The 2009 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 11-15, 2009 St. Louis, USA

# A Multi-Hypothesis Topological SLAM Approach for Loop Closing on Edge-Ordered Graphs

Stephen Tully, George Kantor, Howie Choset, and Felix Werner

*Abstract*— **We present a method for topological SLAM that specifically targets loop closing for edge-ordered graphs. Instead of using a heuristic approach to accept or reject loop closing, we propose a probabilistically grounded multi-hypothesis technique that relies on the incremental construction of a map/state hypothesis tree. Loop closing is introduced automatically within the tree expansion, and likely hypotheses are chosen based on their posterior probability after a sequence of sensor measurements. Careful pruning of the hypothesis tree keeps the growing number of hypotheses under control and a recursive formulation reduces storage and computational costs. Experiments are used to validate the approach.**

## I. Introduction

Simultaneous localization and mapping (SLAM) is the task of incrementally building a map of the environment with a robot while simultaneously performing localization within that map. In the past decade, there has been an intense research effort to solve this problem accurately and efficiently. The methods introduced are based on three types of maps: feature maps [1], [2], grid or sample based obstacle maps [3], [4], and topological maps [5]–[7].

Topological maps are concise maps that represent an environment as a graph, whose vertices are interesting "places" and whose edges represent the paths between them. The advantages of topological maps are their computational efficiency, their reduced memory requirements, and their lack of dependence on metric positioning.

Loop-closing for topological mapping is the problem of detecting when a robot has returned to a previously visited vertex in the 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.

The primary issue with many of the existing topological SLAM techniques is that they commit to a loop-closure heuristically when two observations appear similar. If the loop-closing decision is incorrect, the algorithm cannot recover and the resulting experiment will fail. We use a multi-hypothesis approach that avoids this problem entirely by storing a tree of possible hypotheses, each of which encodes the robot's state and a topological graph.

S. Tully is with the Electrical and Computer Engineering Department at Carnegie Mellon University, Pittsburgh, PA 15213, USA. G. Kantor and H. Choset are with the Robotics Institute at Carnegie Mellon University, Pittsburgh, PA 15213, USA. {stully@ece, kantor@ri, choset@cs}.cmu.edu. F. Werner is with the Information Technology Department at Queensland University of Technology, Brisbane, Australia. felix.werner@nicta.com.au.

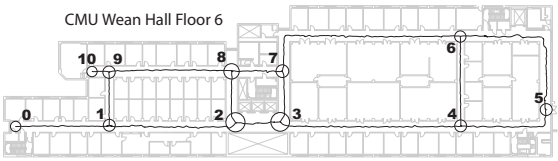


Fig. 1. This is a floor plan of Wean Hall at Carnegie Mellon University with the Voronoi graph drawn to depict the topology.

The contribution of this work is the design of a tree expansion algorithm specific to edge-ordered graphs, as well as the introduction of a customized method for recursively computing the posterior probability over the topological map hypotheses. This posterior probability is based on a Bayesian model selection criterion that prevents over-fitting. Lastly, this work introduces a set of conservative pruning rules that help reduce the number of hypotheses in the tree.

Our experimental evaluation relies on the sensor-based incremental construction of the Voronoi diagram of an environment with a mobile robot, as in [5]. The *Generalized Voronoi Graph* (GVG) is the resulting topological graph whose vertices are points of three-way equidistance and whose edges are obstacle-free paths between vertices. See Fig. 1 for an example map.

## II. Related Work

Many 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. Choset et. al. [5] use the degree and equidistance measures at the nodes of a Voronoi diagram to determine if the robot has returned to a previously visited vertex. Similarly, Tomatis et. al. [8] 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.

A multi-hypothesis approach is necessary to investigate multiple loop closure proposals. Several other papers, besides this one, investigate the use of a hypothesis tree to store possible topological maps. Dudek in [9] constructs the tree and eliminates hypotheses when they are inconsistent. Savelli et. al. in [10] use a tree and analyze the affect of planarity to reduce the number of hypotheses. Neither of these solutions, though, computes a probability measure over the set of hypotheses to infer the correct topology.

Another multi-hypothesis approach is one that samples over the space of possible topologies [11]. This method decides to split or merge vertices based on their metric locations by penalizing the placement of nearby vertices.

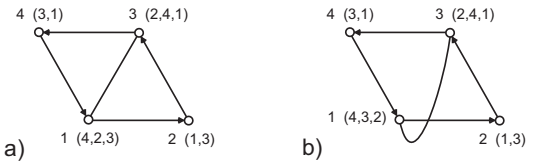


Fig. 2. This is an example of two edge-ordered graphs. The values next to each vertex are the vertex indices, while the parenthetical sequences represent the edge-ordered neighbor lists associated to each vertex. The first mapped edge for each vertex is shown with an arrow.

Although the algorithm is similar in its use of Bayesian inference, we believe it may have difficulty closing large loops with significant positioning error.

De et. al. [12] use an E-M algorithm to generate topological maps that are consistent with the data. They incorporate a model selection criterion to penalize over-fitting with notable success. Unfortunately, their novel approach is only applicable to graphs that have one or two cycles.

## III. Constructing a Hypothesis Tree

We adopt a multi-hypothesis approach to SLAM. Each hypothesis $h$ stores the robot's state on that graph, $X_k^h$, as well as a possible edge-ordered topological graph, $G_k^h$, to be defined shortly. The state is represented by the vertex at which the robot is currently located, $v_k^h$, as well as the edge from which the robot arrived at that vertex, $\alpha_k^h$, thus $X_k^h = (v_k^h, \alpha_k^h)$. The subscript $k$ represents the time step that is associated with that hypothesis.

### A. Edge-ordered Topological Graphs

For each hypothesis $h$, a possible topological graph is stored, which, in our case, is an edge-ordered graph. This type of graph can be represented by the number of vertices $N_k^h$ and a set of circular neighbor lists $L_k^h$ (one list per vertex), thus $G_k^h = (N_k^h, L_k^h)$, as in [13]. A neighbor list, such as $L_k^h(v_k^h)$ stores the vertices in the graph that are neighbors of vertex $v_k^h$ in the order they occur (counter-clockwise from the first mapped edge). An element of the neighbor list $L_k^h(v_k^h, j)$ represents the neighboring vertex of $v_k^h$ along the $j$-th edge. Fig. 2 shows two edge-ordered graphs with similar topologies but different edge-orderings.

For this work, we also consider partially explored maps. In this case, a neighbor list in the graph contains 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.

### B. Incremental Construction of a Hypothesis Tree

Our goal is to incrementally build a set of hypotheses that can completely reproduce the possible map/state pairs at every time step $k$. To do this, we maintain a hypothesis tree where each level of the tree represents a different time step in the experiment. Therefore, a level of the tree is indexed with $k$ and a hypothesis within that level is indexed with $h$. The tree structure we maintain is similar to that in [7], [9].

The robot begins an experiment at one vertex in the map. The robot has no other information except for the degree of that vertex, $\delta_0$, which equals the number of edges emanating from the vertex. Therefore, we initialize the root of the hypothesis tree 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 $\delta_0$ for which each entry is labeled as $unexplored$. All hypotheses in the tree are ultimately spawned from this initial root hypothesis.

The robot is continuously moving. At each 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 previous arrival edge, and produces the following departure edge $\beta_k$ for a new hypothesis that is spawned from hypothesis $h$.

$$\beta_k = (\alpha_{k-1}^h + u_k) \bmod \delta_{k-1} \tag{1}$$

After departing along edge $\beta_k$, the robot drives to a new vertex and then detects the number of edges emanating from that vertex, which is stored as the degree $\delta_k$.

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 relax this assumption in Sec. VII.

**Algorithm 1** Expanding the Hypothesis Tree

**for all** $h \in H_{k-1}$ **do**
  $[v_{k-1}^h,\ \alpha_{k-1}^h,\ N_{k-1}^h,\ L_{k-1}^h] \leftarrow$ LoadHypothesis($h$)
  $\beta_k = (\alpha_{k-1}^h + u_k)\ mod\ \delta_{k-1}$
  **if** $L_{k-1}^h(v_{k-1}^h,\ \beta_k) = unexplored$ **then**
    $h' \leftarrow$ CreateChildHypothesis($h$)
    $L_k^{h'} = L_{k-1}^h$
    $L_k^{h'}(N_{k-1}^h + 1,\ 0) = v_{k-1}^h$
    **for** $e$ = 1 to $\delta_k - 1$ **do**
      $L_k^{h'}(N_{k-1}^h + 1,\ e) = unexplored$
    **end for**
    $L_k^{h'}(v_{k-1}^h,\ \beta_k) = N_{k-1}^h + 1$
    AddChild($h', N_{k-1}^h + 1,\ 0,\ N_{k-1}^h + 1,\ L_k^{h'}$)
    **for** $v$ = 0 to $N_{k-1}^h - 1$ **with** $v \neq v_{k-1}^h$ **do**
      **for all** $\alpha\ \ s.t.\ \ L_{k-1}^h(v, \alpha) = unexplored$ **do**
        $h' \leftarrow$ CreateChildHypothesis($h$)
        $L_k^{h'} = L_{k-1}^h$
        $L_k^{h'}(v, \alpha) = v_{k-1}^h$
        $L_k^{h'}(v_{k-1}^h, \beta_k) = v$
        AddChild($h',\ v,\ \alpha,\ N_{k-1}^h,\ L_k^{h'}$)
      **end for**
    **end for**
  **else**
    $h' \leftarrow$ CreateChildHypothesis($h$)
    $v_k^{h'} = L_{k-1}^h(v_{k-1}^h,\ \beta_k)$
    $\alpha_k^{h'} = e\ \ s.t.\ \ L_{k-1}^h(v_k^{h'},\ e) = v_{k-1}^h$
    AddChild($h',\ v_k^{h'},\ \alpha_k^{h'}, N_{k-1}^h,\ L_{k-1}^h$)
  **end if**
**end for**

When the robot chooses a new motion input $u_k$, we must

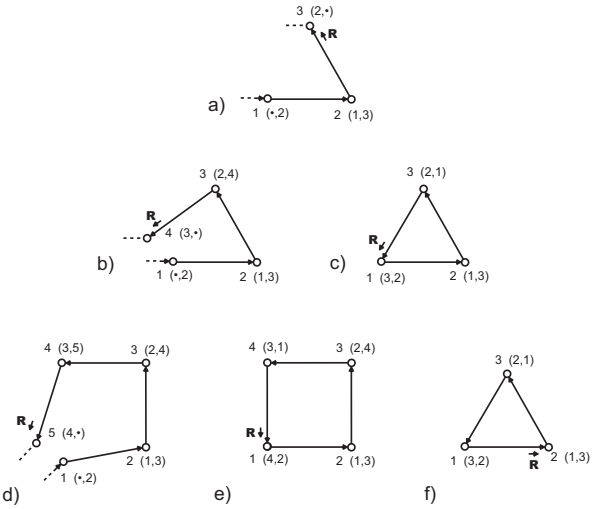


Fig. 3. This is an example of expanding the hypothesis tree due to robot motion. Hypothesis (a) spawns (b) and (c) after one edge traversal. After another edge traversal, hypothesis (b) spawns (d) and (e) while hypothesis (c) spawns only (f). The location of **R** in the figure marks the robot's state.

update the hypothesis tree by expanding all of the leaf nodes of the tree (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 tree 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 tree in the following way. If $L^h_{k-1}(v^h_{k-1}, \beta_k)$ (the neighbor of $v^h_{k-1}$ that is associated to the departing edge $\beta_k$) 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 arrival edge. If $L^h_{k-1}(v^h_{k-1}, \beta_k)$ is $unexplored$, then the algorithm considers several possibilities that would agree with hypothesis $h$. The first possibility is that the robot traverses the unexplored edge and arrives 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. 3 demonstrates the expansion of the hypothesis tree. 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.

## IV. Topological SLAM

In order to solve the problem of topological SLAM, we must determine which hypotheses among the leaf nodes of the hypothesis tree are likely to represent the true state and the true map. To do this, we compute the posterior probability of each hypothesis given a sequence of sensor measurements. 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 map.

### A. Posterior Probability

During time step $k$, the robot leaves the previous vertex, traverses an edge in the graph, and arrives at a new vertex. A measurement $z^e_k$ is obtained during the edge traversal (such as a travel distance measurement) and a measurement $z^v_k$ is obtained when the robot arrives at the new vertex (such as a range measurement to obstacles). The posterior probability of a hypothesis is as follows,

$$p(X^h_k, G^h_k | z_{0:k}, u_{1:k}), \tag{2}$$

where, as before, $X^h_k$ and $G^h_k$ represent the robot's state and graph respectively. Additionally, $z_{0:k} = (z^v_{0:k}, z^e_{1:k})$ is the collection of all measurements during the experiment, which includes the edge measurement sequence, $z^e_{1:k}$, as well as the vertex measurement sequence, $z^v_{0:k}$. The sequence $u_{1:k}$ represents the motion inputs through time step $k$.

The posterior of Eq. 2 can be computed using Bayes law,

$$\begin{aligned} p(X^h_k, G^h_k | z_{0:k}, u_{1:k}) = \\ \eta\ p(z_{0:k} | X^h_k, G^h_k, u_{1:k})\ p(X^h_k, G^h_k | u_{1:k}) \\ \eta\ p(z_{0:k} | X^h_k, G^h_k, u_{1:k})\ p(X^h_k | G^h_k, u_{1:k})\ p(G^h_k | u_{1:k}) \\ \eta\ p(z_{0:k} | X^h_k, G^h_k, u_{1:k})\ p(G^h_k | u_{1:k}), \end{aligned} \tag{3}$$

where $p(z_{0:k} | X^h_k, G^h_k, u_{1:k})$ is the measurement likelihood function and $p(X^h_k, G^h_k | u_{1:k})$ is a prior on the hypothesis. The prior reduces to $p(G^h_k | u_{1:k})$ in Eq. 3 because the probability of the state given the map and inputs, $p(X^h_k | G^h_k, u_{1:k})$, is equal to one. This is because we assume we have a robot that correctly performs the motion input sequence. The scalar value $\eta$ in Eq. 3 is used for normalization over possible hypotheses, such that the following holds true,

$$\sum_{h=0}^{H_k - 1} p(X^h_k, G^h_k | z_{0:k}, u_{1:k}) = 1,$$

where $H_k$ is the number of current leaf nodes in the hypothesis tree. This is valid because the tree's exhaustive expansion guarantees that one of the hypotheses in the leaf nodes of the tree is correct.

### B. Likelihood Function

For a given time step, after expanding the leaf nodes of the tree to account for robot motion, we compute the posterior probability of the new leaf nodes of the tree using Eq. 3. To reduce storage and computation, the likelihood term of a new hypothesis $h'$ can be computed recursively given the likelihood of the parent hypothesis $h$, i.e.,

$$\begin{aligned} & p(z_{0:k} | X^{h'}_k, G^{h'}_k, u_{1:k}) \\ &= p(z^e_k, z^v_k | z_{0:k-1}, X^{h'}_k, G^{h'}_k, u_{1:k})\ p(z_{0:k-1} | X^{h'}_k, G^{h'}_k, u_{1:k}) \\ &= p(z^e_k, z^v_k | z_{0:k-1}, X^{h'}_k, G^{h'}_k, u_{1:k})\ p(z_{0:k-1} | X^h_{k-1}, G^h_{k-1}, u_{1:k-1}) \end{aligned} \tag{4}$$

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. The hypothesis $h'$ and time step $k$ have been reverted back to the parent hypothesis $h$ and the previous time step $k-1$ in the second term of Eq. 4 in order to fit the recursive form. This is done without error or approximation due to the fact that past measurements are only dependent upon the graph and inputs before the tree expansion.

The edge measurement $z_k^e$, according to hypothesis $h$, is associated with edge $\alpha_k^h$ of vertex $v_k^h$. Likewise, the measurement $z_k^v$ is associated with vertex $v_k^h$. For each hypothesis, we maintain the mean of the measurements associated to each edge, which we denote ${\mu_e}_k^h$, as well as the mean of the measurements associated to each vertex, which we denote ${\mu_v}_k^h$. ${\mu_e}_k^h$ is indexed similarly to a neighbor list, e.g. ${\mu_e}_k^h(v_k^h, \alpha_k^h)$, and ${\mu_v}_k^h$ is indexed by the vertex, e.g. ${\mu_v}_k^h(v_k^h)$. Lastly, we keep track of the number of measurements associated to each edge with ${M_e}_k^h$ and each vertex with ${M_v}_k^h$. These are indexed in the same way as the means. This allows for a compact recursive computation for the likelihood update of Eq. 4,

$$p(z_k^e, z_k^v | z_{0:k-1}, X_k^{h'}, G_k^{h'}, u_{1:k}) \propto$$
$$\exp\left(-\frac{1}{2}(z_k^e - {\mu_e}_{k-1}^h(v_k^h, \alpha_k^h))^T {C_k^e}^{-1}(z_k^e - {\mu_e}_{k-1}^h(v_k^h, \alpha_k^h))\right)$$
$$\times \exp\left(-\frac{1}{2}(z_k^v - {\mu_v}_{k-1}^h(v_k^h))^T {C_k^v}^{-1}(z_k^v - {\mu_v}_{k-1}^h(v_k^h))\right) \quad (5)$$

In Eq. 5, the means ${\mu_e}_{k-1}^h$ and ${\mu_v}_{k-1}^e$ are acting as sufficient statistics for the history of sensor measurements $z_{0:k-1}$. The measurements are assumed to have additive zero mean white Gaussian noise with covariances $R_e$ and $R_v$ for the edge and vertex respectively. The following matrices are used in the computation of Eq. 5,

$$C_k^e = \left(1 + \frac{1}{{M_e}_{k-1}^h(v_k^h, \alpha_k^h)}\right) R_e \quad C_k^v = \left(1 + \frac{1}{{M_v}_{k-1}^h(v_k^h)}\right) R_v$$

To revisit the original problem, we would like to compute the posterior probability for each hypothesis when the robot traverses a new edge. To do this, we first update the likelihood function with Eq. 4 by loading the likelihood of the parent hypothesis and incorporating the new information with Eq. 5. The posterior is then easily computed by Eq. 3. Finally, the means ${\mu_e}_k^h$ and ${\mu_v}_k^h$ are updated for the next iteration.

### C. Prior Distribution

Neglected thus far in our discussion is the prior $p(G_k^h | u_{1:k})$ in Eq. 3. 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? There is no way to know the right answer.

But we can do better than a uniform distribution. Consider the following situation: a robot is circling a triangle topology, as in Fig. 4 (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

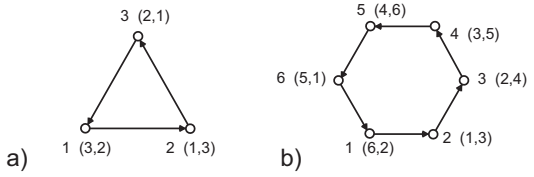


Fig. 4. This is an example of two different topologies that can result in a situation of perceptual aliasing. Both topologies fit the sensor data well.

fit the sensor data very well. On the other hand, the topology in Fig. 4 (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. We use the following distribution for experiments,

$$p(G_k^h | u_{1:k}) \propto \exp\left(-N_k^h \log k\right)$$

When two hypotheses have a similar likelihood, this prior will give preference to the smaller map. This makes sense, because we would like to prevent over-fitting. It turns out that this formulation is equivalent to using the Bayesian information criterion [14] for model selection. The Akaike information criterion [15] is related and is used in [12] with considerable success for a limited class of topologies.

By combining in Eq. 3 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.

## V. Pruning the Hypothesis Tree

The tree expansion algorithm described in Sec. III exhaustively considers all possible loop closures during an experiment. Therefore, even for a small map, the number of leaf hypotheses in the tree can grow very quickly (even to a size that is not computationally feasible). To keep the tree size bounded, we apply a series of pruning tests to the leaf hypotheses at each time step. This 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. We note that by eliminating hypotheses in this step, our approach is no longer Bayes optimal.

### A. Degree Test

In Alg. 1, when $L_{k-1}^h(v_{k-1}^h,\ \beta_k) = unexplored$, the hypothesis tree adds a child hypothesis for every possible loop closure to any vertex $v$ that also has an unexplored edge. If the detected degree of the arrival vertex, $\delta_k$, is unequal to the degree of vertex $v$, then that child hypothesis is immediately discarded. This is because the detected number of edges seen emanating from the new vertex should agree with what is expected for vertex $v$. This test involves no risk of eliminating the true hypothesis.

### B. Likelihood Update Test

When updating the likelihood for a new hypothesis recursively via Eq. 4, we observe whether

$p(z_k^e, z_k^v | z_{0:k-1}, X_k^{h'}, G_k^{h'})$ exceeds a 4-sigma error bound. If true, this would imply that the new measurements $z_k^e$ and/or $z_k^v$ do not agree with the measurements already associated to the corresponding edge/vertex and are therefore outliers in the data. This hints at an incorrect loop closure and thus the hypothesis is pruned. The test we use for pruning is when one of the following conditions is met,

$$(z_k^e - \mu_{e\,k-1}^{\;h}(v_k^h, \alpha_k^h))^T C_k^{e\,-1} (z_k^e - \mu_{e\,k-1}^{\;h}(v_k^h, \alpha_k^h)) > 16$$
$$(z_k^v - \mu_{v\,k-1}^{\;h}(v_k^h))^T C_k^{v\,-1} (z_k^v - \mu_{v\,k-1}^{\;h}(v_k^h)) > 16$$

This test has an extremely small but nevertheless non-zero chance of eliminating the true hypothesis.

### C. Planarity Test

As in [10], we use a strict test to eliminate hypotheses that are not planar. This test can often prune a large number of hypotheses without the risk of discarding the correct hypothesis. The specific planarity test algorithm that we use is related to [13] 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. 2 (b).

### D. Posterior Probability Test

Our last 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. A hypothesis is pruned when the following condition is met,

$$p(X_k^h, G_k^h | z_{0:k}, u_{1:k}) < \tau.$$

## VI. Algorithm Evaluation

The topology we use for experiments is based on the Voronoi diagram: the locus of points equidistant to two or more obstacles. Vertices correspond to points of three-way equidistance and edges correspond to paths between vertex locations. In Fig. 1, a floor plan of the sixth floor of Wean Hall at Carnegie Mellon University is depicted along with its corresponding Voronoi graph.

For all of our topological SLAM experiments, we use a two wheeled differential drive robot that has an array of sonar sensors. The robot can navigate from vertex to vertex in the generalized Voronoi graph (GVG) of an environment using sensor-based control. While traveling along an edge in the graph, the robot records a distance traveled measurement that corresponds to $z_k^e$ in the SLAM formulation of Sec. IV. While visiting a vertex in the graph, the robot records a range measurement to obstacles that corresponds to $z_k^v$ in the SLAM formulation of Sec. IV.

We recorded a library of data from real experiments that were performed in the map depicted in Fig. 1. During the experiments, we had the robot store the aforementioned measurements for each edge and each vertex over several trials. By creating this large library of measurement data, we can 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 obtained.

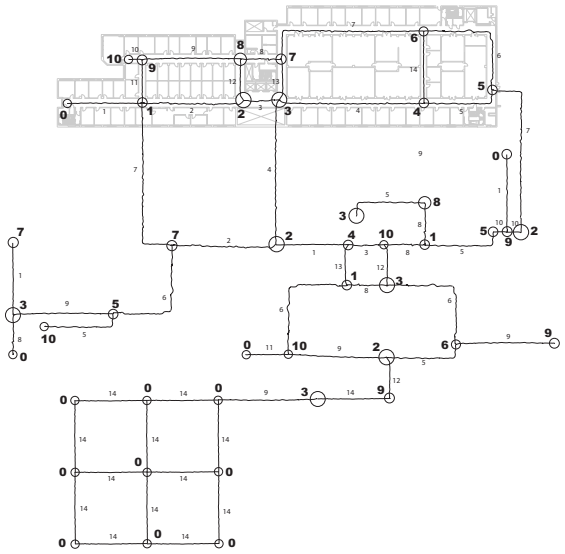

Fig. 5. This is a map created to simulate a much larger and more ambiguous version of Wean Hall at Carnegie Mellon University. Each number next to a vertex represents the corresponding true vertex from which this newly added vertex has been copied. Each number next to an edge represents the corresponding true edge from which this newly added edge has been copied.

We ran an experiment (Experiment 1) in which the robot performs 100 random edge traversals in the map depicted in Fig. 1. The experiment starts with the robot sitting at one of the vertices with no additional information. The tree expansion algorithm from Sec. III and the probability computations from Sec. IV are used to track multiple hypotheses of the map and robot state. For this environment, there are a number of ambiguities that make mapping difficult, namely vertices that share the same equidistance and edges that are the same length. Despite the ambiguities, the robot correctly maps this environment and localizes properly within the 100 edge traversals. At the end of the experiment, there is only one hypothesis that survives the pruning steps in Sec. V, and it is the correct hypothesis with the correct map.

We also ran another more challenging experiment (Experiment 2) that is based on a ground truth map that has a much larger amount of ambiguity. This example is used to demonstrate our algorithm's ability to handle the problem of perceptual aliasing. The graph we used for this experiment is shown in Fig. 5, and was made by adding a number of extra vertices and edges to the original floor plan of Fig. 1. 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 Experiment 2 in the map depicted in Fig. 5 with the robot performing 500 random edge traversals and fully exploring the map. The number of hypotheses tracked throughout the experiment is shown in Fig. 6 (a). In the

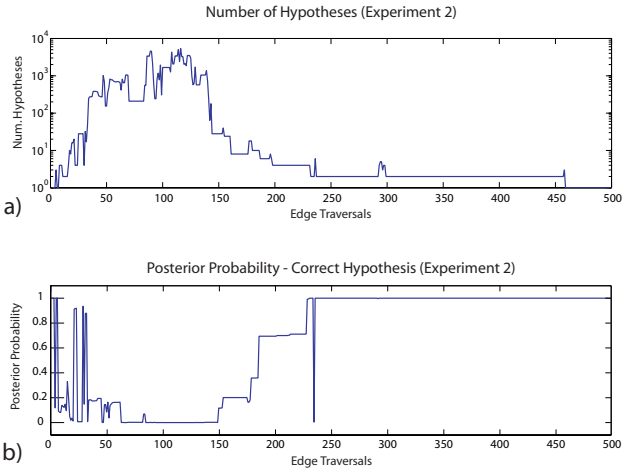


Fig. 6. (a) This is a plot of the number of hypotheses throughout Experiment 2. (b) This is a plot that shows the posterior probability tracked for the correct hypothesis during Experiment 2.

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. 6 (b), the posterior probability for the correct hypothesis is displayed 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 fits the data. Eventually though, as seen in Fig. 6, this other hypothesis becomes inconsistent and is discarded. The correct hypothesis then reemerges as a good candidate with a high probability.

We note that the implementations in [9], [10] 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.

## VII. Conclusion

The contribution of this work is to present a formal, probabilistic method for solving the topological graph loop closing problem. We introduce a tree expansion algorithm and a technique for recursively computing the posterior probabilities for hypotheses in the tree. This is a multi-hypothesis approach, and so it avoids the issue of committing to a false loop closure. Additionally, the posterior probability is properly defined to prevent over-fitting the data while helping to prune inconsistent hypotheses.

The experiments show the algorithms success in situations with perceptual aliasing. The second experiment is especially challenging because the map is large and contains numerous ambiguities. Despite a challenging experimental setup, the algorithm remarkably produces the correct mapping and localization hypothesis with a high level of confidence.

Throughout the paper, we assume the robot turns down the correct departure edge given a motion input $u_k$. This assumption can be made more general by adding a motion model to Eq. 3 according to the law of total probability,

$$p(X_k^h|G_k^h,u_{1:k}) = \sum_{X_{k-1}^h} p(X_k^h|X_{k-1}^h,G_k^h,u_k)p(X_{k-1}^h|G_k^h,u_{1:k-1})$$

The difference would be that for any given tree expansion, a leaf hypothesis would spawn more hypotheses (because the algorithm would consider the possibility that the robot has turned down an incorrect edge).

Despite the fact that our pruning rules are conservative by design, there is still a non-zero chance of eliminating the true hypothesis. If this happens, it is still possible for the algorithm to recover in the following way. The hypotheses that were not pruned are incorrect, and therefore will eventually prove inconsistent with the data. The algorithm will prune these hypotheses as well, leaving zero remaining hypotheses in the tree. In this case, the algorithm can revive the next best branch in the tree from a previous time step and replay the measurement sequence as if it were never pruned.

## References


[1] 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*, vol. 17, no. 3, pp. 229–241, June 2001.

[2] S. Thrun, Y. Liu, D. Koller, A. Ng, Z. Gharamani, and H. Durrant-Whyte, "Simultaneous localization and mapping with sparse extended information filters," *Intl. Journal of Robotics Research*, vol. 23, no. 7-8, pp. 693–716, 2004.

[3] A. Elfes, "Occupancy grids: A probabilistic framework for robot perception and navigation," *PhD thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University*, 1989.

[4] S. Thrun, W. Burgard, and D. Fox, "A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping," *In Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA 2000)*, vol. 1, pp. 321–328, 2000.

[5] H. Choset and K. Nagatani, "Topological simultaneous localization and mapping (SLAM): toward exact localization without explicit localization," *Robotics and Automation, IEEE Transactions on*, vol. 17, no. 2, pp. 125–137, Apr 2001.

[6] B. Lisien, D. Morales, D. Silver, G. Kantor, I. Rekleitis, and H. Choset, "The hierarchical atlas," *Robotics, IEEE Transactions on*, vol. 21, no. 3, pp. 473–481, June 2005.

[7] E. Remolina and B. Kuipers, "Towards a general theory of topological maps," *Artificial Intelligence*, vol. 152, no. 1, pp. 47–104, 2004.

[8] 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.

[9] 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*, pp. 1639–1645, 1993.

[10] 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*, pp. 1511–1517, 2004.

[11] E. Ranganathan, E. Menegatti, and F. Delleart, "Bayesian inference in the space of topological maps," *IEEE Trans. Robot. Autom.*, vol. 22, no. 1, pp. 92–107, February 2006.

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

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

[14] G. Schwarz, "Estimating the dimension of a model," *The Annals of Statistics*, vol. 6, pp. 461–464, 1978.

[15] H. Akaike, "A new look at the statistical model identification," *Automatic Control, IEEE Transactions on*, vol. 19, no. 6, pp. 716–723, Dec 1974.