a new vertex (one hypothesis is spawned for this possibility). Additionally, the algorithm considers that a loop is closed and the robot arrives at a previously visited vertex via one of its unexplored edges. One hypothesis is spawned for each unexplored edge in the graph except for the current departure edge.

Fig. 6 demonstrates the expansion of the hypothesis forest. In this example, an edge traversal causes (a) to spawn hypotheses (b) and (c). This accounts for the possibility of either visiting a new vertex or closing a loop with vertex 1. After a second edge traversal, hypothesis (b) spawns hypotheses (d) and (e) for the same reasoning. Hypothesis (c), though, is a complete graph with no unexplored edges, and therefore spawns just one hypothesis, (f), in which the state has moved according to the robot motion.

4 A Unified Bayesian Framework for Global Localization and SLAM

In order to solve the problem of topological global localization and SLAM, we must determine which hypotheses among the leaf nodes of the hypothesis forest are likely to represent the true state and the true graph. In addition, we need to compute the metric pose of the robot within the topological submap associated with each hypothesis and we also need to estimate the local maps for each metric submap in the topology. To do this, we compute the posterior probability of each hypothesis given a sequence of sensor measurements and robot motion inputs: the hypothesis that better fits the sensor data will produce a higher probability measure and is therefore more likely to represent the true state and graph.

4.1 Bayesian Filtering Framework

The overall filtering goal is to estimate a probability distribution over the hybrid state of the robot $[X_k, S_k, M_k, G_k]$, as follows,

$$p(X_k, S_k, M_k, G_k | u^k, z^k, \nu^k, \zeta^k).$$

where u^k is the sequence of metric motion inputs obtained during an experiment, z^k is the sequence of metric measurements, ν^k is the sequence of topological motion inputs, and ζ^k is the set of topological measurements sensed during submap transitions. The following variable list is provided as a reference.

u^k	→	Metric Motion Sequence
z^k	→	Metric Measurement Sequence
ν^k	→	Topological Motion Sequence
ζ^k	→	Topological Measurement Sequence

The high level posterior probability estimation problem can be factored, as in (Blanco et al. 2008), using the definition of conditional probability,

$$\begin{aligned} &p(X_k, S_k, M_k, G_k | u^k, z^k, \nu^k, \zeta^k) \\ &\quad = \underbrace{p(X_k, S_k | M_k, G_k, u^k, z^k, \nu^k, \zeta^k)}_{\text{Relative Metric Filtering}} \underbrace{p(M_k, G_k | u^k, z^k, \nu^k, \zeta^k)}_{\text{Topological Filtering}} . \end{aligned}$$

The posterior is now separated into a metric filtering problem and a discrete topological filtering problem. The next step we can take is to separate the topological filtering problem into a likelihood term and a prior using Bayes rule,

$$\begin{aligned} & p(M_k, G_k | u^k, z^k, \nu^k, \zeta^k) \\ & \quad = \eta \, \underbrace{p(z^k, \zeta^k | M_k, G_k, u^k, \nu^k)}_{\text{Likelihood Term}} \underbrace{p(M_k, G_k | u^k, \nu^k)}_{\text{Prior Term}} . \end{aligned} \tag{2}$$

The scalar value η in Eq. 2 is used for normalization over the probability distribution.

Our approach is to estimate a discrete probability distribution over possible topological hypotheses (that are by design generated by our hypothesis forest). Thus we are estimating a posterior probability for each hypothesis $[M_k^h, G_k^h]$. We are also estimating X_k and S_k with a bank of continuous filters (one continuous filter per topological hypothesis $[M_k^h, G_k^h]$).

4.2 Relative Metric Filtering

To solve the metric filtering problem, we want to estimate

$$p(X_k, S_k | M_k^h, G_k^h, u^k, z^k, \nu^k, \zeta^k)$$

for each topological hypothesis $[M_k^h, G_k^h]$ that is generated by the hypothesis forest described in Sec. 3.

When the robot transitions to a new submap, we know that the robot is positioned on the origin of the new local submap. This is because the robot hones into a GVG vertex upon transitioning to a new GVG edge. Thus, upon transitioning to a new submap, we can initialize a Kalman estimate for the pose of the robot as follows,

$$\begin{aligned} \hat{X}_k = & \quad 0_{3\times 1} & \text{for} \quad & u_k = \emptyset, \nu_k \neq \emptyset \\ P_k = & \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{bmatrix} & \text{for} \quad & u_k = \emptyset, \nu_k \neq \emptyset \end{aligned}$$

where the diagonal elements are nonzero variances to define the initial uncertainty in the robot pose in relation to the submap origin. We initialize a bank of Kalman filters this way for each hypothesis in the topological hypothesis forest.

Each filter in the bank of Kalman filters that estimate X_k will also include an estimate of the local metric submap $S_k^{M_k}$. Defined this way, the set $S_k^{M_k}$ for all M_k defines the term S_k. Upon transitioning to a new submap, the local submap estimate $S_k^{M_K}$ is either 1) retrieved from a prior estimate if the topological edge has been explored previously, or 2) initialized as an empty map that will be created online using a local metric SLAM approach.

In general, if the submap M_k in graph G_k is already explored, the estimation of the continuous distribution $p(X_k, S_k | M_k, G_k, u^k, z^k, \nu^k, \zeta^k)$ is more closely related to using a Kalman filter (EKF) to track the robot pose given exteroceptive sensor measurements z^k to features in the previously mapped submap $S_k^{M_K}$.

If the submap M_k in graph G_k has not yet been explored according to a hypothesis, then the robot does not have the metric submap with which to localize. In this case, the robot will perform local metric SLAM along the GVG edge of the topology to create a new feature map $S_k^{M_k}$ that can be attached to the topological hypothesis for use with localization when traversing the edge again. The algorithm that is used to build the metric feature map within the topological edge is not important (the algorithm generalizes to any local metric SLAM algorithm).

4.3 Topological Likelihood Estimation

In the previous section, we describe a metric localization and mapping framework that tracks the pose of the robot in each topological hypothesis with a bank of EKF's. In a sense, our formulation is related to performing multi-hypothesis Kalman filtering for SLAM and localization. The task that remains is determining which of these metric Kalman filters is tracking the true pose and estimating the true graph. To do this, we need to estimate the likelihood term from Eq. 2, for each hypothesis h,

$$p(z^k, \zeta^k | M_k^h, G_k^h, u^k, \nu^k), \tag{3}$$

To reduce storage and computation, this likelihood term can be computed recursively for a hypothesis h' given the likelihood of the parent hypothesis h in the hypothesis forest, i.e.,

$$\begin{aligned} & p(z^k, \zeta^k | M_k^{h'}, G_k^{h'}, u^k, \nu^k) \\ & = p(z_k, \zeta_k | z^{k-1}, \zeta^{k-1}, M_k^{h'}, G_k^{h'}, u^k, \nu^k)\ p(z^{k-1}, \zeta^{k-1} | M_k^{h'}, G_k^{h'}, u^k, \nu^k) \\ & = p(z_k, \zeta_k | z^{k-1}, \zeta^{k-1}, M_k^{h'}, G_k^{h'}, u^k, \nu^k)\, p(z^{k-1}, \zeta^{k-1} | M_{k-1}^h, G_{k-1}^h, u^{k-1}, \nu^{k-1}) \end{aligned} \tag{4}$$

In Eq. 4, the likelihood function has been split into two terms using the definition of conditional probability: the second term can be viewed as a prior on the likelihood function for the recursion, while the first term represents the update to the likelihood after receiving a new measurement at time step k. This means that we can take the likelihood of a parent hypothesis in the hypothesis forest, expand the children of the hypothesis using the motion of the robot, and then recursively update the likelihood by multiplying by an update term $p(z_k, \zeta_k | z^{k-1}, \zeta^{k-1}, M_k^{h'}, G_k^{h'}, u^k, \nu^k)$.

The likelihood update that we use experimentally can be written as follows,

$$\begin{aligned} & p(z_k | z^{k-1}, \zeta^{k-1}, M_k, G_k, u^k, \nu^k) \propto \\ & \qquad e^{-\frac{1}{2}(z_k - h(\hat{X}_k))^T R^{-1} (z_k - h(\hat{X}_k))}, \end{aligned} \tag{5}$$

which we can generalize for the topological measurement ζ_k as well with an independent multiplication by an equivalent term. The matrix R in this formulation is the measurement covariance matrix. The term z_k is the metric measurement obtained at time step k and the term $h(\hat{X}_k)$ is the expected measurement given the pose of the robot and the metric submap. This type of update can be used for, say, a range or bearing measurement to a metric feature in the topological submap. We note that this likelihood update process is performed for each submap simultaneously as new measurements are obtained. This means that at the beginning of an experiment, when the robot arrives at a new vertex, the likelihoods of many hypotheses will be high. Eventually, though, as new information is incorporated using this filtering update, the likelihood of the wrong hypotheses will tend to zero.

When applying the likelihood update, it may not seem apparent that the robot will know how to compute the expected measurement given the local metric map associated to the topological edge. If this were the case, the robot would not be able to compute $h(\hat{X}_k)$ in Eq. 5. This is because, at first glance, the equations do not seem to account for the robot's orientation difference between a past visit to this submap and the robot's current visit. But this is actually not the case. When the local feature-based submap is visited for the first time, the robot creates a map that is relative to the departing angle of the robot (i.e. the x-axis aligns with the departing angle of the robot, as seen in Fig. 3). When the robot visits this submap again, and as long as it zeros its heading when departing the GVG vertex, all of the landmarks will be relative to the robot's new coordinate frame.

Eq. 5 does not account for when a robot receives a false-positive measurement. This means that a measurement was obtained that could not be associated to anything in the map. This is depicted in Fig. 7-(c). In this case, we assign a maximum likelihood penalty to the likelihood function. The intuition is that we would like to penalize this hypothesis because in this situation, the submap hypothesis says that there should not have been a measurement when there was, thus the hypothesis itself may not be correct.

Another case that Eq. 5 does not account for is when there is a landmark in the local metric map nearby the robot that should have produced a sensory measurement, but for some reason the robot did not observe any measurements. In this case, we choose not to modify the likelihood at all (except for possible effects from normalization). This is because feature detection, in general, is not always a dependable measurement and thus it would not make sense to penalize this situation.

In Fig. 7, we show three examples of what can occur during a robot's traversal of a topological submap. In (a), the robot measures a metric feature, and because the measurement aligns well with an associated feature in the stored map, the likelihood update in Eq. 5 will favor this hypothesis. In (b), we show a measurement not aligning well with the stored map. The likelihood update will penalize this hypothesis accordingly. In (c), we show a situation in which the measurement cannot be associated with a feature in the map. This suggests that the robot is actually in a different submap than the one hypothesized, and thus this node in the hypothesis forest will be given a maximum penalty for the likelihood term. In (d), we give an example of the the robot passing by a landmark (according to the hypothesis) but no measurements are received by the onboard sensing. In this case, we do not penalize the likelihood term.

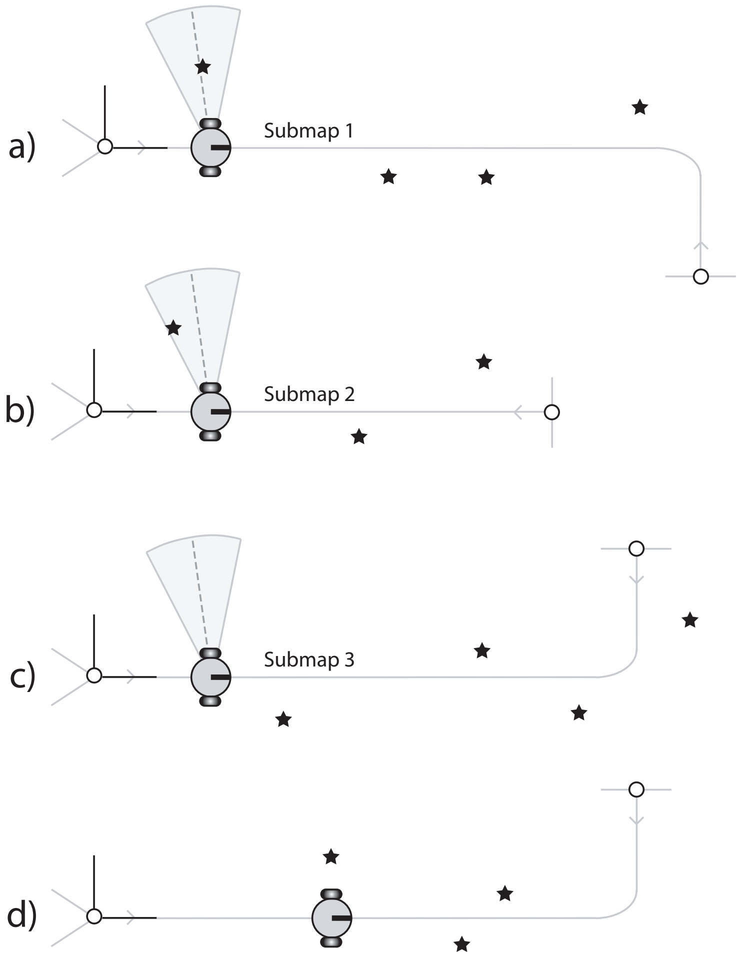


Figure 7: In (a), there is good alignment between the measurement and the expected landmark location, which will cause increase the likelihood. In (b), a measurement is associated but does not align well, which will negatively impact the likelihood. In (c), a measurement that cannot be associated to any known landmark will apply the maximum likelihood penalty. Lastly, in (d), missing a nearby measurement will not be penalized.

4.4 Topological Prior Estimation

Neglected thus far in our discussion is the prior term $p(M_k, G_k | u^k, \nu^k)$ which, when combined with the recursive likelihood computation and the metric localization and SLAM, completes the terms required to jointly estimate the probability distribution over the hybrid state space $[X_k, S_k, M_k, G_k]$. We compute this prior for each hypothesis h. This term can be factored as follows,

$$p(M_k^h, G_k^h | u^k, \nu^k) = \underbrace{p(M_k^h | G_k^h, u^k, \nu^k)}_{\text{State Prior}} \underbrace{p(G_k^h | u^k, \nu^k)}_{\text{Graph Prior}} . \tag{6}$$

The term $p(M_k^h | G_k^h, u^k, \nu^k)$ is a prior on the submap the robot is in (given a graph), which essentially decides the weighting of a hypothesis upon initialization: for global localization, this distribution would be uniform over many submaps in a single graph; for SLAM, there would be one initialized hypothesis and this hypothesis would have a prior of 1.0.

The term $p(G_k^h | u^k, \nu^k)$ in Eq. 6 is a graph prior that weights the hypothesized graphs according to their a priori likelihood. In other words, this term represents, without any sensor information, the probability that the robot happens to be placed in an environment with a topology G_k^h. What should this distribution be? We believe there is actually no correct way to define this distribution.

But we can do better than a uniform distribution. Consider the following situation: a robot is circling a triangle topology, as in Fig. 8 (a), with three different edges. Over time, it would appear that a sensor measurement is repeated every third time step because the robot is traversing the same three edges over and over. The triangle, as the correct map, would fit the sensor data very well. On the other hand, the topology in Fig. 8 (b) would also fit well for the same measurement sequence due to perceptual aliasing. Which topology should be preferred? In some sense, topology (b) is over-fitting the data. Like Occam's razor, the graph that explains the sensor data with the simplest solution should be chosen. We use the following distribution for experiments,

$$p(G_k^h | u^k, \nu^k) \propto \exp\left(-N_k^h \log k\right) ,$$

where N_k^h is the number of topological vertices in the graph according to hypothesis h.

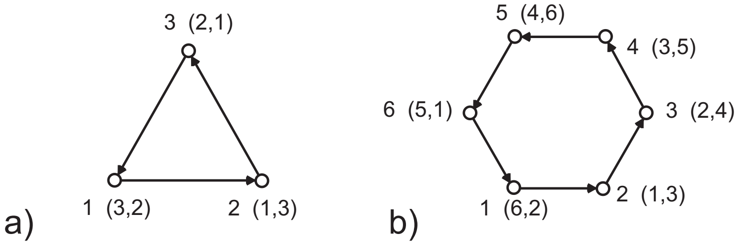


Figure 8: This is an example of two different topologies that can result in a situation of perceptual aliasing. Both topologies fit the sensor data well.

When two hypotheses have a similar likelihood, this prior will give preference to the smaller graph. This makes sense, because we would like to prevent over-fitting the data. It turns out that this formulation is equivalent to using the Bayesian information criterion (Schwarz 1978) for model selection. The Akaike information criterion (Akaike 1974) is related and is used in (De et al. 2008) with considerable success for a limited class of topologies.

By combining in Eq. 2 the prior developed here with the likelihood function of Eq. 4, we are effectively trying to capture the perfect balance between small concise maps that would make sense for a structured environment and large intricate maps that better fit the data.

5 Pruning Hypotheses

The forest expansion algorithm described in Sec. 3 exhaustively considers all possible loop closures during an experiment. Therefore, even for a small map, the number of leaf hypotheses in the forest can grow very quickly (even to a size that is not computationally feasible). To keep the forest size bounded, we apply a series of pruning tests to the leaf hypotheses at each time step. In addition to a pruning step, we introduce a new method called the garbage-collector hypothesis that is a catch-all for pruned hypotheses and that can help the algorithm recover from unmodeled disturbances.

5.1 Topological Hypothesis Pruning

Our pruning stage is crucial in the success of the algorithm and allows for the processing of large and ambiguous maps. We apply only conservative rules to prune hypotheses in order to reduce the chance of eliminating the hypothesis that represents the true map/state.

5.1.1 Degree Test

In Alg. 1, when $L_{k-1}(v_{k-1},\ \beta_k) = \mathit{unexplored}$, the hypothesis tree adds a child hypothesis for every possible loop closure that connects the previous vertex with another vertex v that also has a dangling unexplored edge. If the measured degree δ_k is unequal to the degree of vertex v, then that child hypothesis is immediately discarded. This test involves no risk of eliminating the true hypothesis as long as the degree is measured accurately.

5.1.2 Likelihood Update Test

When updating the likelihood for a new hypothesis recursively via Eq. 4, we observe whether $p(z_k, \zeta_k | z^{k-1}, \zeta^{k-1}, M_k^h, G_k^h, u^k, \nu^k)$ exceeds a 4-sigma error bound. If true, this would imply that the new measurements do not agree with the measurements already associated to the corresponding edge/vertex and are therefore relative outliers in the data. This hints at an incorrect loop closure and thus the hypothesis is pruned. This test has an extremely small but nevertheless non-zero chance of eliminating the true hypothesis.

5.1.3 Planarity Test

As in (Savelli and Kuipers 2004), we use a strict test to eliminate hypotheses whose topological graphs are not planar. For more intricate graphs, and those with many loops, the addition of this test can often prune a large number of hypotheses without the risk of discarding the correct hypothesis. We use the algorithm in (Vijayan and Wigderson 1982) to test planarity because it is specifically designed for edge-ordered graphs. The benefit is that we can prune even more graphs, e.g., those that are planar in a conventional sense but not planar when considering edge-ordering. An example is the graph in Fig. 5 (b).

5.1.4 Posterior Probability Test

Our least conservative pruning rule is to eliminate any hypothesis whose posterior probability drops below a threshold. This implies that the hypothesis is either a very poor fit to the sensor data or is dominated by a hypothesis that can explain the sequence of measurements just as well with a smaller map representation. A hypothesis is pruned when the posterior of a hypothesis $p(M_k^h, G_k^h | z^k, u^k, \nu^k, \zeta^k)$ exceeds a 4-sigma error.

5.2 Garbage-Collector Hypothesis

Unfortunately, by choosing to prune hypotheses that have a nonzero probability, we are sacrificing the Bayes optimality of our hybrid filtering approach. To regain this optimality, we collect the leftover probability from pruned hypotheses into a *garbage-collector hypothesis.* The garbage-collector hypothesis is a catch-all for hypotheses that have been eliminated. More specifically, the garbage-collector hypothesis estimates the probability that the true topological location no longer exists in our current set of hypotheses and must have been wrongly pruned from the hypothesis list. For this reason, this extra hypothesis is a safety net for our filter that can detect when an unmodeled disturbance has led the estimation process astray. We provide a custom Bayesian filter framework to continually update this garbage-collector hypotheses with sensor measurements obtained by the robot. Our solution to this problem is related to the work by Jensfelt and Kristensen (2001).

In this formulation, we choose to only update the garbage collector hypothesis when a topological measurement is obtained. The metric feature measurements within the submap do not play a part in the determination of the garbage collector probability.

The posterior probability of the garbage-collector hypothesis C_k can be written as follows,

$$p(C_k | u^k, z^k, \nu^k, \zeta^k)$$

where u^k, z^k, ν^k, and ζ^k are the motion and measurement input sequences, as before. $p(C_k | u^k, z^k, \nu^k, \zeta^k)$ starts at zero, for at the beginning of an experiment, without having pruned any hypotheses, we know that the true state is still completely covered by the hypothesis set and so the likelihood of the robot being somewhere else is zero.

During our test for pruning, which we apply to each submap M_k, we check if the probability associated with that submap has dropped below a threshold. If it has, that probability is added to the garbage-collector hypothesis, as follows,

$$\begin{aligned} &p(C_k|u^k, z^k, \nu^k, \zeta^k) = \\ &\qquad p(C_k|u^k, z^k, \nu^k, \zeta^k) + p(M_k^h, G_k^h|u^k, z^k, \nu^k, \zeta^k) \\ &\qquad \text{for all } \ [M_k^h, G_k^h] \ \text{ s.t. } \ p(M_k^h, G_k^h|u^k, z^k, \nu^k, \zeta^k) < \gamma. \end{aligned} \tag{7}$$

By keeping the portion of total probability that would have been pruned out, the normalization step will no longer inflate the remaining hypotheses to accommodate for the lost probability. We note that the garbage-collector hypothesis is included when normalizing.

When we perform the measurement update for the likelihood term for all of the topological hypotheses, we also update the probability for the garbage-collector hypothesis. The measurement update is,

$$\begin{aligned} &p(C_k|u^k, z^k, \nu^k, \zeta^k) \propto \\ &\qquad p(\zeta_k|C_k, u^k, z^{k-1}, \nu^k, \zeta^{k-1}) p(C_k|u^{k-1}, z^{k-1}, \nu^{k-1}, \zeta^{k-1}) \end{aligned}$$

which uses Bayes law to split the posterior into a likelihood term and a prior on the posterior. Some of the exponents have been decremented on the RHS in order to depict this update in a recursive form. This is allowed because the extra motion terms u_k and ν_k that were dropped from the prior term on the RHS do not add information to the garbage-collector hypothesis. Only measurements can help resolve whether the state is somehow in one of the pruned hypotheses. Also, the most recent metric measurement z_k is dropped from the RHS because in this formulation, we choose to only update the garbage-collector with measurements at the vertex, in which case z_k is undefined.

The measurement likelihood can be computed by adding a hidden variable using the law of total probability,

$$\begin{aligned} &p(\zeta_k|C_k, u^k, z^{k-1}, \nu^k, \zeta^{k-1}) \\ &\qquad = \int_{-\infty}^{\infty} p(\zeta_k|w_k, C_k, u^k, z^{k-1}, \nu^k, \zeta^{k-1}) \ \times \\ &\qquad\qquad p(w_k|C_k, u^k, z^{k-1}, \nu^k, \zeta^{k-1}) \ dw_k \end{aligned} \tag{8}$$

The hidden variable w_k is the true measurement (for example an equidistance range value at a vertex of the GVG) at the current robot location (which of course we do not know according to the garbage collector but we can marginalize over it). The first term in the integral, $p(\zeta_k|w_k, C_k, u^k, z^{k-1}, \nu^k, \zeta^{k-1})$ is the likelihood of measuring an equidistance ζ_k given that the true equidistance is w_k. The second term is the distribution over what the true equidistance could be given no other information other than the fact that we know the robot is not in one of the existing hypotheses. This is essentially a prior over possible true equidistance values and is difficult to define. We choose a uniform distribution over a fixed range of positive equidistance values $[0,w_{max}]$, as seen in Fig. 9.

After inserting the appropriate models into the measurement likelihood in Eq. 8, we arrive at the following,

$$\begin{aligned} & p(\zeta_k|C_k, u^k, z^{k-1}, \nu^k, \zeta^{k-1}) \\ & \quad = \frac{1}{\sqrt{2\sigma^2\pi}} \int_0^{w_{max}} e^{\frac{-(\zeta_k - w_k)^2}{2\sigma^2}} \left(\frac{1}{w_{max}}\right). \\ & \quad = \frac{1}{2w_{max}} \left(\text{Erf}\left(\frac{\zeta_k}{\sqrt{2\sigma_T^2}}\right) - \text{Erf}\left(\frac{\zeta_k - w_{max}}{\sqrt{2\sigma_T^2}}\right)\right) \end{aligned} \tag{9}$$

where σ_T^2 is again the variance on the equidistance measurement ζ_k.

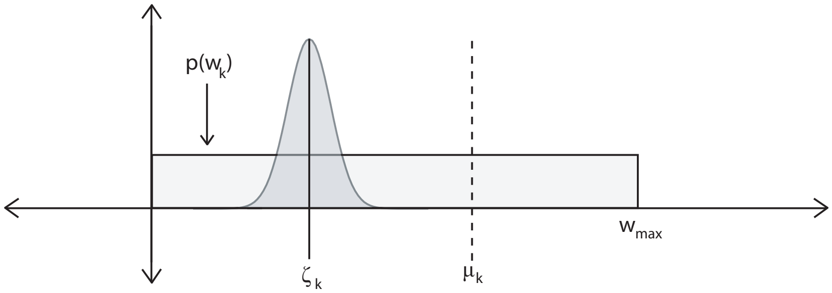


Figure 9: This shows the garbage collector hypothesis measurement update step. Compared to the known equidistance μ_k of the only remaining submap hypothesis, the measured equidistance ζ_k is very different, suggesting that the robot is not where it expects. In this case, the probability of the hypothesis drops and the garbage collector hypothesis probability increases through normalization.

In Fig. 9, we show an example visualization of the garbage-collector measurement update process. In this situation, a noisy measurement ζ_k is obtained by the range sensor of the robot. Associated with this measurement is a Gaussian uncertainty (as can be seen in the figure). There happens to be one remaining hypothesis M_k that has a true equidistance range μ_k, depicted by a dotted line in the figure. It is easy to note that the measurement does not align well with the expected true equidistance value for the submap in which we expect the robot to be. Two things will happen in this situation. The first is that the remaining submap hypothesis will be severely penalized by the disagreement of the sensor measurement through the topological measurement update. The second is that the garbage-collector hypothesis will *not* be severely penalized because the measurement aligns well with the range of possible true equidistances $[0,w_{max}]$ (for any other previously hypothesized submaps). After the normalization step, the garbage-collector hypothesis will then be pushed higher while the probability of the one remaining submap hypothesis will drop. If measurements continue to disagree with the existing submap, the garbage-collector will dominate the probability distribution and localization or SLAM should be reset.

6 Evaluation

To validate our hybrid filtering algorithm, we conducted several experiments with a mobile robot in an office-like environment to test global localization with an a priori map, the kidnapped robot problem, topological SLAM in an ambiguous map, and metric localization/SLAM.

6.1 Experimental Setup

For all of our hybrid topological localization and SLAM experiments, we used a two wheeled differential drive robot that has an array of sonar sensors configured in a ring around the robot. The robot also has an omnidirectional camera system that is implemented with a conventional firewire camera pointing up at a parabolic mirror. An example of the omnidirectional camera image that we use is shown in Fig. 10. Additionally, encoders on the wheels of the robot are used to measure the robot's translational and rotational velocity as it navigates the environment.

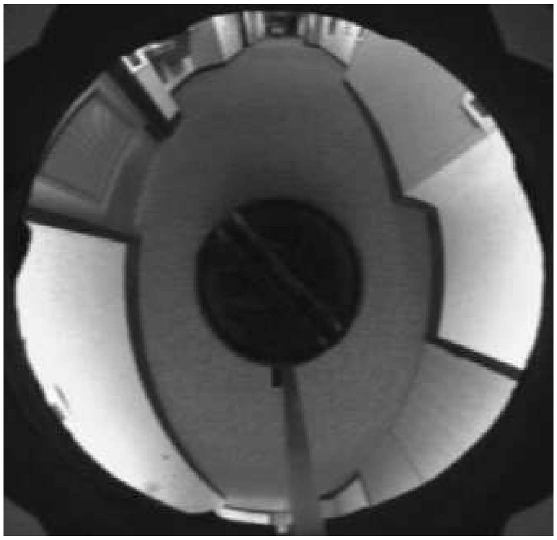


Figure 10: This is an example of the omnidirectional images that we obtain for metric measurements as the robot navigates along a GVG edge in the environment.

We have implemented a sensor-based control law for the robot to navigate along the an edge of the Voronoi diagram using the sonar ring onboard the robot. Our implementation is similar to that which is presented in (Choset and Nagatani 2001). The edge tracing algorithm essentially uses the sonar ring to measure local minima in the sensed range to the obstacles surrounding the robot. The control law seeks to steer the robot so as to equalize the range to the nearby obstacles, thus achieving two-way equidistance. For more details see (Choset and Nagatani 2001).

The sonar ring is also used to detect the GVG vertices in the environment. If, while computing the local minima of the range to the surrounding obstacles, three-way equidistance is observed, then the robot will apply a different control law to drive the robot onto the true GVG vertex position. This is a very robust homing process, which is particularly important for the algorithm we are introducing, as it provides a guarantee that the starting position of the robot upon subsequent visits to the same submap will be repeatable. For more details see (Choset and Nagatani 2001).

While the sonar sensors are used to navigate the GVG, we do not use them for metric feature sensing within the submaps. This is because feature extraction from range data is not necessarily robust. Also, we instead choose to use an inexpensive omnidirectional camera, which can be used to detect robust features in the environment. To perform feature extraction, we have implemented a custom Hough transform that detects vertical lines in the environment that typically correspond to doorways. The reason for using two sensors (sonar and vision) is that both have their advantages in the implementation of our algorithm.

6.2 Global Localization

To validate the use of our approach for solving global localization with an a priori map, we built a large dataset with which we post-process the data offline in order to test various localization tasks. The experiment was performed in an office-like building at Carnegie Mellon University. The dataset spans 5 separate floors of an office building and provides a large discontinuous global map with numerous ambiguities that, for this reason, creates a challenging environment for mobile robot localization. We captured this dataset by having the robot autonomously trace the edges of the GVG while it gathered metric sensor data from the omnidirectional camera and topological sensor data from the sonar range sensors.

The high level topological graphs corresponding to the floor plans of the experimental map are shown in Fig. 11. There are a total of 47 topological vertices in the environment and a total of 106 different submaps in which the robot may reside during localization. The submaps correspond to the GVG edges in the topological graph (two submaps are stored per edge corresponding to one submap per direction of travel along the edge).

For one of the experiments, we had the robot travel from vertex 23 to vertex 22 and then to vertex 25 within the 6-th floor of the building. In Fig. 12, we show the state of the filtering process throughout this experiment. The submap hypotheses all started with equal probability at the beginning of the experiment (signifying the robot's uncertainty in its location). Then, after sensing the clos-

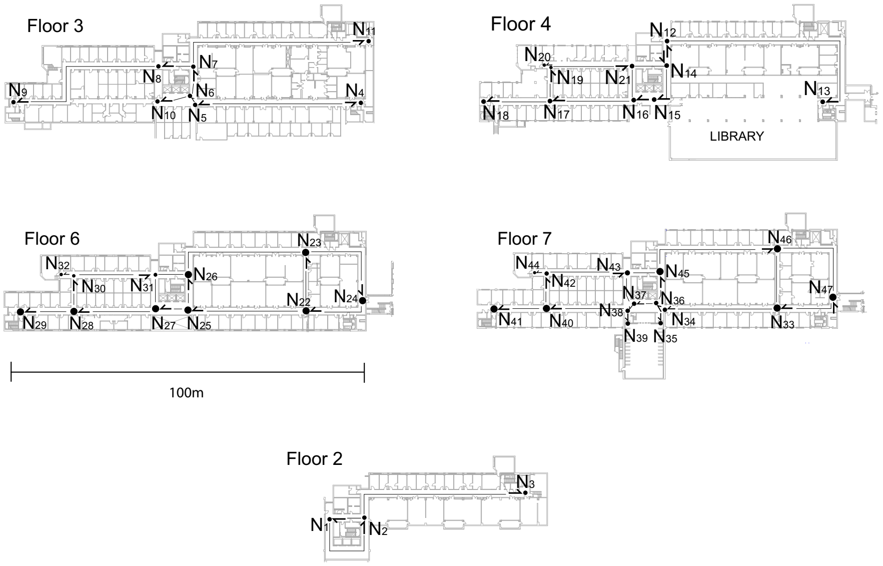


Figure 11: This is the global map for our experiments which is comprised of a disconnected topological graph that includes several floors of an office-like environment.

est vertex in the GVG, the robot acquires an equidistance measurement and applies a topological measurement update. This boosted the probabilities of submaps that have similar known equidistance values and lowered the probabilities of submaps for which the stored equidistance was very different than what was observed. The effect of this update can be seen in region $R1$ in Fig. 12.

The robot then initialized the bank of metric tracking filters for metric localization so that it could estimate its pose within each of the submaps. The robot departed the GVG vertex towards vertex 22. While traveling within the submap, the robot measured visual features that were detected by the omnidirectional camera and applied a measurement update to correct the metric filters as well as the discrete probability distribution over topological hypotheses. The effect of the metric measurements on the topological submap hypotheses can be seen in Fig. 12 (Region $R2$). The majority of submap hypotheses are pruned out because the alignment and association of the measurements do not agree. In this case, only two hypotheses remain (the correct hypothesis and the equivalent hypothesis on the 7-th floor that is ambiguous).

The robot then updates its distribution with another vertex equidistance measurement and another edge traversal with metric measurements (see Fig. 12: Regions $R3$-$R5$). Throughout this edge traversal, two hypotheses are still ambiguous. It is not until the robot makes a third vertex equidistance measurement

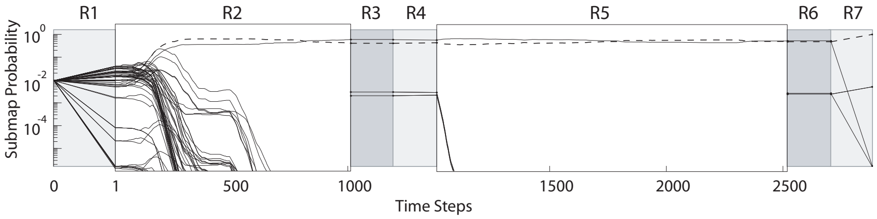


Figure 12: In this experiment, the robot travels from vertex 23 to vertex 22 and then to vertex 25 in the topological graph. In $R1$, the vertex equidistance measurements favors some of the hypotheses. In $R2$ the landmark measurements along the GVG edge rule out all but two hypotheses. In $R3$ and $R4$, the motion model is applied and another equidistance measurement is made. In $R5$ more landmark measurements are made. In $R6$ and $R7$, the robot eliminates all but the correct topological submap hypothesis.

(see Fig. 12 Region $R7$) that the correct hypothesis is detected and dominates the probability distribution. In this experiment, localization was performed successfully without prior information of the robot's location.

The run-time for performing localization with an a priori map scales with the number of submaps in the stored map. In realistic environments, we achieve real-time performance. For a theoretical map with an extremely large number of submaps, the performance would degrade.

6.3 Kidnapped Robot

The second experiment demonstrates the robot's ability to recover from the so called "kidnapped robot" problem. In this experiment, the robot again travels from vertex 23 to vertex 22 and upon arriving at vertex 22, the robot is deliberately moved to a vertex on a completely different floor (the 7-th floor) without the robot knowing. This is an unmodeled disturbance for localization and represents a challenging situation for the system. The robot must determine that the current hypotheses, in which it has developed trust, are not correct.

In Fig. 13, we can see the submap hypotheses as they evolve throughout the experiment. In red, we show the garbage collector hypothesis. The robot is kidnapped at approximately time step 1000. At first, due to ambiguity between the expected hypothesis equidistance and the measured equidistance at the kidnapped location, the robot trusts the current set of hypotheses (the garbage collector has a low probability). Eventually though, after updating the garbage-collector hypothesis three times, the robot has lost it confidence in the current set of hypotheses and believes confidently that its location is in one of the pruned hypotheses (the garbage collector has a high probability). This is a successful kidnapped robot detection. At this point, the global localization algorithm can be reset.

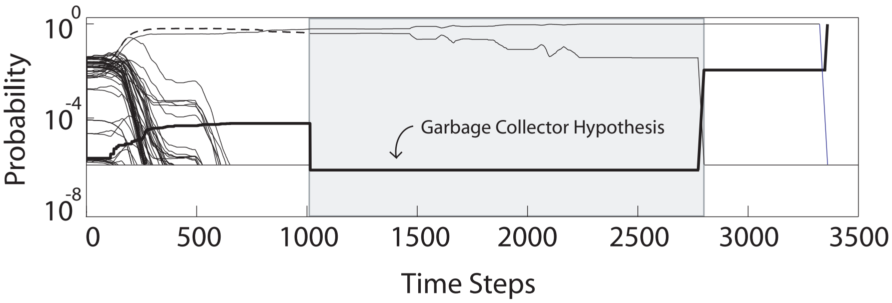


Figure 13: This shows a kidnapped robot localization experiment. Along the first traveled edge, the robot resolves its pose to one of two submap hypotheses. The robot is then carried to another floor in the building. The robot detects the kidnapped robot occurrence with its garbage collector hypothesis after two vertex equidistance measurements.

6.4 Hybrid SLAM

To test SLAM, we recorded a library of data from real experiments that were performed in the map depicted in Fig. 2. During the experiments, we had the robot store a distance traveled measurement for each GVG edge and an equidistance measurement for each GVG vertex over many trials. By creating this large library of measurement data, we are able to post-process the data and completely recreate in simulation the robot performing real experiments and acquiring real sensor measurements but with the added benefit that we can alter the path that the robot takes through the graph by simply reordering the measurement sequence that is provided to the robot.

For validation, we ran a challenging experiment that is based on a ground truth map that has a much larger amount of ambiguity than the map shown in Fig 2. The graph we used for this experiment is shown in Fig. 14, and was made by adding a number of extra vertices and edges to the original floor plan of Fig. 2. This example is used to demonstrate our algorithm's ability to handle the problem of perceptual aliasing because of the large amount of ambiguity. Although this map is artificial, the vertices and edges are duplicated from the original map and therefore we can still recreate real sensor measurements as if the robot were actually traveling in this environment.

We ran the experiment in the map depicted in Fig. 14 with the robot performing 500 random edge traversals and fully exploring the map. The number of hypotheses tracked with our hypothesis forest throughout the experiment is shown in Fig. 15 (a). In the beginning of the experiment, the number of hypotheses grows quickly because of the large amount of ambiguity in the map. Eventually, pruning begins to reduce the number of hypotheses. At the end of the experiment, there is only one hypothesis that remains and it is the correct state and map.

In Fig. 15 (b), the posterior probability for the correct hypothesis is dis-

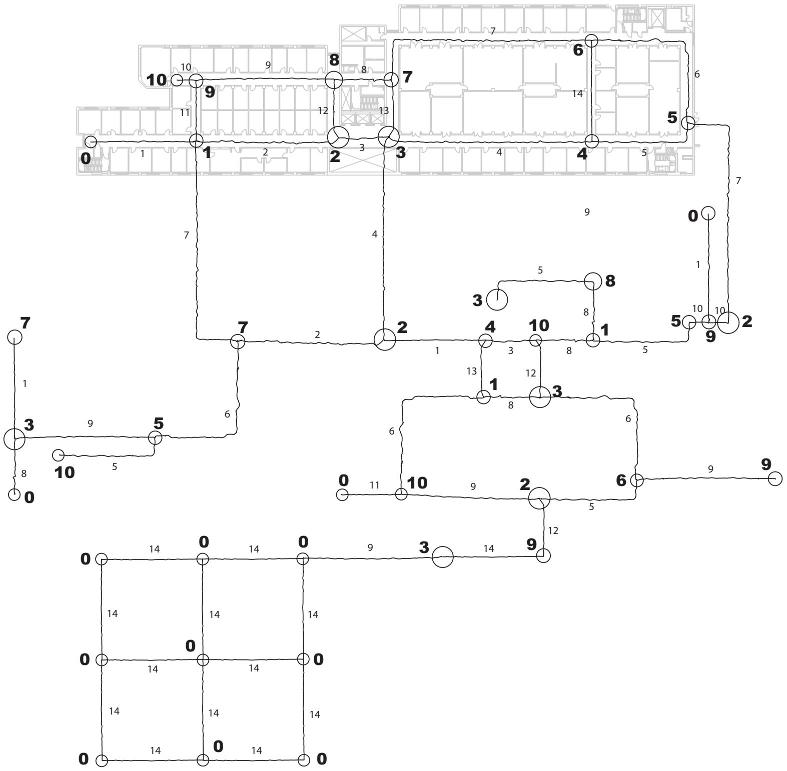


Figure 14: This is a map created to simulate a much larger and more ambiguous version of Wean Hall at Carnegie Mellon University. Each number next to a vertex represents the corresponding true vertex from which this newly added vertex has been copied. Each number next to an edge represents the corresponding true edge from which this newly added edge has been copied.

played throughout the 500 edge traversals for Experiment 2. There are certain times during the experiment when the robot associates a very low probability to the correct hypothesis. This could be caused by the existence of a different hypothesis that has a smaller map but also conforms to the sequence of measurements that was obtained. Eventually though, as seen in Fig. 15, this other hypothesis becomes inconsistent and is discarded. The correct hypothesis then reemerges as a good candidate with a high probability.

In Figure 13, the lower part of the map consists of vertices numbered 0 and edges numbered 14. Because all of these vertices and edges are equivalent, it leads to a highly ambiguous mapping task for the robot. But despite the ambiguity, our algorithm remarkably maps this topology accurately and efficiently. The reason that our approach can handle this extreme case of perceptual aliasing is that the filtering algorithm is naturally (over multiple time steps) evaluating hypotheses based on the sequence of observations that the robot obtains.

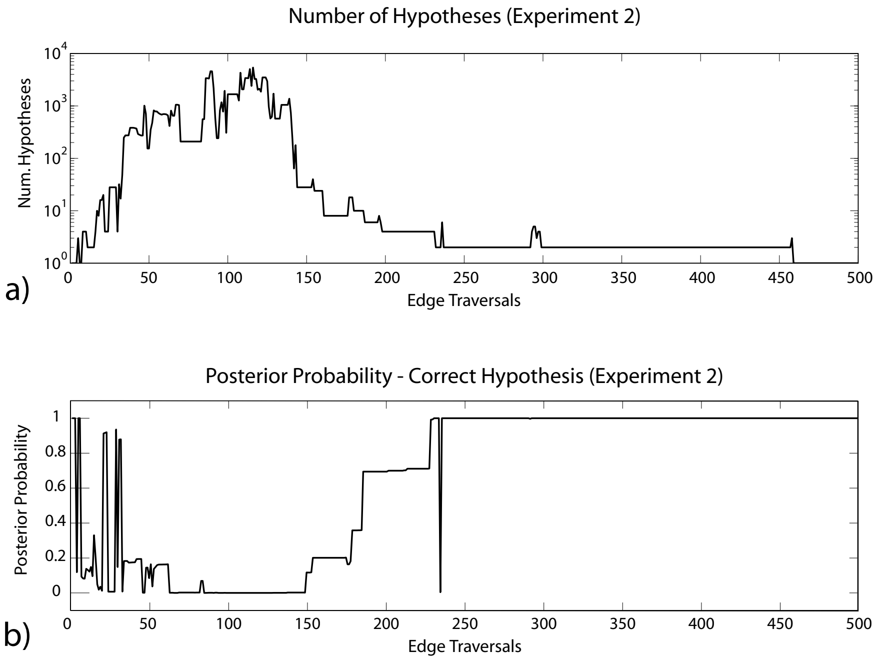


Figure 15: (a) This is a plot of the number of hypotheses throughout our SLAM experiment. (b) This is a plot that shows the posterior probability tracked for the correct hypothesis during our SLAM experiment.

And because there is sufficient uniqueness in neighboring vertices to this ambiguous region, there can only be one map that would explain the sequence of observations that are obtained.

In comparison, we note that the implementations in (Dudek et al. 1993; Savelli and Kuipers 2004) only remove hypotheses in the tree when the graph becomes inconsistent or when planarity fails. If these implementations were run on our data set, we would expect the number of hypotheses to grow beyond what is computationally feasible.

The complexity of performing hybrid SLAM to map the topology of an environment scales with the number of hypotheses that are stored in the hypothesis forest. This number is difficult to predict, and thus the run-time for a given experiment can vary. In general, the number of hypotheses will depend on the complexity of loop-closure and the ambiguity in the map. For extremely large maps, there will be significantly more hypotheses to evaluate which may slow down the processing of this algorithm.

6.5 Metric Localization

When the robot is performing global localization, there are two aspects of determining the robot's position: the discrete localization problem to estimate the submap the robot is in and the metric problem of localizing the robot in the submap. The metric localization task is achieved by using metric measurements obtained along the traversal of the GVG edge to update a bank of extended Kalman filters (one for each submap hypothesis) that is estimating the metric pose (x,y,θ) of the robot in each submap.

For our experiments, the robot obtains bearing measurements to doorway features in the omnidirectional camera images using a line detector. The bearing measurements are used to update the EKF for each submap via a nonlinear measurement model. In Fig. 16, we depict an example of the robot having traversed a GVG edge with bearing measurements to doorway features. Also in Fig. 16, the performance of our metric localization process is compared with an odometry-only result. Using metric localization, the robot was localized to within 0.18 meters of the ground truth position at the end of the edge traversal. On the other hand, using odometry only, the error was 1.92 meters.

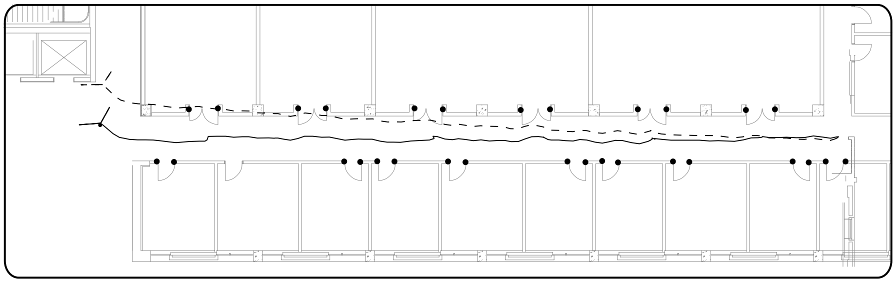


Figure 16: This demonstrates the metric localization performance of our algorithm. The metric estimate using the EKF is shown aligning well with the ground truth position (a GVG vertex location) while the odometry path (dotted line) has significant error.

The run-time for the metric component of our hybrid localization scheme scales with the number of hypotheses. This is because we are running an EKF for each submap hypothesis and each EKF is updated in constant-time (with just estimated 3 parameters for each EKF). We note that only with extremely large maps (with many submaps) will this become computationally intensive.

6.6 Metric SLAM

In the global localization experiment, we assumed an a prior known map to test the robot localizing without a priori knowledge of its location. Instead, if the local metric maps attached to each topological edge are not known before hand, the robot must construct each metric submap as it performs the hybrid metric/topological SLAM algorithm presented in this paper.

In Fig. 17, we show the result of an edge traversal performed by the robot in which the line features extracted from the omnidirectional camera are treated as bearing-only landmark measurements and are mapped with an Iterated Extended Kalman filter (IEKF). This local metric SLAM approach is based on some of our previous work in bearing-only SLAM (Tully et al. 2008, 2010).

The resulting map is shown in Fig. 17 with small (almost fully converged) uncertainty ellipses It is important to show the performance of this local metric approach as it justifies the use of our hybrid filtering algorithm for hybrid metric/topological SLAM. The average error between a mapped landmark location and its ground ground truth location during this edge traversal was 0.1356 meters.

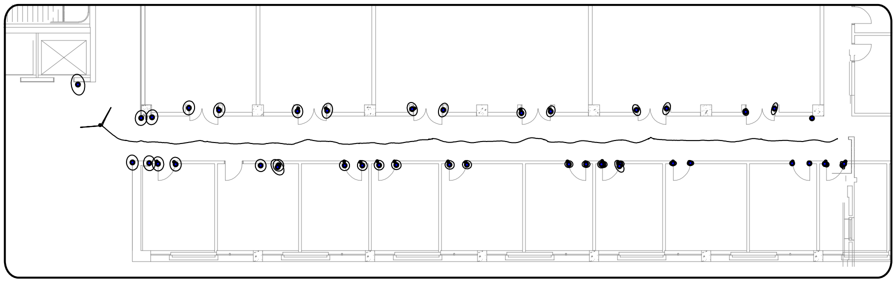


Figure 17: This validates the accuracy of our feature-based metric SLAM method for mapping local maps along a GVG edge.

The run-time of this algorithm will scale with the complexity of conventional metric EKF feature-based SLAM. The implementation is an $O(N^2)$ algorithm, where N in this case is the number of features in the map. On the other hand, it is reasonable to assume that the length and number of features within each topological edge are small. The mapping of each edge with an EKF, therefore, can be performed in real-time.

7 Conclusion

We have presented a novel unified filtering framework for achieving localization and mapping in hybrid metric/topological maps. The contributions of this work are: 1) the introduction of a novel hypothesis forest expansion algorithm specific to edge-ordered graphs for proposing possible topological map/state pairs, 2) a new algorithm for recursively computing the posterior probability of a topological hypothesis given a sequence of sensor measurements and a novel topological prior, 3) the design of conservative pruning rules that reduce the number of hypotheses in the forest, and 4) the use of a garbage-collector technique that is a catch-all for evaluating the likelihood of pruned hypotheses.

In this work we have improved our previous work (Tully et al. 2007, 2009) in several ways. In our original algorithm, we had the robot wait until reaching a topological vertex to apply sensor measurements to the filtering algorithm. Our new algorithm applies this information continuously throughout an edge traversal, which can allow the robot to localize more quickly. Our new approach is also more robust, with a built in method to detect unmodeled disturbances, and improved processing of sensor information within the recursive likelihood update.

When the map has a large number of ambiguities, the size of the tree can grow quickly. The complexity of the algorithm can even become intractable for large ambiguous maps. To approach this problem, we suggest the following implementation change, which is a subject of future work: the algorithm will invoke a cap on the number of hypotheses to avoid complexity issues. But, if good hypotheses are unfairly pruned out due to the cap, the algorithm can eventually revive them and replay the measurement sequence stored in memory to bring them back up to speed at the current time-step. With a cap of 1 this would equate to a depth first search of the correct inferred hypothesis tree instead of what we presented, which is essentially a breadth first search.

In our experiments, we show a robot localizing as well as mapping quickly and efficiently in a large office-like environment with numerous ambiguities. We demonstrate an example localization experiment that localizes properly to a metric location within the correct topological submap. We also show an example experiment that tests our algorithm's ability to handle the so-called "kidnapped robot problem." Lastly, we show an example in which SLAM is performed successfully to determine the correct topology of an environment despite numerous cases of perceptual aliasing.

One weak point of this algorithm is that it is designed for structured environments. Despite this fact, we believe that there are significant academic contribution that we are presenting here. The method we propose is useful for a variety of indoor environments that have structure, thus the work is applicable to robots navigating indoor environments. Also, the method is very generalizable to any local metric method that can be used, as long as the environment can be subdivided using a higher level topological graph. Lastly, in the future, it may be that hybrid approaches, such as the one we present here, will improve the accuracy and efficiency of SLAM and localization in outdoor environments as long as appropriate topologies and control laws can be defined.

The overall impact of our work is that we can achieve global localization or SLAM more efficiently than conventional metric methods. When an algorithm like GraphSLAM (Thrun and Montemerlo 2006) attempts to close a large loop, it must perform a global data association problem to associate a sensor measurement with one of many vertices in the graph network. This can be infeasible with large loops and long paths, especially when their is significant positional drift. Instead, we can perform data association at a higher level (on the a topological representation of the environment), thus drastically reducing the number of hypotheses that need to be evaluated. While it is the topological nature of our solution that allows for efficient estimation, we believe it is our algorithms that can make topological mapping and localization work robustly and effectively.

Acknowledgments

The authors would like to thank Deryck Morales, Hyungpil Moon, and Felix Werner for their help with algorithm development, software integration, and experiments.

References

H. Akaike. A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6):716–723, Dec 1974. ISSN 0018-9286.

A. Angeli, S. Doncieux, J.-A. Meyer, and D. Filliat. Incremental vision-based topological SLAM. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 1031 – 1036, sept. 2008.

T. Bailey and H. Durran-Whyte. Simultaneous localization and mapping: Part II. *IEEE Robotics and Automation Magazine*, pages 109–117, JUNE 2006.

J.-L. Blanco, J.-A. Fernandez-Madrigal, and J. Gonzalez. A new approach for large-scale localization and mapping: Hybrid metric-topological slam. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 2061 – 2067, April 2007.

Jose-Luis Blanco, Juan-Antonio Fernandex-Madrigal, and Javier Gonzalez. Toward a unified bayesian approach to hybrid metrictopological slam. *IEEE Transactions on Robotics and Automation*, 24(2):259 – 270, 2008.

W. Burgard, D. Fox, D. Hennig, and T. Schmidt. Estimating the absolute position of a mobile robot using position probability grids. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-96)*, 1996.

Wolfram Burgard, Andreas Derr, Dieter Fox, and Armin B. Cremers. Integrating global position estimation and position tracking for mobile robots: The dynamic markov localization approach. In *In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1998.

A. Cassandra, L. Kaelbling, and J. Kurien. Acting under uncertainty: Discrete bayesian models for mobile-robot navigation. In *Proceedings of the International Conference on Intelligent Robots and Systems*, 1996.

H. Choset and J. W. Burdick. Sensor based planning, part I: The generalized voronoi graph. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 1995.

H. Choset and K. Nagatani. Topological simultaneous localization and mapping (SLAM): toward exact localization without explicit localization. *Robotics and Automation, IEEE Transactions on*, 17(2):125–137, Apr 2001. ISSN 1042-296X.

J. Civera, A. J. Davison, and J. M. M. Montiel. Inverse depth to depth conversion for monocular SLAM. In *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, April 2007.

A. De, J. Lee, and N. Cowan. Toward SLAM on graphs. *Workshop on the Algorithmic Foundations of Robotics, 2008. WAFR 2008.*, December 2008.

F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte carlo localization for mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 1999.

M. Dissanayake, P. Newman, H. Durrant-Whyte, S. Clark, and M. Csorba. A solution to the simultaneous localisation and map building (SLAM) problem. *IEEE Transactions of Robotics and Automation*, 17(3):229–241, June 2001.

G. Dudek, P. Freedman, and S. Hadjres. Using local information in a non-local way for mapping graph-like worlds. *Proc. the 3rd International Conference on Artificial Intelligence*, pages 1639–1645, 1993.

Gregory Dudek, Paul Freedman, and Souad Hadjres. Using multiple models for environmental mapping. *Journal of Robotic Systems*, 13(8):539–559, 1996.

H. Durran-Whyte and T. Bailey. Simultaneous localization and mapping: Part I. *IEEE Robotics and Automation Magazine*, pages 99–108, JUNE 2006.

A. Eliazar and R. Parr. Dp-slam: Fast, robust simultaneous localization and mapping without predetermined landmarks. In *In Proc. of the Int. Conf. on Artificial Intelligence (IJCAI)*, 2003.

D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte carlo localization: Efficient position estimation for mobile robots. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 1999a.

D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 11:391–427, 1999b.

G. Grisetti, C. Stachniss, and W. Burgard. Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 2432 – 2437, April 2005.

P. Jensfelt and S. Kristensen. Active global localization for a mobile robot using multiple hypothesis tracking. *IEEE Transactions on Robotics and Automation*, 17(5), October 2001.

K. Kouzoubov and D. Austin. Hybrid topological/metric approach to slam. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 1, pages 872 – 877, April 2004.

B. Kuipers, J. Modayil, P. Beeson, M. MacMahon, and F. Savelli. Local metrical and global topological maps in the hybrid spatial semantic hierarchy. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 5, pages 4845 – 4851, April 2004.

Benjamin Kuipers and Yung-Tai Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and Autonomous Systems*, 8(1-2):47 – 63, 1991. Special Issue Toward Learning Robots.

B. Lisien, D. Morales, D. Silver, G. Kantor, I. Rekleitis, and H. Choset. The hierarchical atlas. *Robotics, IEEE Transactions on*, 21(3):473–481, June 2005. ISSN 1552-3098.

D. Marinakis and G. Dudek. Pure topological mapping in mobile robotics. *Robotics, IEEE Transactions on*, 26(6):1051 – 1064, December 2010.

M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the 2002 AAAI National Conf. Artificial Intelligence*, 2002.

M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, 2003.

J. M. M. Montiel, J. Civera, and A. J. Davison. Unified inverse depth parameterization for monocular SLAM. In *Robotics Science and Systems*, 2006. URL `http://www.roboticsproceedings.org/rss02/p11.html`.

I. Nourbakhsh, R. Powers, and S. Birchfield. Dervish: An office-navigating robot. *AI Magazine*, 16(2):53–60, 1995.

Ananth Ranganathan and Frank Dellaert. Online probabilistic topological mapping. *The International Journal of Robotics Research*, 2011.

E. Ranganathan, E. Menegatti, and F. Delleart. Bayesian inference in the space of topological maps. *IEEE Trans. Robot. Autom.*, 22(1):92–107, February 2006.

E. Remolina and B. Kuipers. Towards a general theory of topological maps. *Artificial Intelligence*, 152(1):47–104, 2004.

S. Roumeliotis and G. Bekey. Bayesian estimation and kalman filtering: A unified framework for mobile robot localization. In *Proceedings of the International Conference on Robotics and Automation*, April 2000.

F. Savelli and B. Kuipers. Loop-closing and planarity in topological map-building. *Intelligent Robots and Systems, 2004. IROS 2004. IEEE/RSJ International Conference on*, pages 1511–1517, 2004.

G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6: 461–464, 1978.

R. Simmons and S. Koenig. Probabilistic robot navigation in partially observable environments. In *Proceedings of the International Joint Conference on Artificial INtelligence*, 1995.

R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. *Autonomous Robot Vehicles, Springer*, 1990.

S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2):99–141, 2000.

Sebastian Thrun and Michael Montemerlo. The graph slam algorithm with applications to large-scale mapping of urban structures. *The International Journal of Robotics Research*, 25(5-6):403–429, 2006.

N. Tomatis, I. Nourbakhsh, and R. Siegwart. Hybrid simultaneous localization and map building: a natural integration of topological and metric. *Robotics and Autonomous Systems*, 2002.

S. Tully, H. Moon, D. Morales, G. Kantor, and H. Choset. Hybrid localization using the hierarchical atlas. *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 2857–2864, Nov 2007.

S. Tully, H. Moon, G. Kantor, and H. Choset. Iterated filters for bearing-only SLAM. In *Proceedings of the 2008 IEEE International Conference on Robotics and Automation*, May 2008.

S. Tully, G. Kantor, H. Choset, and F. Werner. A multi-hypothesis topological slam approach for loop closing on edge-ordered graphs. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 4943 – 4948, October 2009.

S. Tully, G. Kantor, and H. Choset. A single-step maximum a posteriori update for bearing-only SLAM. In To appear *in proceedings of the 2010 AAAI Conf. Artificial Intelligence*, July 2010.

G. Vijayan and A. Wigderson. Planarity of edge ordered graphs. *Technical Report 307, Princeton University*, TR307, December 1982.

F. Werner, F. Maire, and J. Sitte. Topological SLAM using fast vision techniques. In *FIRA, ser. LNCS*, number 5744, pages 187 – 196, August 2009a.

F. Werner, F. Maire, J. Sitte, H. Choset, S. Tully, and G. Kantor. Topological slam using neighbourhood information of places. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 4937 – 4942, October 2009b.