The relative orientations are

$$R_1 = \begin{bmatrix} 0.9563 & -0.0928 & -0.2774 \\ -0.2761 & 0.0268 & -0.9608 \\ 0.0966 & 0.9953 & 0 \end{bmatrix}$$
$$R_2 = \begin{bmatrix} 0.7951 & 0.3082 & 0.5222 \\ 0.4869 & 0.1888 & -0.8528 \\ -0.3614 & 0.9324 & 0 \end{bmatrix}$$
$$R_3 = \begin{bmatrix} -0.9876 & -0.0271 & -0.1544 \\ -0.1543 & -0.0042 & 0.9880 \\ -0.0274 & 0.9996 & 0 \end{bmatrix}$$
$$R_4 = \begin{bmatrix} -0.9876 & 0.0271 & 0.1544 \\ 0.1543 & -0.0042 & 0.9880 \\ 0.0274 & 0.9996 & 0 \end{bmatrix}.$$

With the CPU times of 4.7 ms and 175.8 ms, we have

$$P = [-0.0080\ 0.0023\ -0.0030\ -0.0449\ 0.0220\ -0.0359]^T$$
$$z_{\max}(-P) = 0.5159 < 1.$$

Hence, the grasp is force closure.

V. Conclusion and Future Work

A shortcut is found to simplify Liu's ray-shooting based algorithm for force-closure test [4]. The optimal objective value of the LP problem (2) with respect to $t = -P$ is the ratio of d_2 to d_1; that is, $z_{\max}(-P) = d_2/d_1$. If $z_{\max}(-P) < 1$, then the grasp is force closure; otherwise, it is not. Consequently, we can skip the steps of computing Q, d_1, and d_2. Having the geometric insight into the maximum $z_{\max}(-P)$, we can apply it to optimal grasp planning as a force-closure index. As this work goes beyond the topic of the paper, it is decent to leave it for the future.

Acknowledgment

The authors wish to thank the editors and the reviewers for fair ratings of this paper and helpful suggestions for revision.

References

[1] J. K. Salisbury and B. Roth, "Kinematic and force analysis of articulated hands," *ASME J. Mech. Transm. Autom. Design*, vol. 105, no. 1, pp. 35–41, Mar. 1983.

[2] B. Mishra, J. T. Schwarz, and M. Sharir, "On the existence and synthesis of multifingered positive grips," *Algorithm.*, vol. 2, no. 4, pp. 541–558, 1987.

[3] R. M. Murray, Z. X. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL: CRC, 1994.

[4] Y.-H. Liu, "Qualitative test and force optimization of 3-D frictional form-closure grasps using linear programming," *IEEE Trans. Robot. Autom.*, vol. 15, no. 1, pp. 163–173, Feb. 1999.

[5] D. Ding, Y.-H. Liu, Y. Wang, and S. G. Wang, "Automatic selection of fixturing surfaces and fixturing points for polyhedral workpieces," *IEEE Trans. Robot. Autom.*, vol. 17, no. 6, pp. 833–841, Dec. 2001.

[6] S. R. Lay, *Convex Sets and Their Application*. New York: Wiley, 1982.

The Hierarchical Atlas

Brad Lisien, Deryck Morales, David Silver, George Kantor, Ioannis Rekleitis, and Howie Choset

***Abstract*—This paper presents a new map specifically designed for robots operating in large environments and possibly in higher dimensions. We call this map the *hierarchical atlas* because it is a multilevel and multiresolution representation. For this paper, the hierarchical atlas has two levels: at the highest level there is a topological map that organizes the free space into submaps at the lower level. The lower-level submaps are simply a collection of features. The hierarchical atlas allows us to perform calculations and run estimation techniques, such as Kalman filtering, in local areas without having to correlate and associate data for the entire map. This provides a means to explore and map large environments in the presence of uncertainty with a process named *hierarchical simultaneous localization and mapping*. As well as organizing information of the free space, the map also induces well-defined sensor-based control laws and a provably complete policy to explore unknown regions. The resulting map is also useful for other tasks such as navigation, obstacle avoidance, and global localization. Experimental results are presented showing successful map building and subsequent use of the map in large-scale spaces.**

***Index Terms*—Concurrent mapping and localization, generalized Voronoi diagram, Kalman filtering, mobile robots, simultaneous localization and mapping (SLAM), topological maps.**

I. Introduction

This paper presents a new map organization for mobile robots which embodies scalability in both storage and computation to address common robot tasks in large-scale environments. These tasks include simultaneous localization and mapping (SLAM), path planning, global localization, and obstacle avoidance in nonstatic environments. This paper addresses each of these tasks, and presents experimental results obtained with a mobile robot in a large environment containing cycles, to show how the new map is well-suited to address these tasks.

The successful implementation of these tasks depends on a reliable and usable map. With the choice of three basic types of maps, topological, grid-based, and feature-based, it seems that one must settle for drawbacks inherent in each type in order to take advantage of its particular benefits. Topological maps scale nicely to large planar environments and to environments of higher dimension by storing a minimal amount of information. Such a minimalistic representation lacks the necessary information to localize arbitrarily (can only localize to nodes in the topological graph) and to disambiguate similar topological regions.

Grid-based approaches offer discretized renditions of unstructured free spaces which can be used for many robot tasks. However, the high resolution required for accurate representations demands large amounts of memory to store and computation time to maintain. Feature-based methods extract distinct landmark features from the environment for use in robot localization, but do not explicitly address obstacles unless the obstacles have structured, observable characteristics. Feature- and

Manuscript received December 16, 2003; revised July 20, 2004. This paper was recommended for publication by Associate Editor N. Amato and Editor F. C. Park upon evaluation of the reviewers' comments. This work was supported by the Office of Naval Research under Award N000,140,410,239. This paper was presented in part at the IEEE Conference on Intelligent Robots and Systems, Las Vegas, NV, 2003.

The authors are with Carnegie Mellon University, Pittsburgh, PA 15217 USA (e-mail: blisien@cmu.edu; deryck@cmu.edu; dsilver@andrew.cmu.edu; kantor@ri.cmu.edu; rekleitis@cmu.edu; choset@cs.cmu.edu).

Digital Object Identifier 10.1109/TRO.2004.837237

grid-based methods also grow in complexity with the size of the environment, in a manner which precludes them from scaling well to large environments and higher dimensions.

Our approach is to combine the strengths of a topological map with those of a feature-based map. We use a topological map to decompose the space into regions, within which we can build a feature-based map of moderate computational size to allow arbitrary localization. This frees us from having to perform a complex mapping algorithm in a large, global map. Our contribution is a new hierarchical map where the *generalized Voronoi graph* (GVG) [1], [2] serves as our high-level topological map, organizing a collection of feature-based maps at the lower level. We term this hybrid map the *hierarchical atlas*.

By choosing the GVG as the basis for our topology, we inherit all of its well-documented properties. In addition to segmenting the environment into manageable feature regions, the GVG offers the ability to plan paths and safely navigate in the presence of obstacles, a characteristic which most feature-based maps lack. Moreover, the GVG embodies an exploration strategy with which we can autonomously and completely chart an environment. The feature-based maps encode the necessary information with which we can localize arbitrarily while navigating, as well as provide detailed descriptions of areas which we can use to disambiguate regions.

II. Prior Work

SLAM [3] is the process of building a map of an unknown environment, while at the same time using that map to maintain an accurate estimate of the pose of the robot within the environment. The hierarchical atlas was originally developed to address the problems of autonomous exploration and SLAM in large-scale environments [4], and so we will discuss other maps in this context.

A. Feature-Based Maps

Conventional SLAM techniques have generally built feature-based maps. This process involves fusing observations of features or landmarks with dead-reckoning information to track the location of the robot in the environment and build a map of landmark locations. The numerous implementations typically include variations on the Kalman filter [5]–[9] or particle filters [10], [11]. The extended Kalman filter (EKF) [6], [9] uses a linear approximation of the system to maintain a state vector containing the locations of the robot and landmarks, as well as an approximation of correlated uncertainty in the form of a covariance matrix. One well-known weakness of the EKF is the growth of complexity due to the update step which requires computational time proportional to the square of the number of landmarks. This becomes prohibitive in large environments.

Each of these feature-based methods has its advantages and disadvantages. Yet, common to all is the increase in computational complexity with the size of the environment and number of landmarks. A number of techniques have been proposed to alleviate this problem, such as the extended information filter [12], [13], the unscented Kalman filter [7], and fastSLAM [11], but growth of complexity is inherent in maintaining a global map. Also, these methods do not provide a complete or structured means to direct exploration.

B. Submapping Strategies

Since the complexity of the global map cannot be avoided, some researchers have proposed dividing the global map into submaps, within which the complexity can be bounded. Connections between submaps are represented by an *ad hoc* topology consisting solely of their interconnection. Chong and Kleeman [14] introduced a topology of multiple, connected local maps where a new map is started when the variance in robot position becomes too large. These locally accurate maps are linked through coordinate transformations between their origins.

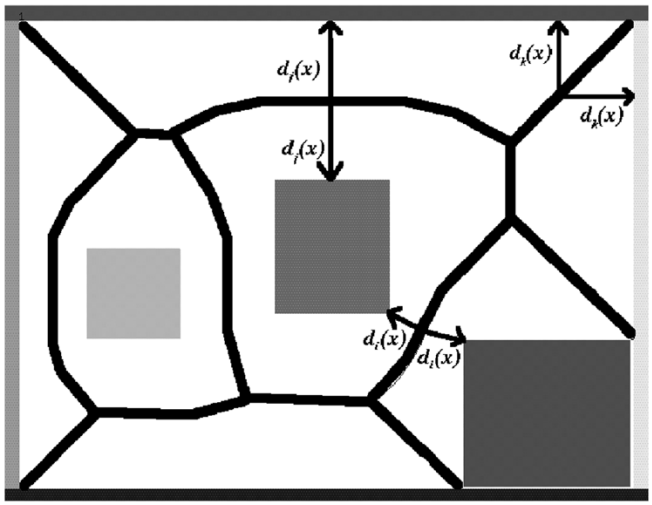

Fig. 1. In the plane, the GVG is the set of points equidistant to two obstacles.

Bosse *et al.* [15] introduced the term *atlas* to refer to a collection of submaps built with a similar method of creating a new map when the uncertainty of the robot location grows above some limit. Simhon and Dudek [16] proposed a strategy to create new maps in the presence of feature-rich regions or *islands of reliability*. On a related note, Thrun [17] uses a topological map to segment a grid-based map into submaps as a postprocessing step.

C. Topological Maps

Kuipers and Byun [18] developed a three-level hierarchy of control, topology, and geometry, with which they simulated an exploration and mapping strategy. The control level determined distinctive places, the topological level tied these distinctive places together, and the geometric level built metric maps around this framework. The authors particularly like that Kuipers, as well as Mataric [20], impart an underlying philosophy on how topological maps can be used for task decomposition. Mataric [20] was among the first researchers to successfully develop a mapping, path planning, and navigation strategy based on a topological map. In more recent work, Kuipers *et al.* [19] built detailed grid maps in the vicinity of nodes to aid in recognizing nodes while mapping.

Choset and Nagatani [21] use the GVG as the topology for their map. Nodes of the GVG are either *meet points*, the set of points equidistant to three or more obstacles, or *boundary points*, where the distance between two obstacles equals zero. These nodes are connected by edges which are paths of two-way equidistance (Fig. 1). The definition of the nodes and edges automatically induces well-defined control laws that allow a robot to trace an edge (either known or unknown *a priori*) and home in to a meet point. Exploration is achieved by having a robot sequentially traverse unexplored edges emanating from meet points. If the robot encounters a boundary node or a previously visited meet point (i.e., there is a cycle), the robot follows the partially explored GVG to a meet point with an unexplored edge associated with it. When there are no meet points with unexplored edges, exploration is complete. Therefore, in addition to prescribing low-level control laws, the GVG also

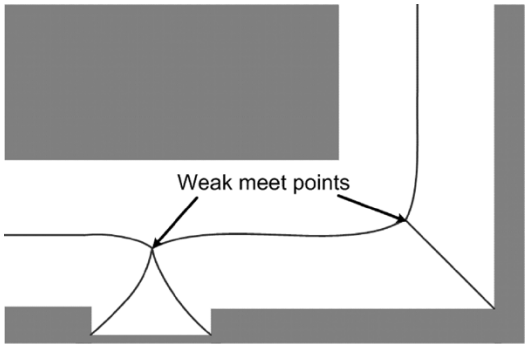

Fig. 2. Examples of weak meet points resulting from a door jamb and corner.

provides an arbitration scheme among the control laws to achieve exploration.

Localization with the GVG is trivial in environments for which the GVG is known to be a tree; the topology automatically dictates at which node the robot is located. The challenge arises when the GVG of the environment contains cycles and the nodes look similar to each other. Here, Choset and Nagatani [21] use the relationships among neighboring nodes to localize the robot; this is called *topological graph matching*. To further enhance the topological matching, Choset and Nagatani also use metric information about the nodes and edges. Dudek *et al.* [22] propose an exploration strategy which maintains a tree of all possible representations of the topological structure to ensure that the correct representation is always present in the exploration tree.

III. Hierarchical Atlas

The hierarchical atlas combines the strengths of a topological map with those of a feature-based map. The topological map decomposes the space into regions within which a feature-based map of moderate computational complexity is built. The complexity is limited, because the sizes of the subspaces and the number of features are also limited.

The topological graph of the hierarchical atlas is based on the GVG [1], [2], selected because its nodes have a definite location in the free space, and its edges not only connect neighboring nodes, but also define paths through the free space. Moreover, these paths can be traversed using sensor-based control laws [23]. In other words, the GVG is both abstract and embedded in the free space.

We do not use the full GVG. We eliminate *boundary edges*, which are those edges that terminate at an obstacle in a boundary point. Such edges can arise from door jambs and corners, as shown in Fig. 2, or simply from noise in the sensor data. Because they are potentially "unstable," add no topologically significant information, and are easily detectable, boundary nodes, boundary edges, and the *weak* meet points associated with them are left off the graph. The resulting structure is known as the reduced GVG (RGVG) [24].

Each submap of the hierarchical atlas is an *edge map*, a local map of one edge referenced from one meet point toward another. Thus, for each edge in the RGVG, there are two edge maps, one originating from each meet point. Since the nodes of the RGVG serve as origins for the edge maps, the local maps are tied both to the topological map and to the free space (Fig. 3). This is a key advantage that the RGVG contributes to the hierarchical atlas over the submapping strategies presented in Section II-B. Having stable origins that can be easily acquired allows for substantial simplification of edge-map alignment when comparing two edge maps (Section IV-E).

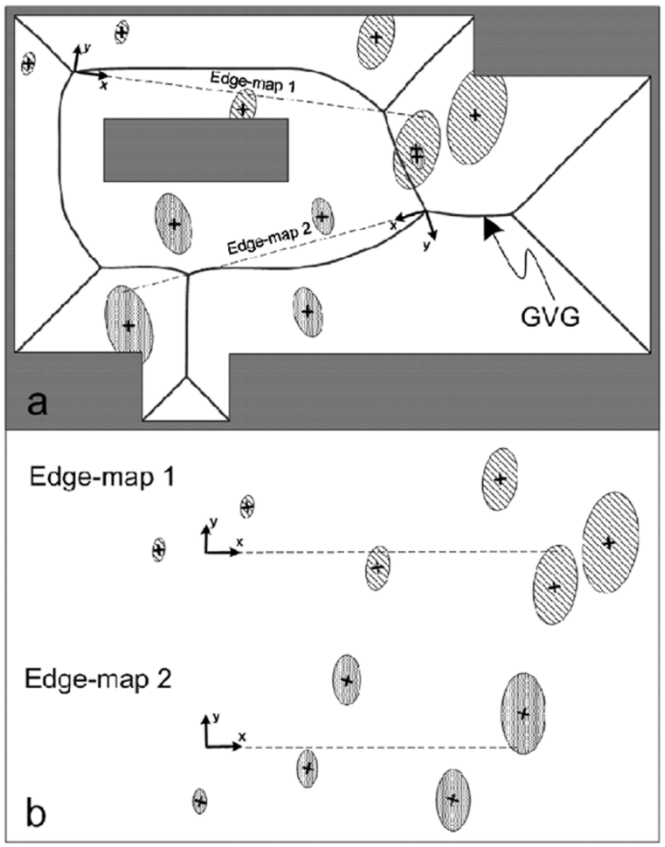

Fig. 3. (a) Placement and orientation of edge-map frames are determined by meet points in the GVG. (b) Resulting edge maps are stored as individual, abstract structures.

IV. Building the Hierarchical Atlas: HSLAM

SLAM involving exploration in large-scale spaces was the original motivation for developing hierarchical SLAM (HSLAM), which inspired us to create the hierarchical atlas. There are four criteria that HSLAM was designed to address: resolution versus scalability; reduction of computational complexity; obstacle avoidance; and exploration strategy. The tradeoff between resolution and scalability arises, since the environment must be represented with enough detail to be able to disambiguate regions and to use observations from the environment to control position uncertainty. On the other hand, the information must not be excessive, and must be organized in such a way so as to remain tractable in large environments. Computational complexity is closely related to the previous issue, and is important to moderate since the map must be built online if the robot is to use the map while exploring. In this paper, we do not calculate the actual improvement, but intuition suggests that complexity is reduced because instead of considering one large space, we consider several small ones. Obstacle avoidance is crucial, in order to be able to deploy an autonomous system which must harm neither itself nor the environment. Finally, for a robot to autonomously build a complete map, it must have a strategy to direct the full exploration of the environment.

HSLAM embodies solutions to each of these issues. The low-resolution, scalable topological map is used to organize the high-resolution local feature maps. Computational complexity is managed by breaking the feature map into submaps. The nodes also simplify the computation required for alignment during submap comparison. The GVG captures the salient aspects of the free space to enable obstacle avoidance and

path planning. Finally, the GVG offers the strategy of topological graph exploration.

A. Graph Exploration

All navigation and decision making are performed in the context of graph exploration, using the RGVG as explained in Section II-C. Graph exploration is achieved by tracing edges and building edge maps until no node in the graph has an unexplored edge. For tree-like environments, graph exploration is trivial, since there can never be an ambiguity in node location. However, when a cycle is traversed in the GVG, the robot must be able to determine that it has returned to a previously visited location. Since topological information alone is not sufficient to disambiguate regions, HSLAM compiles node information and creates the high-detail edge maps. We feel this is an important contribution of this work: the deliberate and complete exploration of unknown spaces.

B. Node Information

The characteristics of a node include its *degree* and *equidistance value*. The node degree is the number of nonboundary edges emanating from the meet point. Generally, this number is three, however, four-way meet points occur frequently in office environments.

The node equidistance value is the distance to the nearest obstacles at a meet point. This number is the result of sensor data, and thus has uncertainty associated with it. A scalar Mahalanobis distance test is used to determine possible matches, according to

$$\chi_{ij}^2 = \frac{(\rho_i - \rho_j)^2}{\sigma_i^2 + \sigma_j^2} \tag{1}$$

where ρ_i and ρ_j are the equidistance values for nodes i and j, and σ_i^2 and σ_j^2 are the variances in ρ_i and ρ_j. If χ_{ij}^2 falls below a threshold, then the equidistance for node i and node j match. The variance in equidistance after homing was determined by analyzing data collected at various meet points in the environment.

C. Edge-Map Creation

While traversing each edge, a feature map of landmark locations is built using an EKF [6]. The locations of the (r)obot and (l)andmarks are maintained in a state vector $\mathbf{X}$, with covariance matrix P, where

$$\mathbf{X} = [x_r \quad y_r \quad \theta_r \quad x_{l1} \quad y_{l1} \quad \cdots \quad x_{ln} \quad y_{ln}]^T. \tag{2}$$

When the robot is at a meet point and facing the edge departure direction, $\mathbf{X}$ and P are initialized to all zeros, defining the location of the robot at the origin. Every sensor update consists of an odometry measurement and a series of range and bearing measurements to visible point landmarks. The odometry is used as an input $\mathbf{U}$ to predict the state of the robot at the next step, $\hat{\mathbf{X}}_{k+1}$, according to the nonlinear state transition equation $f(\mathbf{X}, \mathbf{U})$. The corresponding prediction covariance $\hat{P}_{k+1}$ is calculated by

$$\hat{P}_{k+1} = \nabla f P_k \nabla f^T + Q \tag{3}$$

where Q represents the uncertainty associated with the odometry input.

When a feature is sensed, its measurement must either be associated with an existing feature in the map, or it must be added as a new measurement. The location of the robot, along with the locations of landmarks, are used to predict the expected values of range and bearing in a measurement estimate, $\hat{Z}$, using the sensor model equation, $h(\mathbf{X})$. The measurements of range and bearing, Z, are checked against the estimates with a Mahalanobis distance χ, computed by

$$\chi_{ij}^2 = (Z_i - \hat{Z}_j)^T S_{ij}^{-1} (Z_i - \hat{Z}_j) \tag{4}$$

where S_{ij} is the measurement covariance for the measurement/estimate pair. Each measurement is associated with the existing landmark yielding the minimum distance χ, provided that this minimum falls below an acceptance threshold. If multiple landmarks pass the acceptance threshold, then the measurement is discarded to avoid false matches. If the smallest χ exceeds a high threshold, meaning that the measurement is extremely unlikely to have been of any existing landmark, then the measurement is used to initialize a new landmark. With a set of range and bearing estimates, we can compute ∇h and the Kalman gain

$$K = \hat{P} \nabla h^T S^{-1}. \tag{5}$$

The difference between the estimated and measured values, or innovation, is used to compute a state update

$$\mathbf{X}_{k+1} = \hat{\mathbf{X}}_{k+1} + K(Z - \hat{Z}). \tag{6}$$

The updated covariance is then given by

$$P_{k+1} = \hat{P}_{k+1} - KSK^T. \tag{7}$$

Once the robot arrives at the terminating meet point, the state vector contains our feature map of landmarks, the final robot location is the location of the end meet point, and the covariance matrix represents the uncertainty in the map. We rotate the map and its covariance matrix through a standard linear transformation to align the x axis to pass through the terminal meet point. Thus, every aspect of the edge map's coordinate frame is completely tied to nodes of the GVG.

While traversing the edge, the robot records all odometry data and measurements to landmarks. When it arrives at the opposite meet point, the robot reverses this sensor log through the EKF to produce the edge map as referenced from the opposite meet point. So, each edge has two representations stored in the atlas.

We use a basic EKF to build our feature maps, showing that our approach can tame the computationally unruly algorithm in large environments. It should be noted that any method for creating feature maps could be used in its place.

D. Edge-Map Association

In the previous section, we discuss data association for landmarks within a single edge map. In this section, we consider data association for edge maps within the entire atlas. In other words, we need a means by which we can determine whether a recently constructed edge map is new or may already exist.

When the robot homes into a meet point, there is characterized uncertainty in the robot location relative to the true meet-point location. Since this uncertain robot location is used as the origin for a map, two maps of the same edge built at different times will have slightly different origins. The difference in origins can be viewed as an offset $\mathbf{Q}$, consisting of

$$\mathbf{Q} = [Q_x \quad Q_y \quad Q_\theta]^T \tag{8}$$

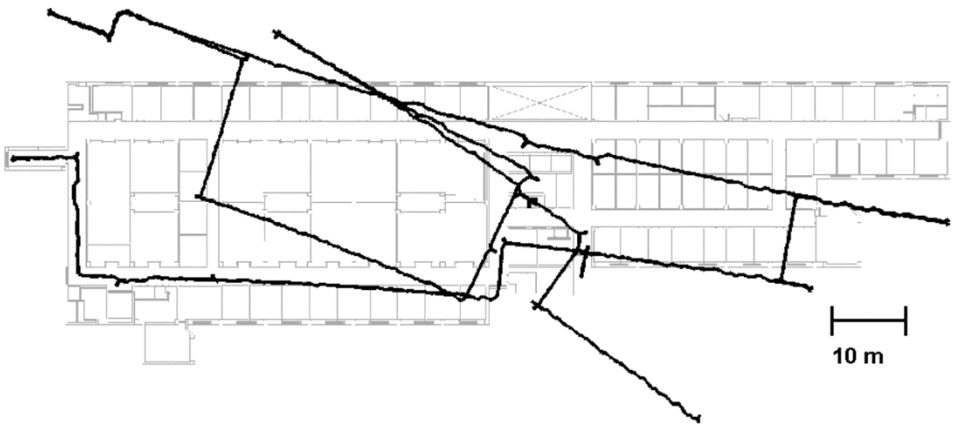

Fig. 4. Odometry logged during the autonomous HSLAM experiment is shown overlaid on a schematic of the environment. Small orientation errors accumulate to result in large position uncertainty over the length of the path.

where Q_x and Q_y represent the spatial offset and Q_θ is the offset in orientation. This offset must be determined to align the two maps before they can be matched.

The location of a landmark from the (C)andidate map can be expressed in the frame of the (E)xisting map by using the frame offset according to

$$X_C^E = \begin{bmatrix} x_C^E \\ y_C^E \end{bmatrix} = \begin{bmatrix} Q_x + x_C \cos Q_\theta - y_C \sin Q_\theta \\ Q_y + x_C \sin Q_\theta + y_C \cos Q_\theta \end{bmatrix}. \quad (9)$$

The offset $\mathbf{Q}$ is initially assumed to be zero with covariance matrix T. The matrix T is initialized with $x - y$ covariance, according to the uncertainty of the existing edge-map origin combined with the uncertainty of homing in on a meet point. The variance in Q_θ is an angular variance at the origin obtained by combining the spatial variance of the meet points at the distance of the opposite meet point.

The first step in the alignment is associating the landmarks between the two maps. This is accomplished with a Mahalanobis distance test for each landmark, with the innovation covariance matrix S consisting of

$$S_{ij} = P_{ii} + P_{jj} + M_j T M_j^T \quad (10)$$

where i and j correspond to landmarks in the existing map and candidate edge map, respectively. The matrix M_j is essentially a Jacobian which maps the effect of the offset into an $x-y$ uncertainty of the landmark j in the existing edge-map frame. This mapping is drawn from (9).

Once the landmarks in the two maps have been associated, an offset between the two maps is computed with a weighted least-squares algorithm. This is achieved by choosing a $\mathbf{Q}$ which minimizes

$$J_{\mathrm{LS}} = \left(\mathbf{X}_C^E - \mathbf{X}_E^E\right)^T \Sigma^{-1} \left(\mathbf{X}_C^E - \mathbf{X}_E^E\right) \quad (11)$$

where the landmark locations $\mathbf{X}$ for both edge maps are expressed in the frame of the existing edge map, and Σ is the combined uncertainty given by

$$\Sigma = P_E + P_C + MTM^T. \quad (12)$$

The resulting offset $\mathbf{Q}$ with updated covariance matrix T are then used in a map-wide Mahalanobis distance comparison

$$\chi^2 = \left(\mathbf{X}_C^E - \mathbf{X}_E^E\right)^T S^{-1} \left(\mathbf{X}_C^E - \mathbf{X}_E^E\right). \quad (13)$$

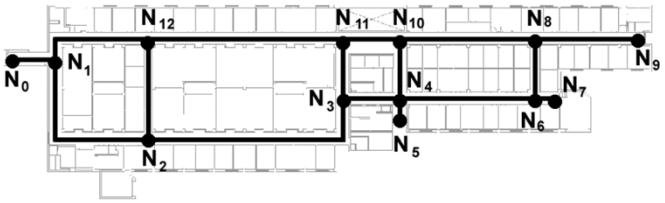

Fig. 5. RGVG for the sixth floor of Wean Hall is shown, where dark circles represent nodes with their indexes and lines show their interconnections.

The innovation covariance matrix is the combined uncertainty represented in the frame of the existing edge map. Therefore, the covariance matrix of the candidate frame must be rotated to reflect the offset, resulting in

$$S = P_E + GP_CG^T + MTM^T \quad (14)$$

where G is composed of two-by-two rotation matrices on the block diagonal, and M is recomputed according to the new offset.

There are actually two levels of information which must be used in order to confirm a successful match. Since the Mahalanobis distance increases for a fixed probability of match as degrees of freedom are increased, edge maps which have few associated landmarks could yield deceptively low Mahalanobis distances. Therefore, a minimum threshold of associated landmarks must be passed before a low Mahalanobis distance can signal a match.

While exploring a large unknown environment, a match between edges does not give an absolute confirmation that the edges are the same, since we must allow for isomorphic edge maps. Edge-matching information is better suited to eliminating nonmatching candidates in order to narrow the search pool of possible topological configurations. In practice, we found that the edge maps for the environments in which we ran experiments proved to be detailed enough to uniquely disambiguate edges. We exploited this observation to close loops in our implementation of autonomous HSLAM.

E. Edge Map Reobservation

When an edge is retraversed and the new edge map is positively associated with an existing edge, the information in the newly created edge map can be used to update the existing map. This is achieved by a merging operation [25] based on the Kalman filter.

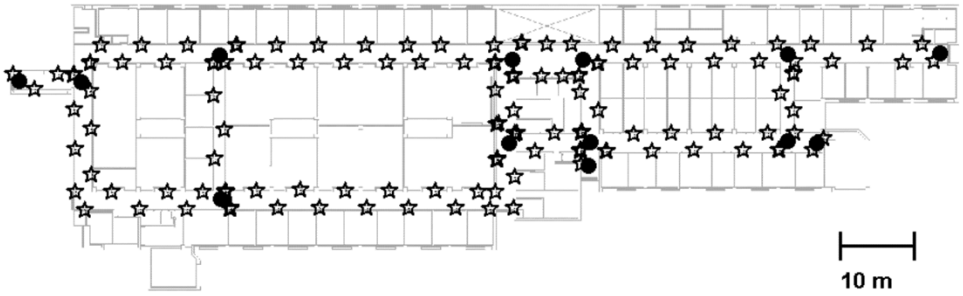

Fig. 6. Embedding of the edge maps of the hierarchical atlas in the environment. Nodes are shown as dark circles and the positions of landmarks as stars.

First the two edge maps must be aligned by following the procedure outlined in Section IV-D. Once the offset has been found, the existing edge map can be updated with a correction, $\Delta\mathbf{X_E}$, computed by

$$\Delta\mathbf{X}_E = P_E H^T S^{-1}\left(\mathbf{X}_C^E - \mathbf{X}_E^E\right) \tag{15}$$

where S is computed according to

$$S = HP_E H^T + GP_C G^T + MTM^T. \tag{16}$$

The purpose of the matrix H is to arrange the landmarks in the existing edge according to the order of association. The covariance of the existing edge is also updated by

$$\Delta P_E = -(P_E H^T S^{-1})S(P_E H^T S^{-1})^T. \tag{17}$$

The offset $\mathbf{Q}$ is a new measurement of the meet-point location relative to the existing location. The offset is used to move the origin of the edge map by shifting the locations of the other landmarks. The amount that the origin is moved depends on the uncertainty in the origin of the existing edge map, relative to the uncertainty of the new measurement. The location and uncertainty of the existing edge-map origin is maintained and updated with each observation by a merging operation similar to the one previously discussed. Since the opposite meet-point location also gets updated by (15), the edge must be rotated to once again align the opposite meet point with the x axis, as when the edge map was first created.

The final step is to insert any additional landmarks seen during the recent traversal of the edge. To assure that the landmark is new and not simply a poor estimate of an existing landmark, the minimum χ^2 for the landmark must pass a high threshold, similar to adding a landmark to the state vector, as discussed in Section IV-C.

F. Experimental Results

Our techniques were implemented on a Nomadic Scout mobile base using an omnidirectional camera setup to obtain range and bearing measurements to engineered landmarks. All navigation of the GVG was performed by processing data from the Scout's sonar sensors to build a local map of obstacles. Experimental data were collected while the robot autonomously mapped the sixth floor of Wean Hall at Carnegie Mellon University (Pittsburgh, PA).

To begin mapping, the robot accessed the GVG and then traversed the GVG to the meet-point location of node N_0 (see Fig. 5). The robot then traverses to node N_1, followed by node N_2, and so on around the outside of the loop by always choosing to explore edges to the right. As the robot compiles node information (Section IV-B) and edge-map information (Section IV-C), it is tested against the data which exists in the incomplete atlas. When the robot completes the outside loop, returns to node N_1, and traces the edge to N_0, the data from the two adjacent nodes and the edge map between them match the previously observed information stored in the incomplete atlas. The information compiled for the recent edge map is then used to update the matched edge, as described in Section IV-D. The closest node with unexplored edges is then node N_{12}, so the robot *back traces* the graph and continues to trace unexplored edges and close loops until no node in the graph has any unexplored edges.

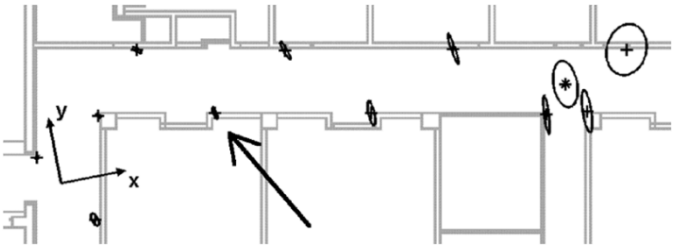

Fig. 7. Edge map connecting node $\boldsymbol{N_1}$ with $\boldsymbol{N_{12}}$ is shown with landmark locations as crosses and the opposite meet-point location as an asterisk, all with uncertainty ellipses. The edge-map origin is also shown to demonstrate how the edge map is referenced to the meet points.

Fig. 8. Landmark location indicated in Fig. 7 is shown in greater detail, demonstrating the inconsistency between the edge map and the floor plan. The corner of the doorway should lie at the location of the landmark.

The path of the robot as reckoned from odometry can be seen superimposed on a schematic of the sixth floor of Wean in Fig. 4. This is obviously insufficient for mapping, as small errors in orientation quickly accrue to cause sizable error in position. The RGVG of the environment is shown in Fig. 5, and the hierarchical atlas can be seen in Fig. 6. This figure was generated by tying the edge maps to meet-point locations determined on the floor plan. It can be seen that the hierarchical atlas closely resembles the true layout of the floor.

The individual edge map connecting node N_1 with node N_{12} is shown in Fig. 7. Landmark locations are shown as crosses, and the opposite meet point as an asterisk. Both are shown with their corresponding covariance ellipses. To facilitate repeatable experiments,

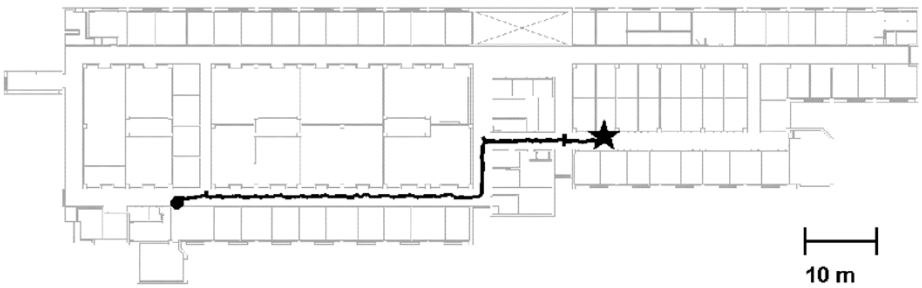

Fig. 9. Path planned from a conference room to an office on the sixth floor of Wean Hall is shown with the start as a dark circle and the goal as a star.

landmarks are generally placed along the walls of the corridors at corners. Most of the landmark locations align well with corners in the floor plan, except for one, indicated in Fig. 7 and shown in greater detail in Fig. 8. After investigating this problem, it was determined by manually measuring that the floor plan was actually incorrect, showing that our algorithm was accurate enough to identify errors in the schematic.

V. Other Applications of the Hierarchical Atlas

Although originally developed for the purposes of SLAM, the hierarchical atlas is well suited to handling other typical tasks of mobile robot operation. In fact, the reason that the hierarchical atlas is so well suited to the tasks of navigation, obstacle avoidance, and global localization, is that a robot needs the functionality for handling these tasks to achieve autonomous SLAM.

A. Navigation

In large environments, it would be impractical to pose navigation problems with start and goal locations in terms of global coordinates. Imagine receiving directions from Pittsburgh to New York City in terms of latitude and longitude. Rather than choosing a path heading northeast along whichever back road was heading in the proper direction, you would want to access a well-connected interstate to get into the city, after which you would leave the highway and ultimately find your destination.

The GVG provides this *roadmap* [23] through the environment and does so as the "safest" route, i.e., the furthest from all obstacles. While tracing between nodes of the GVG, there is no need to track the global position of the robot, since the path is determined entirely by the topology. If the robot is on the GVG and the goal location is closer than any obstacle, the robot is guaranteed a safe, straight-line path to the goal. Once on the *departure edge*, that edge which contains the goal, our edge map allows accurate localization to determine when to depart from the GVG, and then how to track the path of departure to the goal location, as referenced in the edge map.

Fig. 9 shows the path that the robot planned and traversed to get from a conference room to an office on the sixth floor of Wean Hall. The path is shown as recorded by the EKF along the corridors. Fig. 10 is a closer view of the robot path as it arrives at the goal, showing the point of departure from the GVG. When the robot reaches the node from which the goal is specified in $x-y$ coordinates, N_4, it loads the goal edge map into memory and initializes its location as the origin with uncertainty, according to the edge-map origin uncertainty and homing uncertainty. This quantity is rather large initially, but quickly decreases as measurements are made to landmarks and the robot localizes itself within the map. This can be seen in Fig. 10 as the consecutively shrinking covariance ellipses. It should be noted that though the meet-point origin has large uncertainty, the landmarks have much smaller uncertainty with respect to each other. The reduction of uncertainty as the robot drives to the goal shows the robot localizing within the local map of landmarks in which the goal location is described.

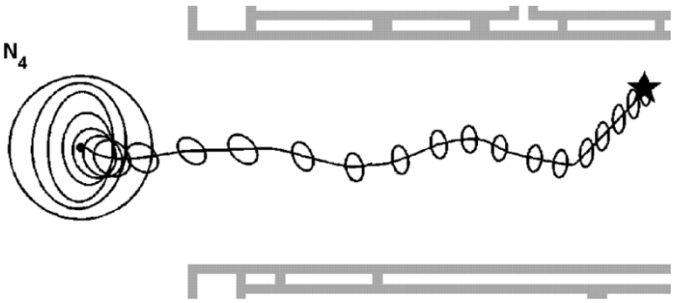

Fig. 10. Path of the robot on the departure edge is shown navigating from node N_4 to the goal location (star). Samples of the robot covariance are plotted along the path to show the uncertainty decreasing as the robot localizes in the edge map.

B. Global Localization: Wake-Up Problem

The hierarchical atlas is well suited to handle the global localization "wake-up" robot problem. The goal of the wake-up robot problem is to determine where in the environment a robot is initialized, given a map of the environment. This problem is nearly solved by the HSLAM implementation, since we require similar functionality to recognize cycles.

To resolve its location, the robot first accesses the GVG and then follows the GVG edge to a meet point. At the meet point, the node information of Section IV-B is used to eliminate nodes which are highly improbable matches and generate a list of candidate nodes. The robot then drives to a neighboring node, building an edge map while traversing. Upon arrival at the opposite meet point, the robot again compares node information to discard improbable matches. To further narrow the candidate pool, the recently constructed edge map is used to test against the remaining hypotheses, as described in Section IV-D. If an ambiguity still exists, further edges are traversed until it is settled.

To demonstrate global localization, the robot was arbitrarily placed on the sixth floor of Wean Hall and given the hierarchical atlas of Fig. 6. The robot accessed and then traced the GVG to N_1, at which point it compares the recently sensed node information to the nodes in the graph, yielding a candidate set of N_1, N_2, N_3, N_6, N_8, and N_{12}. The robot then chooses to trace the edge leading to N_2. Upon arrival, the robot could be at one of 18 locations, since the six original candidate nodes each has three possible destinations. Due to the arrangement of the original candidate nodes in the graph, some of the 18 possible nodes are repeats, and thus there are only 12 unique locations. The node information then dismisses six of the 12 nodes, coincidentally leaving the original six nodes: N_1, N_2, N_3, N_6, N_8, and N_{12}. The edge map for

Fig. 11. Path that the robot traversed during the "kidnapped" robot experiment is shown in contrast to the navigation experiment shown in Fig. 9. The start is denoted by a dark circle and the goal by a star.

the recently traversed edge then rules out all but one possible location, N_2, and the robot has successfully localized.

C. Global Localization: Kidnapped Problem

The "kidnapped" robot problem adds another layer of difficulty to global localization. In this situation, the robot first has a belief about its location in the environment, but is either moved by an outside agent or makes a poor decision and becomes lost. The difficulty lies in recognizing that the robot is lost. Once this is realized, the robot must "unlearn" its belief, and the problem becomes equivalent to the wake-up problem of Section V-B.

We tested the kidnapped problem with the same goal location as the navigation problem of Section V-A. The path of the robot can be seen in Fig. 11. The robot is intentionally "confused" by starting it on the opposite side of the GVG. When the robot accesses the GVG and turns right, it expects to drive to node N_2 of Fig. 5, but instead ends up at node N_1. Continuing on its topologically planned path to node N_4, it turns to the edge which leads to N_{12}. Expecting to arrive at N_3, the robot tests the recently traversed edge map against the expected edge map, which results in a failed match. Because of the failed test, the robot determines that it is lost and begins global localization with the information sensed at nodes N_1 and N_{12}, along with the edge map connecting them. With this information, the robot successfully localizes itself to node N_{12}, at which point, the robot plans a new path to node N_4, eventually arriving at the goal.

D. Nonstatic Obstacle Avoidance

Since the sensor-based GVG is used as the basis for all navigation in the hierarchical atlas framework, the robot has the ability to avoid obstacles that are not stored in the map. In fact, no obstacles are ever stored in the map, but are instead handled "on the fly" at the time of traversal.

It should be clarified that by "nonstatic" obstacles, we have in mind stationary objects that could be added to or removed from the environment, such as clutter in corridors. Our approach is not yet engineered to handle obstacles in motion, such as a large number of people moving in the vicinity of the robot.

This functionality was important when running experiments in an office environment over a period of a few months, since chairs, desks, boxes, and other clutter made their way into and out of the corridors regularly. To demonstrate this behavior, we will use the final portion of the navigation example of Section V-A where the robot departs the GVG. Fig. 12 shows the sonar map and robot path to the goal logged during the original navigation experiment. The sonar map roughly describes the status of the environment when originally mapped. Fig. 13 shows the sonar map and resulting robot path to the goal logged when some exaggerated clutter was added to the environment.

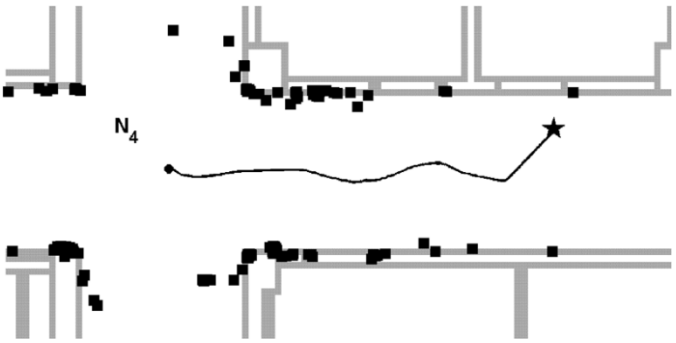

Fig. 12. Sonar map as the robot traveled along the final edge of Fig. 9 to the goal is shown. The goal meet point is shown by a dark circle and the goal location by a star. Filtered sonar points are represented by dark squares.

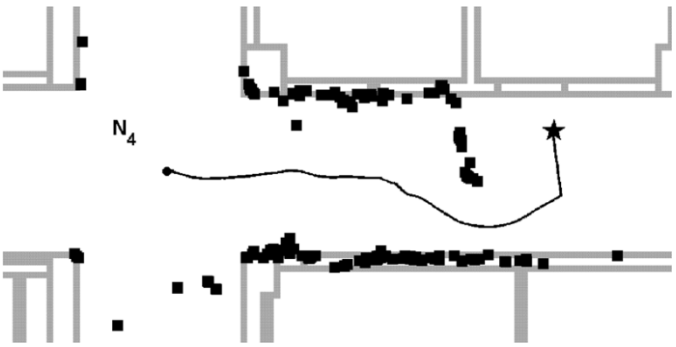

Fig. 13. As in Fig. 12, this figure shows the robot navigating to the goal on the final edge. However, the path is partially obstructed by an obstacle that was not present when the map was created. The goal meet point is shown by a dark circle and the goal location by a star. Filtered sonar points are represented by dark squares.

Despite the addition of clutter placed directly in the first path of the robot, it was able to navigate to the goal along the safest path, which remained furthest from all obstacles.

VI. Conclusions and Future Work

In this paper, we outlined the uses of the hierarchical atlas as a tool to enable navigation, exploration, mapping, and global localization for large-scale environments in the presence of uncertainty. We have also presented experimental results proving the feasibility and effectiveness of these techniques in a large, planar environment.

In order to detect when the robot is lost, a feature map of each edge is built as the robot navigates the GVG to a goal location. At each meet

point, the robot constructs an edge map from the recently traversed edge and then compares it with the edge map from which the robot believes was traveled using the techniques of Section IV-D. If the current edge map does not associate with the expected edge map, the robot declares itself lost and begins global localization initialized with the most recent two nodes and edge map. Once the robot has determined its location, it can replan a path and continue on to the goal.

The key feature of this hierarchical approach is the meaningful embedding of submaps into the free space. Since the RGVG has a concrete basis in the free space, we are able to tie our edge maps to the environment. Through the RGVG, we also acquire a *deliberate and complete* exploration strategy and the ability to autonomously navigate in the presence of obstacles, both of which are lacking from typical feature-based approaches.

We term this approach *hierarchical*, because we believe the topological architecture induces a natural hierarchy of symbols and connections among them in the free space. Similar to Kuipers, our symbols are the nodes of the RGVG, and the connections are the edges. However, our approach is hierarchical because the connections themselves can be further discretized into a collection of symbols and connections. Such lower-level symbols are the features of the edge maps At an even lower level, the robot's environment can be modeled by a local map, such as a fine grid. Finally, our approach decomposes the actual tasks in a hierarchical fashion. For example, with navigation, at the highest level, the topological map dictates the sequence of submaps through which to pass. This, in turn, determines which controls to invoke. At the next lower level, the feature maps allow for localization along the edge, and finally, at the lowest level, the odometry and sensor information provide feedback for control laws that follow the edge and drive toward the goal.

We do not believe the RGVG is the only choice for a topological map, nor are we committed to using Kalman filtering to create the lower-level edge maps. We believe the RGVG is a good example of a topological map, because there is a one-to-one relationship between the elements of the fundamental groups of the RGVG and the free space. This means that for every cycle in the free space, there is a corresponding cycle in the RGVG and *vice versa*. However, in its current form, the choice of the RGVG is not appropriate in wide-open environments which do not have particularly rich topologies. This suggests that we need a different high-level map other than the RGVG in such cases. Future work will consider developing such a map.

We chose the RGVG over the GVG because the RGVG handles instabilities inherent in the GVG. However, the RGVG does not handle all of them. Future work will also consider using the edge maps to better resolve such instabilities. The RGVG (as well as the GVG) generalizes into higher dimensions [1]. Whereas in the plane, RGVG edges are the set of points equidistant to two obstacles in three dimensions, RGVG edges are the set of points equidistant to three obstacles. We can use the same control laws to trace RGVG edges in three dimensions [2] and the same Kalman-filtering equations described in this paper to generate a hierarchical atlas in three dimensions.

Finally, instead of a Kalman filter, our approach could have easily used Bayesian methods for the low-level mapping. One advantage of Bayesian techniques over Kalman filtering is that they allow for multimodal hypotheses. It is worth noting that our topological framework allows for multiple hypotheses in global localization where the topological map is used to select which hypotheses to consider. It is our belief that the complexity of global localization is greatly alleviated by a coarse discretization of the world. However, when the discretization is based on an arbitrary method, such as a square grid, the accuracy of the discrete representation, and thus the localization, is directly related to the resolution of the grid. The RGVG offers a coarsely discretized map without any sacrifice in completeness of the representation of the environment or accuracy in localization.

ACKNOWLEDGMENT

The authors would like to sincerely thank A. Costa and L. Crosbie for their assistance in putting this paper together. They are also grateful for the insightful comments of the anonymous reviewers for this work.

REFERENCES

[1] H. Choset and J. Burdick, "Sensor based motion planning: The hierarchical generalized Voronoi graph," *Int. J. Robot. Res.*, vol. 19, no. 2, pp. 96–125, Feb. 2000.

[2] ——, "Sensor based motion planning: Incremental construction of the hierarchical generalized Voronoi graph," *Int. J. Robot. Res.*, vol. 19, no. 2, pp. 126–148, Feb. 2000.

[3] J. Leonard and H. Durrant-Whyte, "Simultaneous map building and localization for an autonomous mobile robot," in *Proc. Int. Workshop Intell. Robots Syst.*, May 1991, pp. 1442–1447.

[4] B. Lisien *et al.*, "Hierarchical simultaneous localization and mapping," in *Proc. Int. Conf. Intell. Robots Syst.*, vol. 1, Oct. 2003, pp. 448–453.

[5] R. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME J. Basic Eng.*, vol. 82, pp. 35–45, 1960.

[6] R. Smith *et al.*, "Estimating uncertain spatial relationships in robotics," in *Autonomous Robot Vehicles*, I. Cox and G. Wilfong, Eds. New York: Springer-Verlag, 1990.

[7] S. Julier and J. Uhlmann, "Simultaneous localization and map building using split covariance intersection," in *Proc. Int. Conf. Intell. Robots Syst.*, vol. 3, 2001, pp. 1257–1262.

[8] S. Roumeliotis and G. Bekey, "Bayesian estimation and Kalman filtering: A unified framework for mobile robot localization," in *Proc. Int. Conf. Robot. Autom.*, vol. 3, Apr. 2000, pp. 2985–2992.

[9] G. Dissanayake *et al.*, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Trans. Robot. Autom.*, vol. 17, no. 3, pp. 229–241, Jun. 2001.

[10] S. Thrun *et al.*, "A probabilistic approach to concurrent mapping and localization for mobile robots," *Machine Learning*, vol. 31, pp. 29–53, 1998.

[11] M. Montemerlo *et al.*, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proc. AAAI Nat. Conf. Artif. Intell.*, Edmonton, AB, Canada, 2002.

[12] E. Nettleton *et al.*, "Closed form solutions to the multiple platform simultaneous localization and map building (SLAM) problem," in *Sensor Fusion: Architectures, Algorithms, and Applications IV*, B. V. Dasarathy *et al.*, Ed. Bellingham, WA: SPIE, 2000, vol. 4051, pp. 428–437.

[13] S. Thrun *et al.*, "Simultaneous mapping and localization with sparse extended information filters," in *Proc. 5th Int. Workshop Algorithmic Found. Robot.*, J.-D. Boissonnat *et al.*, Eds., Nice, France, 2002.

[14] K. Chong and L. Kleeman, "Large scale sonarray mapping using multiple connected local maps," in *Proc. Int. Conf. Field Service Robot.*, Dec. 1997, pp. 538–545.

[15] M. Bosse *et al.*, "An atlas framework for scalable mapping," in *Proc. Int. Conf. Robot. Autom.*, vol. 2, Sep. 2003, pp. 1899–1906.

[16] S. Simhon and G. Dudek, "A global topological map formed by local metric maps," in *Proc. Int. Conf. Intell. Robots Syst.*, vol. 3, Victoria, BC, Canada, Oct. 1998, pp. 1708–1714.

[17] S. Thrun, "Learning metric-topological maps for indoor mobile robot navigation," *AI J.*, vol. 99, no. 1, pp. 21–71, 1998.

[18] B. Kuipers and Y. Byun, "A robot exploration and mapping strategy based on a semantic hierachy of spatial representations," *Robot. Auton. Syst.*, vol. 8, pp. 46–63, 1991.

[19] B. Kuipers *et al.*, "Local metrical and global topological maps in the hybrid spatial semantic hierarchy," in *Proc. Int. Conf. Robot. Autom.*, vol. 5, Apr. 2004, pp. 4845–4851.

[20] M. Mataric, "Integration of representation into goal-driven behavior-based robots," *IEEE Trans. Robot. Autom.*, vol. 8, no. 3, pp. 304–312, Jun. 1992.

[21] H. Choset and K. Nagatani, "Topological simultaneous localization and mapping (SLAM): Toward exact localization without explicit localization," *IEEE Trans. Robot. Autom.*, vol. 17, no. 2, pp. 125–137, Apr. 2001.

[22] G. Dudek, P. Freedman, and S. Hadjres, "Using multiple models for environmental mapping," *J. Robot. Syst.*, vol. 13, no. 8, pp. 539–559, Aug. 1996.

[23] J. Canny, "Constructing roadmaps of semi-algebraic sets I: Completeness," *Artif. Intell.*, vol. 37, pp. 203–222, 1988.

[24] K. Nagatani and H. Choset, "Toward robust sensor-based exploration by constructing reduced generalized Voronoi graph," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Seoul, Korea, Nov. 1999, pp. 1687–1698.

[25] R. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *Int. J. Robot. Res.*, vol. 5, no. 4, pp. 56–68, Winter 1986.