Proceedings of the 2007 IEEE/RSJ International
Conference on Intelligent Robots and Systems
San Diego, CA, USA, Oct 29 - Nov 2, 2007

Hybrid Localization Using the Hierarchical Atlas

Stephen Tully, Hyungpil Moon, Deryck Morales, George Kantor, and Howie Choset

Abstract— **This paper presents a hybrid localization scheme for a mobile robot using the hierarchical atlas. The hierarchical atlas is a map that consists of a higher level topological graph with lower level feature-based metric submaps associated with the graph edges. Our method employs both a discrete Bayes filter and a Kalman filter to localize the robot in the map. This framework accommodates localization in a map with no prior information (global localization) and localization in a map with an incorrect pose estimate (kidnapped robot). Our approach efficiently scales to large environments without sacrificing accuracy or robustness. We have verified our method with large-scale experiments in a multi-floor office environment.**

I. Introduction

Accurate and reliable mobile robot localization is essential for tasks such as autonomous navigation and exploration. If a robot is lost or uncertain about its location in the environment, it must infer its pose based on exteroceptive information. For most localization techniques, this is accomplished by estimating a probability distribution over possible robot poses in a map. When one location hypothesis dominates the probability distribution, usually due to sufficiently informative measurements, localization is complete.

Existing solutions using metric maps accurately localize the robot to any position in the environment without any prior pose information. However, for a fine resolution, the computation and storage requirements can become intractable for large environments. On the other hand, methods that employ topological maps scale efficiently in storage and computation with environment size by limiting localization to topological node positions. Unfortunately, this gain in efficiency comes at the cost of detailed environment information.

We present a hybrid solution to mobile robot localization that couples a discrete Bayesian filter for localizing on a topological graph with a continuous Kalman filter for localizing metrically within submaps. Our method is implementable for real-time, large-scale, indoor applications and solves two different scenarios of localization: the global localization problem (when a robot is given a map but has no prior pose information) and the kidnapped robot problem (when a robot is given a map and currently maintains an incorrect estimate of its pose in the environment). Decomposing the environment into submaps, for this hybrid method, is based on the *hierarchical atlas* [1], a topological graph with feature-based metric submaps attached to the graph edges.

S. Tully is with the Electrical and Computer Engineering Department and H. Moon, D. Morales, G. Kantor, and H. Choset are with the Robotics Institute at Carnegie Mellon University, Pittsburgh, PA 15213, USA. {stully@ece, hyungpil@, deryck@andrew,kantor@ri,choset@cs}.cmu.edu

In this paper, we first derive our hybrid localization method and provide motivation for its use with the hierarchical atlas. Then, after providing details of the implementation, we present an extensive experiment that covers 6 floors of a large office building with multiple cases of topological and feature-based ambiguities.

II. Background

A. Localization Strategies

When addressing the task of localization, every approach must maintain and update a representation of the robot pose probability distribution. Many approaches define the state-space as the continuous spatial dimensions of the environment. A traditional technique for location estimation is to use Kalman filters. Due to the necessity for a multi-modal distribution when solving the robot localization problem (caused by observation ambiguities and initial location uncertainty), several approaches choose to allocate multiple Kalman filters: one for each initialized location hypothesis [2], [3]. These multi-hypothesis tracking algorithms do not provide a bound to the number of initialized hypotheses and often initialize additional filters for failed data associations. Also, due to odometric error in large-scale maps with limited features, false measurement associations may occur which could potentially result in the robot's false overconfidence in an incorrect pose hypothesis.

Markov localization [4] also attempts to localize the robot spatially in the environment. The method first discretizes the environment with some desired resolution, then assigns a probability to every state. 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 [4]. This method does not scale well for a fine resolution in a large environment. Monte Carlo localization [5] also uses an environmental discretization as its state space, but represents the probability distribution with a set of weighted representative samples of the state space, called particles. The update of M particles can be implemented with complexity $O(M)$ [5]. However, representing the initial uniform distribution of robot pose probability 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 robot pose, the number of particles increases with increased environment size.

Topological methods discretize the environment to a minimal number of abstract nodes in a graph. Thus, topological state space representations generally scale well with environment size. A common method is to use topological features to localize a robot to a node in the graph. Unfortunately,

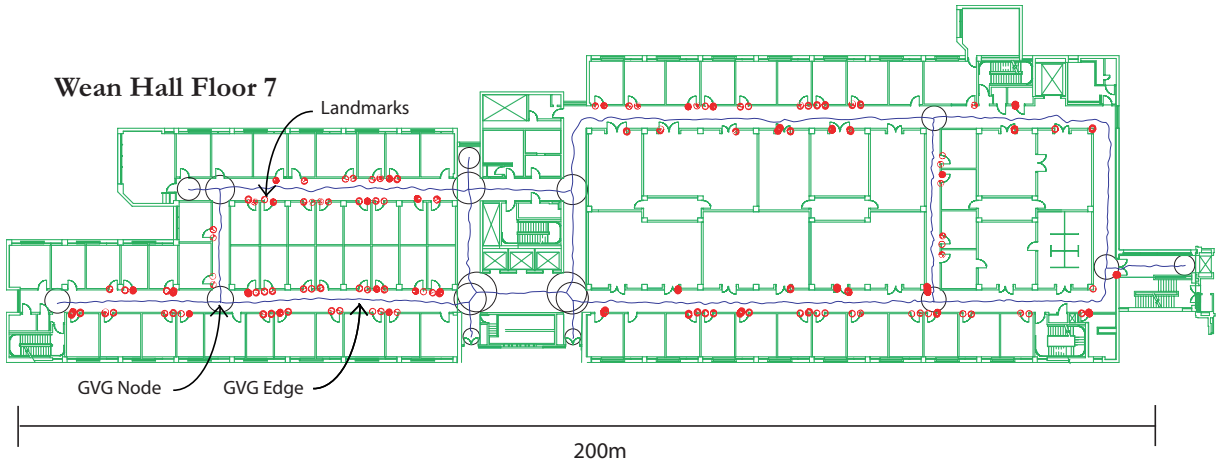

Fig. 1. One of six floors collectively mapped using the hierarchical atlas, which is embedded onto the floor plan of Wean Hall Floor 7 at Carnegie Mellon University. Nodes are shown with circles and landmarks mapped along edges are shown with red (dark) ellipses.

similarity in topological nodes is common, and additional information is necessary to resolve 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 [6]–[8], while others apply *graph-matching* [9]–[11], which uses observations of neighboring nodes to prune a set of candidate locations. The complexity of topological methods varies with the implementation.

B. Hierarchical Atlas

Originally presented to address simultaneous localization and mapping, the hierarchical atlas combines the scalability of a topological representation with the detail of feature-based metric maps. The atlas uses a topology to decompose the space into submaps which are tractable by conventional metric mapping methods. This allows for a high resolution representation of the free space while maintaining low computation and storage costs for localization and navigation.

The topology used by the hierarchical atlas is based on the generalized Voronoi graph (GVG) [9], the locus of points equidistant to two or more obstacles, whose nodes have a definite location in the free space and whose edges define obstacle-free paths between neighboring nodes. This topology is both abstract and embedded in the free space, and as such can be traversed using sensor-based control. Attached to each graph edge is a feature-based *edge map*. Both the topological decomposition of an environment and the corresponding edge maps can be seen in Fig. 1.

There have been other mapping strategies which decompose a large environment into submaps [12]–[14]; in fact the term "atlas" originates in [14]. However, the topology of these approaches are not necessarily navigable using sensor-based control, nor embedded in the free space by definition.

In [1], we introduce global localization for the hierarchical atlas. The robot maintains a list of candidate topological states and gathers information while traversing the graph. States that do not "match" the exteroceptive observations are pruned based on a naive yes/no heuristic until only one candidate remains. We have extended this work with the proposed hybrid solution to allow for accurate metric localization and, more importantly, to provide a robust Bayesian framework that, in our experiments, provides a significant improvement when handling map ambiguities and kidnapped robot situations.

III. Hybrid Localization

One contribution of this work is the following formulation of mobile robot localization as a multi-hypothesis estimation problem in a hybrid state space: a discrete component m_k identifies in which submap the robot currently resides among all submaps of a global map, and a continuous component X_t defines the relative metric pose of the robot in the coordinate frame of that submap. The time step, t, is a variable related to the elapsed time since the robot switched discrete states (or equivalently, when the robot transitioned between two submaps). Additionally, k is a discrete time step variable that increments when transitioning between submaps.

A. Separating the Task

To localize the robot, we must maintain a probability distribution over this hybrid state space, where the states with a higher probability correspond to likely robot locations. The probability of a hybrid state $[m_k, X_t]$ can be expressed as

$$p(m_k, X_t | u^{k-1}, v^{t-1}, z^k, y^t, \Theta), \tag{1}$$

where u^{k-1} is the collection of discrete motion inputs (u_0,u_1,...,u_{k-1}) that have been applied since the robot began an experiment. Each of these inputs causes the robot to change to a new discrete state and therefore increment the discrete timestep variable k. A *right* or *left* command at a hallway intersection that causes the robot to turn onto a different GVG edge (and therefore transition to a new submap) is an example of such a discrete motion input. Likewise, v^{t-1} is the collection of all continuous motion inputs (v_0,v_1,...,v_{t-1}) that are integrated while the robot is traveling within submap m_k. A vector of wheel velocities

or motor voltages is an example of such an input. The collection of measurement vectors is denoted by z^k, where (z_0,z_1,...,z_k) is obtained within submaps (m_0,m_1,...,m_k) respectively. Finally, y^t is the collection of measurements (y_0,y_1,...,y_t) observed while the robot travels within submap m_k. We note that a single submap measurement z_k will include all measurements y^t. The map used for localization is represented by Θ.

The joint probability in Eq. 1 can be manipulated through the definition of conditional probability to obtain

$$p(m_k, X_t|u^{k-1}, v^{t-1}, z^k, y^t, \Theta) = p(m_k|u^{k-1}, v^{t-1}, z^k, y^t, \Theta) \quad \times \\ p(X_t, |m_k, u^{k-1}, v^{t-1}, z^k, y^t, \Theta). \qquad (2)$$

The first term on the right side is a discrete probability distribution over possible submaps m_k. The second term is a continuous probability distribution over the relative metric pose of the robot given submap m_k. In this section, we will show that the first term can be estimated with a recursive Bayesian filter and the second term can be reduced to mobile robot tracking. This is similar to a multi-hypothesis tracking solution, except with one key advantage: the hypothesis size is bounded by the number of submaps in the environment. In many cases, this will provide a significant computational advantage over other methods.

B. Discrete Bayesian Submap Filtering

We have shown that estimating the probability distribution over possible hybrid robot states can be separated into two tasks. The first task is to estimate a discrete probability distribution over possible submaps. When one value in this distribution approaches one, we can be confident that the robot is currently positioned somewhere in that corresponding submap. The discrete distribution component of Eq. 2 can be simplified to

$$p(m_k|u^{k-1}, z^k, \Theta). \qquad (3)$$

We will leave out v^{t-1} because this term refers to motion inputs within submap m_k, not motion inputs that influence discrete state transitions. We are not necessarily assuming independence by dropping this term, though. Any information inferred by v^{t-1} that is correlated with m_k can be included as a measurement in z_k. For example, we later compute a path length statistic from the motion inputs v^{t-1} and include this as a measurement when updating the discrete probability distribution over m_k. Additionally, we can omit y^t because it is subsumed by z^k. Using Bayes law, Markov independence, and the law of total probability, we can rewrite Eq. 3 as follows:

$$p(m_k|u^{k-1}, z^k, \Theta) = \frac{p(z_k|m_k, \Theta)}{p(z_k|u^{k-1}, z^{k-1}, \Theta)} \quad \times \\ \sum_{m_{k-1}} p(m_k|m_{k-1}, u_{k-1}, \Theta) p(m_{k-1}|u^{k-2}, z^{k-1}, \Theta).$$

This produces a recursive update rule for the discrete probability distribution, which is now in the form of a measurement model, a motion model, and a prior. m_{k-1} is the submap in which the robot was previously located. It is important to note that the motion model we adopt for this recursive update rule involves the probability of making a discrete transition between two submaps m_{k-1} and m_k. For example, when using the hierarchical atlas, an appropriate motion model may represent the probability of turning to the correct GVG edge. This is unlike conventional motion models in which metric location uncertainty is propagated and dispersed to accommodate for odometric error.

C. Relative Metric Pose Estimation

The next task is to estimate a continuous probability distribution for the relative metric pose X_t within submap m_k. The corresponding term in Eq. 2 can be simplified to

$$p(X_t|m_k, v^{t-1}, y^t, \Theta). \qquad (4)$$

One assumption that we make is that a mobile robot, when making a transition from submap m_{k-1} to submap m_k, will be located at the origin of the latter when the transition occurs. Although not common for all submap techniques, this is the case when adopting the hierarchical atlas. This is because all submaps in the hierarchical atlas are defined between GVG nodes, so the termination and beginning of submaps always coincide. We can ignore u^{k-1} in Eq. 4 because these motion inputs are applied at the boundaries of submaps and therefore have no bearing on the estimation of the relative pose X_t within the submap. We can also ignore z^k because all previous measurements have no bearing on the relative pose estimate if it is known exactly at $t = 0$ and all relevant current measurements in the vector z_k are present in y^t.

In our case, knowing the relative pose exactly when the robot makes a discrete transition between two submaps provides the following key advantage: $p(X_t|m_k, v^{t-1}, y_t, \Theta)$ can be estimated with a tracking based technique throughout the time the robot is located within the submap. Tracking can be solved efficiently using a unimodal filter, such as a Kalman filter. It can now be seen that our estimation technique is related to multi-hypothesis tracking, with the advantage that we must track at most one estimate per submap. This is useful when the number of submaps is small relative to the area of a given map.

IV. Probabilistic Models for Localization

In this section we will define the state space, motion model, and measurement models that are used when specifically applying the hybrid localization technique to the hierarchical atlas.

A. State Space

We define the discrete state and continuous metric state as

$$m_k = \begin{bmatrix} n_k \\ e_k \end{bmatrix} \quad \text{and} \quad X_t = \begin{bmatrix} x_t \\ y_t \\ \phi_t. \end{bmatrix}$$

respectively. Index, n_k, is the GVG node to which the robot is heading and e_k is the index of the edge the robot is traversing

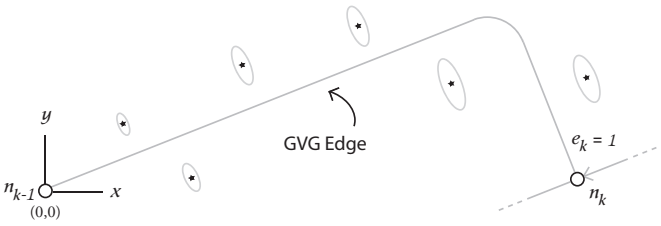

Fig. 2. A typical submap with appropriate labels to demonstrate the state space used for localization.

(relative to a reference edge of node n_k). The origin of the submap is defined by the location of node n_{k-1}, and the coordinate frame of the submap is defined so as to align the locations of both the departure and arriving nodes on the x-axis. The robot pose in the submap is denoted by x_t, y_t, and ϕ_t all in the local coordinate frame.

Fig. 2 depicts one possible submap and demonstrates the specific state space representation used for this localization method. We note that e_k is equal to one for this example because it is offset by one in the counter-clockwise direction from the reference edge (drawn with an arrow).

B. Measurement Model

The measurement z_k is a vector of all exteroceptive information gathered during timestep k that can help distinguish candidate submaps. The specific measurement vector used for this paper is composed of the following discrete and continuous measurements:

$$\begin{aligned} \hat{\delta}_k &\rightarrow \text{Node Degree} \\ \hat{\varepsilon}_k &\rightarrow \text{Node Equidistance} \\ \hat{d}_k &\rightarrow \text{Edge Travel Distance} \\ \hat{f}_k &\rightarrow \text{Feature Map Landmark Locations} \end{aligned}$$

Each of these terms will be precisely defined in the subsections that follow.

The measurement model is defined:

$$\begin{aligned} p(z_k|m_k,\Theta) &= p(\hat{\delta}_k,\hat{\varepsilon}_k,\hat{d}_k,\hat{f}_k|m_k,\Theta) \\ &= p(\hat{\delta}_k|m_k)p(\hat{\varepsilon}_k|m_k)p(\hat{d}_k,\hat{f}_k|m_k). \quad (5) \end{aligned}$$

The node degree and equidistance measurements are independent and therefore can be separated into the first two terms of Eq. 5. The feature map landmark locations, on the other hand, are correlated with robot motion, and therefore correlated with the travel distance. We will now define each component of the measurement model and their corresponding probability distributions:

1) Node degree: The number of edges connecting a node to its neighbors is referred to as its degree. The measured degree $\hat{\delta}_k$ at timestep k is invariant over repeated visitations to that node, so we define $p(\hat{\delta}_k|m_k)$ accordingly, using empirical data.

2) Node equidistance value: Nodes in the GVG are defined as equidistant to three or more obstacles. Thus, when the robot arrives at node n_k, the distance to its three or more closest obstacles approaches equality. The mean of these values is labeled the *equidistance value*. It is important to note that the equidistance measurement at each node is stable over repeated visitations. The hierarchical atlas stores, for every node, an equidistance value and its standard deviation, which together represent a Gaussian distribution. During localization, an equidistance $\hat{\varepsilon}_k$ is measured when the robot arrives at a node. The probability of observing $\hat{\varepsilon}_k$ given that the robot has arrived at node n_k is

$$p(\hat{\varepsilon}_k|n_k) = \frac{1}{\sqrt{2\pi(\sigma^2_{k_\varepsilon})}} \exp\left(-\frac{(\hat{\varepsilon}_k-\varepsilon_k)^2}{2(\sigma^2_{k_\varepsilon})}\right).$$

3) Feature Map / Travel Distance: The robot constructs a metric feature map while traversing an edge. The map is built using an extended Kalman filter whose state $\hat{s}_k$ consists of the robot pose, the locations of measured landmarks $\hat{f}_k$, and the total distance traveled $\hat{d}_k$ while traversing the edge. The filter also produces a covariance matrix P for the measured feature map. The probability of measuring $\hat{d}_k$ and $\hat{f}_k$ given the traversal of submap m_k is

$$p(\hat{d}_k,\hat{f}_k|m_k) = \frac{\exp\left(-\frac{1}{2}(\hat{s}_k-s_k)^T(P)^{-1}(\hat{s}_k-s_k)\right)}{(2\pi)^{2N}|P|^{\frac{1}{2}}}.$$

C. Motion Model

When the robot arrives at node n_{k-1} via edge e_{k-1}, it must choose a new edge to traverse. We define u_{k-1} as a command for the robot to take a specific departing edge relative to the arriving edge e_{k-1} of node n_{k-1}. Therefore, u_{k-1} defines the desired submap transition.

Let $m_{k-1} \overset{u_{k-1}}{\rightarrow} m_k$ indicate that submap m_k neighbors m_{k-1} via the departing edge u_{k-1}, and let $m_{k-1} \overset{u_{k-1}}{\not\rightarrow} m_k$ indicate that submap m_k neighbors m_{k-1} via a departing edge unequal to u_{k-1}. The motion model we use for discrete submap localization is

$$p(m_k|m_{k-1},u_{k-1},\Theta) = \begin{cases} p(u_{k-1}) & \text{if } m_{k-1} \overset{u_{k-1}}{\rightarrow} m_k \\ \frac{1-p(u_{k-1})}{\delta_{k-1}-1} & \text{if } m_{k-1} \overset{u_{k-1}}{\not\rightarrow} m_k \\ 0 & \text{otherwise.} \end{cases}$$

This represents that the robot traverses the selected edge index u_{k-1} with probability $p(u_{k-1})$, and the remaining $1-p(u_{k-1})$ probability is evenly distributed among all other edges of the state m_{k-1}. This is to account for erroneous departures for structurally complicated nodes. In normal indoor environments with orthogonal edges, $p(u_{k-1})$ approaches 1.0, which means the robot almost always drives down the correct edge after leaving a node. For states m_k that do not neighbor m_{k-1}, the probability $p(m_k|m_{k-1},u_{k-1},\Theta)$ is zero. This does not imply that we are ignoring the chance of passing through a GVG node. Instead, we claim that such an occurrence is a special case of the kidnapped robot problem. In fact, the robot accidentally kidnaps itself. Thus, solving the kidnapped robot problem will be sufficient, which is discussed in Section V-C

V. Localization on the Hierarchical Atlas

A. Global Localization

Global Localization is performed by initializing the discrete state space with a uniform probability distribution.

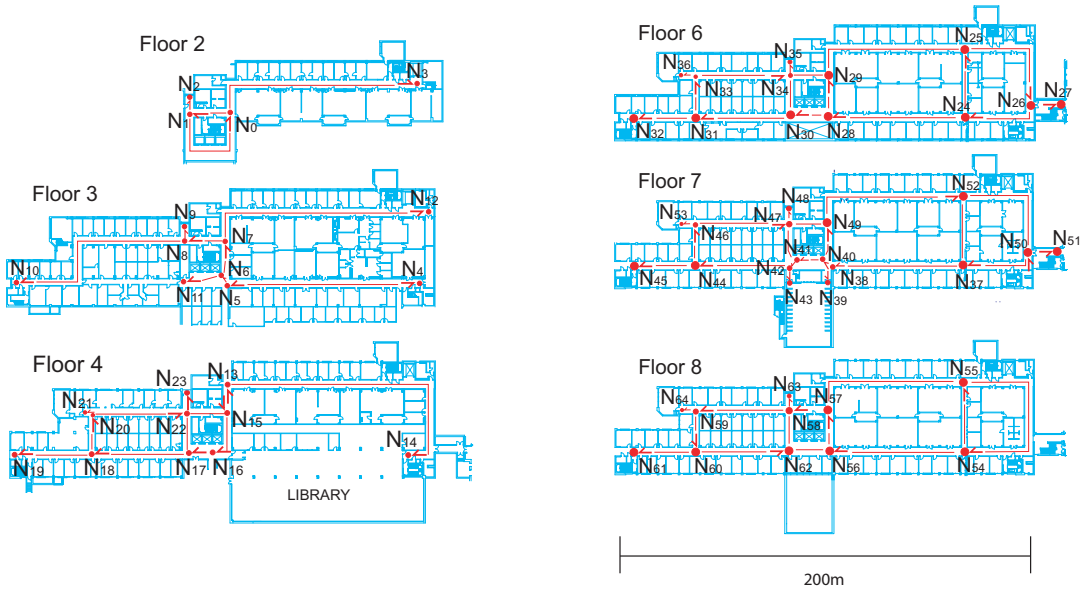

Fig. 3. Six floors of Wean Hall at Carnegie Mellon University were used for localization experiments. Here, the node/edge topological representation is embedded in a sequence of floor plan images. Arrows designate the reference edge for each node..

The robot starts in an unknown place in a priorly mapped environment and is told to run an algorithm so that the robot is autonomously traversing GVG edges. When the robot homes into its first node, it can apply a measurement update (except in this case there is no feature map to compare, so that component is not included). Then the robot chooses a departing edge and propagates the motion model. The instant the mobile robot departs the GVG node and begins traversing a new edge, the robot has started exploring a new submap and has begun traveling a path of two-way equidistance toward another node in the map.

During the edge traversal, the robot estimates the relative metric pose $p(X_t|m_k, v^{t-1}, y^t, \Theta)$ using a conventional Kalman filter based localization method. This filtering task is inexpensive because the size of the Kalman filter state is limited to the degrees of freedom of the robot. On the other hand, the robot must maintain a bank of these Kalman filters to estimate the relative metric location: one per submap. If this turns into a large computational burden for numerous submaps, the robot should simply wait until the discrete distribution is better resolved, and then only estimate the relative metric pose for likely submaps.

While the robot estimates its relative pose in the submap, it is simultaneously building a feature map using conventional EKF SLAM. This is independent of the relative pose estimation, and is only for incorporating metric information into the discrete submap update, as discussed in Sec. IV-B.3.

When the robot arrives at the destination node, it can apply a full discrete measurement update (using the measured node equidistance, node degree, and the feature map built using EKF SLAM during the edge traversal). When the next motion model is applied, the continuous tracking filters can be discarded and reinitialized at the origins of each submap. The robot continues traversing the GVG until a majority of the probability distribution is concentrated to one topological state. At this point, the continuous tracking filter associated with that submap will provide the most likely metric location.

Algorithm 1 Hierarchical Atlas Hybrid Localization

```
InitializeUniform()
for k ← 1 to T do
    TraverseEdge(u_{k-1})
    for all m_k do
        p(m_k) ← Σ_{m_{k-1}} p(m_k|u_{k-1}, m_{k-1}) p(m_{k-1})
    end for
    p(z_k) ← 0
    for all m_k do
        if p(m_k) < γ, PruneState(m_k)
        else p(z_k) ← p(z_k) + p(δ_k, ε_k, d_k, f_k|m_k) p(m_k)
             p(m_k) ← p(δ_k, ε_k, d_k, f_k|m_k) p(m_k)
    end for
    if p(z_k) < ε, InitializeUniform()
    else Normalize()
end for
```

B. Reducing the State Space (Hypothesis Pruning)

The number of discrete states in the discrete probability distribution is determined by the number of submaps in the global map. Therefore, the computational complexity will scale linearly with the number of submaps. One way to reduce computation is to prune out hypotheses from the state space if they have a negligible probability. Eliminating one or more submaps from the state space can often remove a significant region of the global map, narrowing the search.

$$\text{if } p(m_k|u^{k-1}, z^k, \Theta) < \gamma, \text{ remove } m_k$$

This pruning method is an approximation and can be risky because no matter how unlikely a hypothesis may become,

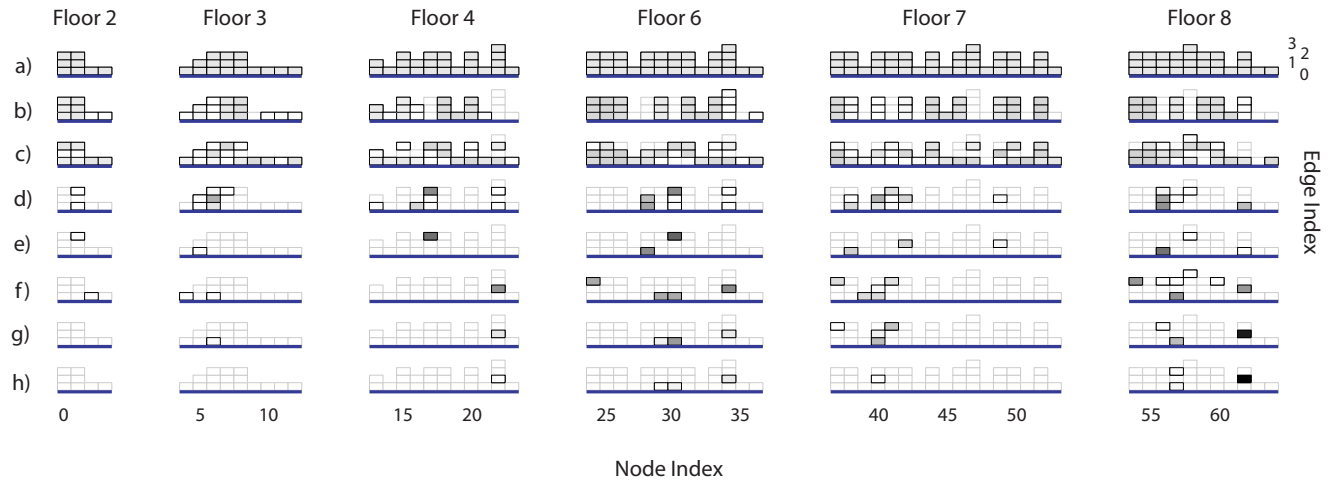

Fig. 4. A visual representation of the discrete probability distribution during one trial of global localization. Darker bins correspond to states with higher probability. Bins that appear faint represent improbable hypotheses that have been pruned from the state space. Each stage represents the following: (a) initial distribution, (b) initial node update, (c) motion model, (d) second node update, (e) edge update, (f) motion model, (g) node update, and (h) edge update.

there will still be a non-zero chance that the robot is located in that submap. By refusing to apply the Bayesian update for eliminated states, there is no way for the state to regain any portion of the probability distribution (even when matching well to map measurements). We argue that this situation is a special case of the kidnapped robot problem: the robot has a false estimate of its location (in one of the states that have not been pruned), when in reality it is located somewhere else (in one of the pruned states). Therefore, we will choose to incorporate pruning due to its computational advantages, but provide a solution to the kidnapped robot problem that encompasses the issues associated with removing hypotheses from the Bayesian update. The algorithm for updating the discrete probability distribution, which includes pruning, is outlined in Alg. 1.

C. The Kidnapped Robot Problem

The kidnapped robot problem is when the probability distribution is biased toward one or more incorrect hypotheses. This may happen if there is an unmodeled motion disturbance: for example, the robot could be physically picked up and moved to another location, the robot may skip over a topological node, or the robot may be located in a pruned state. To detect such an occurrence, we monitor $p(z_k)$, which is the the probability of making a measurement z_k at timestep k. This can be computed as follows (using the law of total probability):

$$p(z_k) = \sum_{m_k} p(z_k|m_k)\ p(m_k).$$

A small $p(z_k)$ value suggests that new measurements do not agree with any of the submap hypotheses. Therefore, if $p(z_k)$ falls below a threshold, we will conclude that the robot has been "kidnapped". At this point, the probability distribution over possible submaps is reset to a discrete uniform distribution, and the global localization algorithm is restarted. The detection of a kidnapped robot is shown in Alg. 1 on line 13.

VI. Experimental Results

Our localization experiments span six floors of a large office building environment (Wean Hall at Carnegie Mellon University), see Fig. 3. This map spans an area of size greater than 20,000 square meters and contains numerous topological and feature map ambiguities, resulting in a rigorous test for localization algorithms. It should be noted that the entire map was constructed online prior to any localization experiments using the hierarchical SLAM method in [1].

Our experimental platform uses an ultrasonic sensor array, the arc transversal median method [15], and sensor-based control to accurately traverse the GVG of an environment. An omnidirectional camera is used to detect features in the environment by comparing acquired SIFT keypoints [16] to a training set of descriptors associated with specific visual landmarks (primarily doorways). The location of these features are what constitute the local feature maps that are associated with edges in the topology. The feature maps are constructed using conventional Kalman filter SLAM, with resulting landmark locations for one of the six floors shown in Fig. 1.

In this section, we will discuss one global localization experiment and one kidnapped robot experiment to demonstrate the localization process for these two scenarios. We will then provide statistics regarding the performance of the algorithm over many trials to provide a basis for the robustness of this technique. We note that our analysis will focus on achieving topological localization. This is because our relative metric localization is a form of mobile robot tracking, which is a well studied technique with well known results [17].

A. Global Localization Experiment

Here we describe a global localization experiment and the result obtained when performing this experiment in the

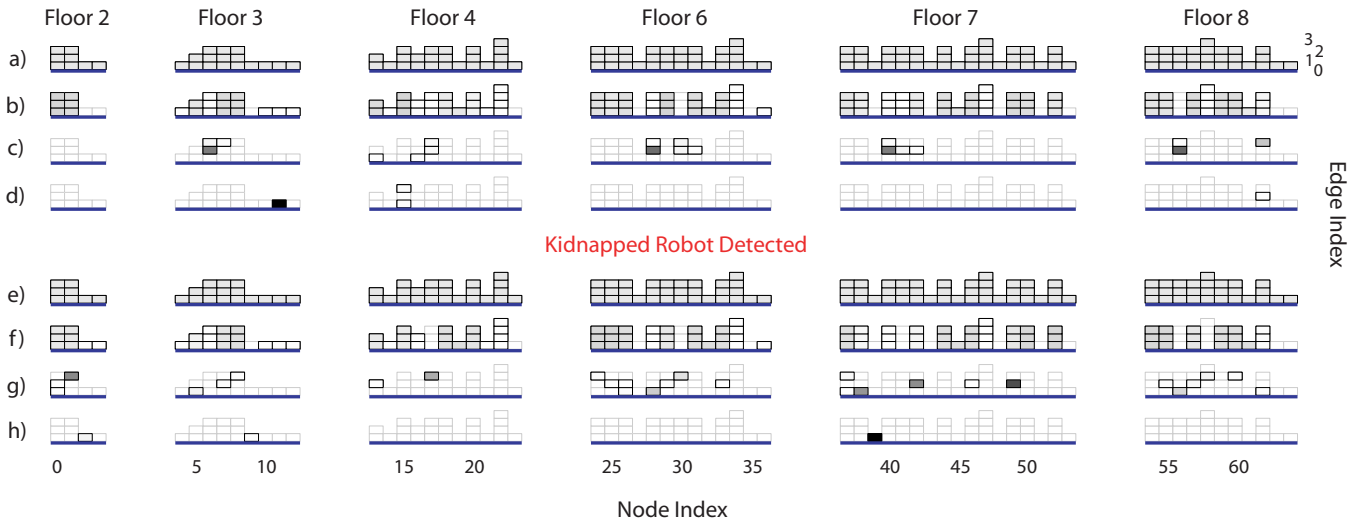

Fig. 5. A visual representation of the discrete probability distribution during one trial of kidnapped localization. Darker bins correspond to states with higher probability. Bins that appear faint represent improbable hypotheses that have been pruned from the state space. Each stage represents the following: (a) initial distribution, (b) first measurement update, (c) second measurement update, (d) third measurement update, (e) reinitialization, (f) first measurement update, (g) second measurement update, and (h) third measurement update.

provided map. If placed randomly between node 54 and 55 in the map (see Fig. 3), the robot accesses the GVG edge and drives toward node 54 ($m_0 = [54, 1]$). At this point, the submap distribution is uniform. In Fig. 4, each bin represents the probability of each state m_k over different stages in the algorithm, and the uniform distribution appears in (a).

The robot arrives at node 54 and updates the state distribution with the equidistance and node degree measurements. At this point, the robot maintains a higher probability for submaps with an arriving node of degree 3 and an average equidistance (see Fig. 4 (b)). The robot departs the node onto one of the edges, which changes the current submap. Therefore, the motion model is propagated (see Fig. 4 (c)).

The robot then completes traversing the edge and has built a local feature map of the traveled edge. The robot arrives at $m_1 = [56, 0]$, and updates the state probability distribution with the measurement model. Here, we separate the effect of applying each component of the measurement model to demonstrate the effect of each measurement on the probability distribution. In Fig. 4 (d) we update with node measurements at node 56, and in (e) we update with the edge travel distance and feature map. Note that the edge test eliminates many improbable discrete states, but is not enough to complete localization due to the ambiguity of the map.

Notice that at step (e), the majority of discrete states have a very low probability because measurements up until this point did not agree with the corresponding states. We prune these states and only update the states that remain. The robot has not yet localized completely, though, so it departs node 56 and repeats the same process for a new edge/node. Fig. 4 (f) shows that some states are revived after propagating the motion model when departing. Fig. 4 (g) and (h) depict the result of the node measurement update and the edge measurement updates (travel distance and feature map), respectively. The robot finally completes the localization, figuring out that it has arrived at node 62 from edge 1 (state $[62, 1]$) with a probability near 99%.

B. Kidnapped Robot Experiment

The following experiment, represented by Fig. 5, is one of many successful kidnapped robot trials we obtained during experiments (see Table I). The robot initializes the state space with a uniform probability distribution and travels from node 7 to 6 and then to 11 (Fig. 5 (a)-(d)). This provides enough information to successfully localize the robot to state $m_2 = [11, 0]$ with 98% probability. When the robot begins to travel back towards node 6, we physically relocate the robot to another floor (state $m_3 = [37, 0]$). The robot homes into node 37 and recognizes that the travel distance and equidistance measurements obtained are improbable given the expected arrival node. The robot successfully concludes that it is kidnapped and resets the localization process with a uniform distribution. The robot later localizes successfully at node 39 with new information, see Fig. 5 (e)-(h).

C. Overall Experimental Results

The following results were obtained when testing the localization algorithm on the robot at random initialized locations in the environment.

TABLE I
ROBOT LOCALIZATION RESULTS

	# of trials	# of successes	# of failures
Global localization	30	29	1
Kidnapped Robot	20	20	0
Total	50	49	1

In Sec. V-C, we briefly discuss the case where a mobile robot skips over a topological node while traversing an edge, therefore producing a submap transition that does not follow the provided motion model. Adopting a more appropriate

motion model is one solution to this issue, but generating sufficient data on which to base the motion model parameters would be a tedious operation because, in our experiments, the robot rarely skips over a topological node. Instead, we claim that such an occurrence can be treated as a special case of the kidnapped robot problem. In several trials, we forced the robot to skip over a topological node by blocking an entrance to one of the hallways of the node the robot was approaching. For example, in one trial, the robot traveled from node 14 to 13 to 15 and correctly hypothesized with 99% confidence that it was located at $[15, 0]$. When the robot started moving towards node 16, we blocked the left turn (leading out of the map), which caused node 16 to appear like a region of two-way equidistance. The robot skipped the node and later homed into node 17. The recorded path length and equidistance measure were not what the robot expected, and it recognized that it must have been kidnapped. The robot reset the state space with a uniform distribution and later localized sucessfully at node 21.

In Table I, we refer to a global localization failure. For this specific failure, the robot believed a node existed which was not in the map. We believe this can happen for unstable meetpoints, as described in [9]. After homing into the false node location, the robot performed a recursive update for the discrete submap distribution. Normally, this would cause the robot to detect a kidnapped situation and reset the localization algorithm, but unluckily, the measurements for the false node happened to agree with a different hypothesis on another floor. The robot claimed that it localized at that hypothesis with a high probability, and therefore the experiment was unsuccessful. On the other hand, we believe that additional edge traversals would have detected the incorrect hypothesis, restarting the localization algorithm as if it were a kidnapped robot situation. This is the only failure we encountered among the 50 total localization experiments.

VII. Conclusion

This paper presents a hybrid localization scheme that scales well to large environments and robustly solves both the global localization problem and the kidnapped robot problem. Localization is separated into two tasks: estimating a discrete probability distribution over possible submaps and a continuous distribution over the relative pose within each submap.

We show that the discrete distribution can be incrementally updated using a recursive Bayesian filter, and when decomposing a map based on the hierarchical atlas, the estimation of the continuous relative pose within a submap can be reduced to mobile robot tracking (which is implementable with a unimodal filter). This makes our method similar to multi-hypothesis tracking, except with a key advantage: the number of hypotheses is limited to the number of submaps in the environment, which in our experience is small in relation to the area of the free-space. In addition, our tracking filters are reset to known node locations in the map when performing a discrete state transition. This can correct hypotheses that have been affected by unmodeled disturbances.

We have performed experiments in a map that spans 6 floors of an office environment and covers a space of 20,000 square meters. Despite having such a large map, with many topological and feature-based ambiguities, we have shown that our hybrid method successfully and efficiently localizes a robot topologically and metrically in real-time for both the global localization problem and the kidnapped robot problem.

Acknowledgments

The authors would like to thank David Lowe for providing SIFT source code, Wolfram Burgard for his advice, and Aaron Hoy for his assistance with experiments.

References

[1] B. Lisien, D. Morales, D. Silver, G. Kantor, I. Rekleitis, and H. Choset, "The hierarchical atlas," *IEEE Transactions on Robotics*, vol. 21, pp. 473 – 481, June 2005.

[2] P. Jensfelt and S. Kristensen, "Active global localization for a mobile robot using multiple hypothesis tracking," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 5, October 2001.

[3] 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.

[4] D. Fox, W. Burgard, and S. Thrun, "Markov localization for mobile robots in dynamic environments," *Journal of Artificial Intelligence Research*, vol. 11, pp. 391–427, 1999.

[5] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust monte carlo localization for mobile robots," *Artificial Intelligence*, vol. 128, no. 1-2, pp. 99–141, 2000.

[6] I. Nourbakhsh, R. Powers, and S. Birchfield, "Dervish: An office-navigating robot," *AI Magazine*, vol. 16, no. 2, pp. 53–60, 1995.

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

[8] 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.

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

[10] B. Kuipers and Y.-T. Byun, "A robot exploration and mapping strategy based on a semantic hierachy of spatial representations," *Robotics and Autonomous Systems*, vol. 8, pp. 46–63, 1991.

[11] G. Dudek, P. Freedman, and S. Hadjres, "Using multiple models for environmental mapping," *Journal of Robotic Systems*, vol. 13, no. 8, pp. 539–559, 1996.

[12] K. S. Chong and L. Kleeman, "Large scale sonarray mapping using multiple connected local maps," in *Proceedings of the International Conference on Field and Service Robotics*, December 1997, pp. 538–545.

[13] S. Simhon and G. Dudek, "A global topological map formed by local metric maps," in *Proceedings of the International Conference on Intelligent Robots and Systems*, vol. 3, Victoria, Canada, October 1998, pp. 1708–1714.

[14] M. Bosse, P. M. Newman, J. J. Leonard, and S. Teller, "An atlas framework for scalable mapping," in *Proceedings of the International Conference on Robotics and Automation.*, September 2003.

[15] H. Choset, K. Nagatani, and N. A. Lazar, "The arc-transversal median algorithm: A geometric approach to increasing ultrasonic sensor azimuth," *IEEE Transactions on Robotics and Automation*, vol. 19, pp. 513–522, 2003.

[16] D. Lowe, "Distinctive image features from scale invariant features," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[17] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.