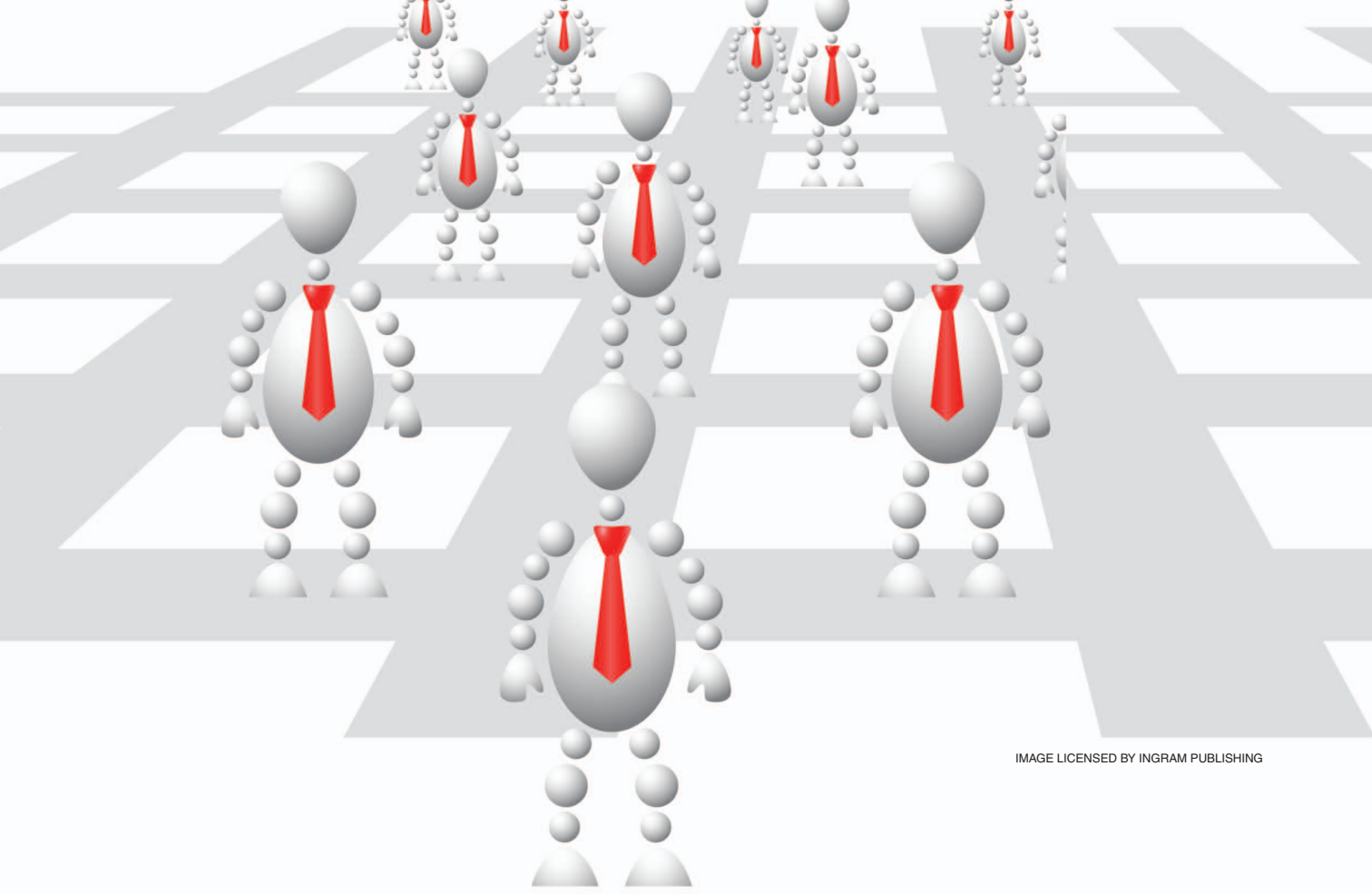

Group Mapping

A Topological Approach to Map Merging for Multiple Robots

By Sajad Saeedi, Liam Paull, Michael Trentini, Mae Seto, and Howard Li

Simultaneous localization and mapping (SLAM) is required for mobile robots to be able to explore a prior unknown space without a global positioning reference. Multiple robots can achieve exploration tasks more quickly but with added complexity. A useful representation of the map for SLAM purposes is as an occupancy grid map. In the most general case of multiple-robot SLAM, occupancy grid maps from multiple agents must be merged in real time without any prior knowledge of their relative transformation. In addition, the probabilistic information of the maps must be accounted for and fused accordingly. In this article, the generalized Voronoi diagram (GVD) is extended to encapsulate the probabilistic information encoded in the occupancy grid map. The new construct called the probabilistic GVD (PGVD) operates directly on occupancy grid maps and is used to determine the relative transformation between maps and fuse them. This approach has three major benefits over past methods: 1) it is effective at finding relative transformations quickly and reliably, 2) the uncertainty associated with transformations used to fuse the maps is accounted for, and 3) the parts of the maps that are more certain

Digital Object Identifier 10.1109/MRA.2014.2304091
Date of publication: 7 May 2014

1070-9932/14/$31.00©2014IEEE

are preferentially used in the merging process because of the probabilistic nature of the PGVD.

Introduction

The ability of an autonomous agent to sense its environment and situate itself within this environment is a cornerstone of mobile robotics. SLAM is the process of sensing an unknown environment and concurrently generating a consistent map of the world by fusing the available sensor data. As such, the pose of a robot can be estimated and an environment map can be built [2].

Extensive literature exists on SLAM for a single robot (for a review, see [3]). However, exploring unknown environments with multiple mobile agents has received comparatively less attention and can have significant advantages, including the following:

- Exploration and mapping can be done more rapidly.
- A distributed system is more robust to failures [4].
- The results are more accurate due to the redundancy of data.

With the stated benefits of multiple-robot SLAM comes significant challenges in implementation. The two main problems to overcome are map merging without a global position reference and map fusion incorporating map uncertainty. This article addresses both issues using only information within the maps in a robust and scalable way.

Abstract geometrical perception is a foundation for high-level reasoning and knowledge sharing. When multiple robots are supposed to explore and map an unknown environment cooperatively, there is a need to provide a logical infrastructure so that the robots can share their spatial perception and decide how to use the shared knowledge. If the salient information from the maps is extracted and shared among robots, the speed and accuracy of the mutual perception will improve, and communication channels will not be burdened with large amounts of unprocessed data.

A topological map is an abstract world representation in the form of connected paths and intersections. Humans and insects use topological maps to navigate, argue about paths and positions, and avoid obstacles [5]. As an example, pigeons have been shown to use highways and their intersections as a topological map to fly over long distances [6].

The general approach taken here is to achieve SLAM through graph matching where a graph is some reduced-form topological representation of the higher-level map structure [7]. The GVD described in the following sections is an appropriate graph structure.

Multiple-Robot SLAM

Past approaches to collaborative SLAM can be generally categorized based on whether they share raw sensor data [8] or processed maps [4]. Sharing raw sensor data results in more flexibility but requires high bandwidth and reliable communication between robots as well as more processing power. In contrast, sharing maps helps use less bandwidth and reduces the need to process raw data; however, the performance is dependent on the map quality. The latter method is referred to as *map merging* or *map fusion* and is the approach taken here.

Multiple-robot SLAM can also be categorized based on the method used to process measurement data. In feature-based SLAM [9], which is usually performed by cameras, unique objects called *features* or *landmarks* are extracted from measurements and used for localization. The spatial distribution of the features represents a model of the world. Feature-based multiple-robot SLAM has been implemented using an information filter [10], an extended Kalman filter [11], and a particle filter [12].

An alternative paradigm is view-based SLAM [13], which uses entire laser scans. Scans are matched using scan-matching algorithms. Our approach uses view-based SLAM. Thrun proposes a probabilistic multiple-robot view-based SLAM algorithm in [14]. This method is robust; however, the approximate initial poses of the robots are assumed to be known before the start of the mission.

A view-based multiple-robot SLAM algorithm is proposed by Howard using a particle filter [8], where it is assumed that robots will meet each other in the environment to determine their relative poses. This method is moderately fast but demands high computational power and memory since it is based on particle filtering.

An effective and fast approach to multiple-robot SLAM rests on the concept of map merging or map fusion [15]. In [4], a solution is presented based on occupancy map merging. This method uses map distance as a similarity index and tries to find similar patterns in two maps based on a random walk algorithm. The drawback of this method is that it can fail, especially when there are few similar patterns in both maps. This method is time-consuming and therefore problematic in large-scale maps that are common in indoor environments. A similar method is proposed in [16], with simulated annealing and hill climbing used to merge maps. This method becomes ineffective in maps with fewer overlaps.

The contribution of this research is a novel map fusion algorithm that exploits the properties of the GVD to achieve fast and accurate map fusion for large maps. In addition, the uncertainty in the maps is used to build a PGVD that encapsulates not only the topological structure of the map but also the confidences associated with different areas of the map. Once the PGVDs are built, the edges are matched using a two-dimensional (2-D) cross-correlation that will preferentially match the areas of the maps that have higher confidences. The resulting transformation can align maps to generate a global map. However, there is an uncertainty associated with this calculated transformation, which should also be propagated to the maps. The novel linearized uncertainty propagation (LUP) approach proposed in this research accounts for the uncertainty of the transformation. In LUP, if the transformation for each cell of the map is linearized, then the Gaussian uncertainty of the transformation is propagated to the transformed cell. This process is performed on all transformed cells; therefore, the resulting map carries the uncertainty of the transformation. Final map fusion is then

achieved with an entropy filter. The result is a fast, reliable, and robust method of fusing maps for multiple-robot SLAM.

It should be mentioned that the assumption in this article is that the individual maps developed by each robot are accurate and consistent.

To summarize, the proposed novel method for map fusion has the following key advantages.

- It accounts for uncertainties in the occupancy grid maps using the PGVD.
- It considers the uncertainty of the calculated transformation by linearization.
- It is fast and robust compared to other methods.
- It is able to preferentially match areas of the maps that are more certain.

Background

The GVD is a type of roadmap [17] that is the locus of points that are equidistant to at least the two closest obstacles. The GVD has the following two important properties that will be exploited.

1) The GVD is connected because the set of free space is connected and connectivity is maintained under a deformation retraction [17].
2) The GVD of a map is unique and is invariant to transformations because it is a retraction [17].

The GVD can be interpreted as a topological representation of the map structure that contains the key information intrinsic to the map but in a much more compact form.

There are different methods to generate GVD for a map. Mathematical morphological operations [18] are a fast and reliable method to build a GVD. In Blum's method, introduced in [19], the GVD is developed using a maximal inscribed circle method that is inefficient for large maps. A dynamic version of the GVD, which improves performance near nonconvex obstacles, is proposed by Lau et al. [20]. Although this method has good results, it is time-consuming.

Beeson et al. [21] propose an extended Voronoi graph algorithm to improve the efficiency of building the GVD when the robot is limited with its sensory horizon. This method performs well in noisy environments by eliminating spurious junctions.

Previous Topological Approaches to SLAM

Voronoi graphs and graph matching in its different forms have many applications in SLAM. There are many topological solutions for single-robot SLAM, such as the works by Choset et al. [22], [23], the annotated generalized Voronoi graph (AGVG) [24], the work by Beeson et al. [25], Bayesian inference [26], and the semantic approach with place labeling by Friedman et al. [27].

Choset and Nagatani [22] propose a topological SLAM algorithm for a single robot based on the GVD. In [24], an AGVG is used for single-robot SLAM. The proposed method is based on a matching scheme for solving the data association problem. This method identifies the corresponding parts of the map in two tree-formed Voronoi graphs. One form represents a local observation, and the other form represents the internal map of the robot.

In [28], a solution to detect and recognize topological features is proposed using Delaunay triangulations. In [27], a semantic approach is used by a robot to generate a topological map that can identify different places. Identified places can be used for different autonomy applications.

A multiple-robot SLAM based on topological map merging using both structural and geometrical characteristics of the Voronoi graph is proposed in [29]. The assumption in this article is that a robot will be able to recognize areas of the map that correspond to vertices. In this case, the topological map is built on the occupied space as opposed to the free space. The method in [29] is claimed to be fast. However, a limitation is that the maps are not updated.

In this article, a skeletonization approach is extended to be probabilistic and used for map matching of multiple robots. A common problem encountered when using skeletonization for SLAM is that the skeleton has no closed-form solution, and heuristic methods for generating it tend to be slow. In our approach, morphological operations are used to generate the GVD, as explained in the "Building the Probabilistic GVD" section. The proposed approach is fast enough to be used in real time and guarantees connectedness of the skeleton [18]. In addition, the probabilistic nature of the proposed PGVD allows areas of the maps that are known with higher certainty to be preferentially matched.

Transforming Occupancy Grid Maps with an Uncertain Transformation

In general, any type of uncertainty in this context is identified as having five main causes: environment, sensors, robots, models, and computations. Thrun et al. state that "uncertainty arises if the robot lacks critical information for carrying out its task" [3].

Map fusion involves dealing with two types of uncertainty.

1) *Uncertainty due to pose and sensor measurements*: This kind of uncertainty is represented within the occupancy grid map. Specifically, it originates from the uncertain pose of each robot and is embedded in the map of the robot. This can be considered as a form of uncertainty at the individual robot level. Throughout this article, the terms *uncertainty* and *map uncertainty* refer to this type unless otherwise stated.
2) *Uncertainty due to transformation*: If we can estimate the uncertainty in the transformation that relates two maps, then we can account for this uncertainty in the fusion process. In this article, the terms *transformation uncertainty*, *rotation uncertainty*, and *translation uncertainty* are used to refer to this type of uncertainty.

The uncertainty associated with the relative transformation matrix is represented as a covariance matrix. The covariance error is propagated through the probabilistic transformation function using the LUP formulation, a novel map-merging method introduced in [1] and expanded upon here. This section addresses the propagation of the uncertainty in transformation given the transformation matrix and

its covariance as a map is being transformed so that it may be merged with another.

Assume that $q = [q_i \;\; q_j]^T$ is a multivariate random variable denoting the position of the cell (i, j) from map m. The probability density function (PDF) of q is shown by $p(q)$ and it is assumed to be a delta function. The mean value of q is shown by $\mu_q = [\mu_{q_i} \;\; \mu_{q_j}]^T$. The point q is transformed to another point, r, with the PDF of $p(r)$. The transformation is performed by $f(\cdot)$ defined as

$$r = f(\psi, x, y, q) = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix} q + \begin{bmatrix} x \\ y \end{bmatrix}, \tag{1}$$

where ψ is the rotation angle and x, y are translation elements.

In general, the nonlinearity and uncertainty of the transformation will cause a non-Gaussian distribution for r. However, it is possible to approximate the non-Gaussian distribution with a Gaussian distribution. By linearizing the transformation using the first-order Taylor expansion, a Gaussian distribution is assigned to the transformed point

$$p(r) = |2\pi\Sigma_r|^{-\frac{1}{2}} \exp\left\{\frac{-(r-\mu_r)^T \Sigma_{r-1}(r-\mu_r)}{2}\right\} \sim \mathcal{N}(r; \mu_r, \Sigma_r), \tag{2}$$

where the mean of the normal distribution, μ_r, is calculated by (1). Assume that the covariance of the transformation matrix has the form

$$\Sigma_{tr} = \begin{bmatrix} \sigma_{xx}^2 & \sigma_{xy}^2 & \sigma_{x\psi}^2 \\ \sigma_{xy}^2 & \sigma_{yy}^2 & \sigma_{y\psi}^2 \\ \sigma_{x\psi}^2 & \sigma_{y\psi}^2 & \sigma_{\psi\psi}^2 \end{bmatrix}. \tag{3}$$

Through linearization, the covariance of the transformed point, Σ_r, is calculated by

$$\Sigma_r = F_{xy\psi} \Sigma_{tr} F_{xy\psi}^T + F_q \Sigma_q F_q^T, \tag{4}$$

where $F_{xy\psi}$ and F_q are the Jacobians of $f(\cdot)$ defined as

$$F_{xy\psi} = \frac{\partial f(\psi, x, y, q)}{\partial(\psi, x, y)}, F_q = \frac{\partial f(\psi, x, y, q)}{\partial(q)}, \tag{5}$$

and Σ_q is the covariance of the point q which is zero. Note that the covariance Σ_r is a function of q, the location of the cell. Figure 1 depicts the problem with a hypothetical nonlinear function representing the transformation. Figure 1(a) shows the point q, which should be transformed by the given transformation function. The position of this point is deterministic, therefore its distribution is a delta function. However, the transformation function is not certain, shown by setting boundaries around the nominal transformation function [Figure 1(b)]. A Gaussian distribution is assigned to the transformed point [Figure 1(c)].

Equation (4) gives the shape of the Gaussian distribution for every transformed point. Figure 2 shows this process. The point q from map m is transformed to point μ_r. By applying the Gaussian defined in (4), depicted by $\mathcal{N}(r; \mu_r, \Sigma_r)$, the point μ_r takes a Gaussian form.

The next step is to extend this formulation to all points of the map. Algorithm 1 explains this process. For simplicity, we

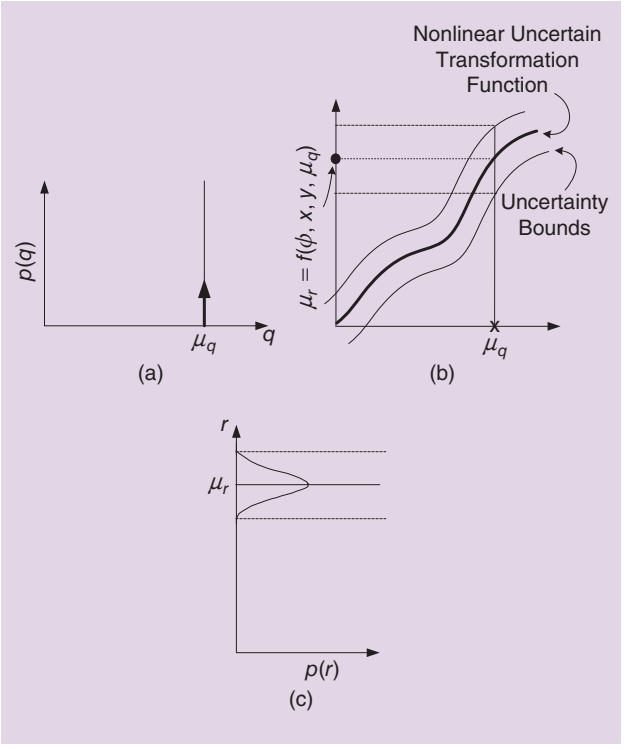

Figure 1. (a) A point q with the distribution of $p(q)$ is transformed by an uncertain transformation function. (b) The uncertainty of the transformation function is shown by setting boundaries around the nominal function. (c) After linearization, the transformed point, r, will have a Gaussian distribution, $p(r)$.

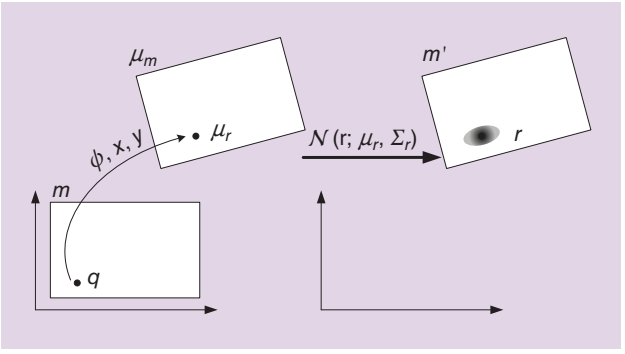

Figure 2. The linearized transformation uncertainty propagation. Point q is transformed to point μ_r according to the rotation ψ and the translation x, y. Then a Gaussian kernel, $\mathcal{N}(r; \mu_r, \Sigma_r)$, is convolved with the transformed point. Σ_r is calculated through linearization.

use $T_{x,y,\psi}(q)$ to denote that the point q is rotated according to ψ and then translated according to (x, y). The same concept applies for transformation of a map, m, shown by $T_{x,y,\psi}(m)$.

First, the map m is transformed according to the given transformation, shown by $T_{x,y,\psi}(\cdot)$ (line 1). This is shown in Figure 2, where the resulting map is marked by μ_m. In line 2, the final map, which includes the uncertainty of the transformation, is initialized by μ_m. In line 4, for every point of the transformed map, the covariance is calculated using (4). In line 5, the Gaussian kernel based on the covariance is calculated. Note that the Gaussian kernel is center originated, which

Algorithm 1. Linearized uncertainty propagation.

Require: Occupancy grid map: m,
Transformation: ψ, x, y,
Uncertainty of the transformation: Σ_{tr}.
Ensure: Transformed occupancy grid map: m'.
1: $\mu_m = T_{x,y,\psi}(m)$
2: $m' = \mu_m$
3: **for all** $\mu_r = [\mu_{r_i}, \mu_{r_j}]^T \in \mu_m$ **do**
4: Calculate Σ_r using (4)
5: $h \leftarrow \mathcal{N}(\mu_r, \Sigma_r)$
6: $m'(\mu_{r_i}, \mu_{r_j}) \leftarrow \mu_m(\mu_{r_i}, \mu_{r_j}) \circledast h(\mu_{r_i}, \mu_{r_j})$
7: **end for**

means that the center point of the kernel is $h(0,0)$. The mean of the kernel is the transformed point, while its 2×2 covariance matrix is calculated by (4). It is important to note that for each point of the transformed map, the kernel takes different values. Then, in line 6, every transformed point is convolved with the Gaussian kernel. The operator $\circledast$ denotes the convolution of a Gaussian kernel with the map [18] and is defined as

$$m'(k,l) = \mu_m(k,l) \circledast h(k,l) = \sum_{j=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} \mu_m(i,j)\, h(k-i, l-j). \quad (6)$$

For simplicity of implementation, it is assumed that the size of the kernel is the same as μ_m.

Now m' can be fused with its pair map using the entropy filter method detailed in the "Map Fusion with the Entropy Filter" section.

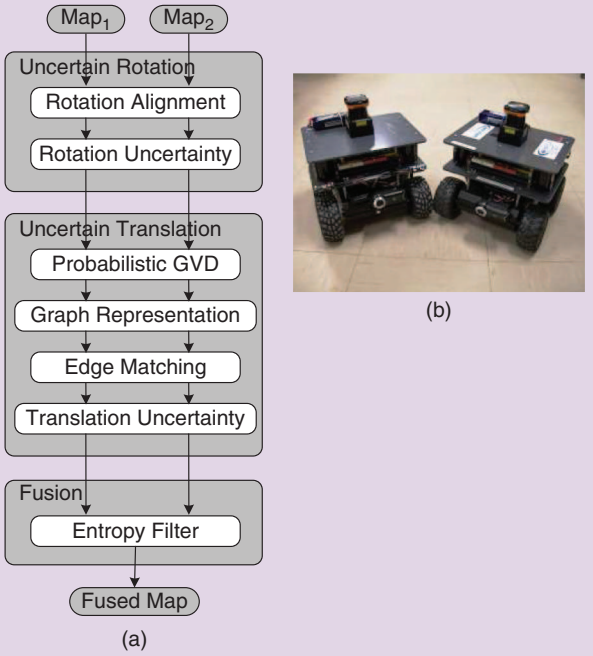

Figure 3. (a) The proposed map fusion algorithm. Two input maps, map_1 and map_2, are fused by finding their relative transformation matrix. No prior information is available regarding the relative position of the two respective robots. (b) The experimental robots, CoroBots, each equipped with a laser ranger and wheel encoders.

Probabilistic Map Merging with the GVD

In this section, we present the details of the map fusion process with the probabilistic GVD introduced in [1]. Consider the case of two mobile robots, R_1 and R_2, equipped with laser rangers exploring an environment and building occupancy grid maps (OGMs) [3]. In an OGM representation, each cell in map $m_k(i,j)$, $k = \{1,2\}$, is a binary random variable (RV) where $p(m_k(i,j) = 1) = p(m_k(i,j))$ is the probability that the cell at location (i,j) is occupied in the map of robot k. It is convenient to represent the OGM using the log odds representation of occupancy [3]

$$l_k(i,j) = \log \frac{p(m_k(i,j))}{1 - p(m_k(i,j))}. \quad (7)$$

Without loss of generality, assume that R_2 transmits its local map, map_2 to R_1 through a wireless channel. R_1 is now responsible for incorporating the transmitted map into its own local map, map_1. There are three main challenges that need to be overcome.

1) The relative transformation from map_1 to map_2 needs to be found.
2) The uncertainty of the transformation should be accounted for.
3) The OGM probabilities from map_2 need to be incorporated with the OGM probabilities of map_1.

An overview of the elements of the algorithm is shown in Figure 3(a). The subsequent sections will describe each of the block components in detail.

A simulated example accompanies each step of the algorithm to aid with explanation. Figure 4(a) shows the simulated environment, where three poles are located inside a rectangular room. The two robots map the room starting near the big pole but moving in opposite directions. Figure 4(b) and (c) shows the two local maps after some time has passed. Without loss of generality, it is assumed that the second map [Figure 4(c)] is fused into the first map [Figure 4(b)]. Free, occupied, and unknown cells are shown by different shades of gray, using the OGM standard. The darker the grid cell, the higher the probability of occupancy.

Uncertain Rotation Alignment

In structured environments such as urban or indoor settings, the relative rotation between maps can be found easily using the Radon transform, as shown by [30]. The Radon transform is the projection of the image intensity along a radial line oriented at a specific angle. The peak points in the Radon transform will correspond to the straight line segments in the image. As a result, it is possible to resolve the relative rotation, ψ, between two images by looking for peaks in the Radon images of both maps. However, due to environment similarity, four rotation hypotheses are considered and only one is accepted by a similarity index [30]. At the output of this block, there are the two aligned maps, m_1 and m_2, with the same size of $M \times N$, given by

$$m_1 = \text{map}_1, \qquad m_2 = T_{0,0,\psi}(\text{map}_2). \quad (8)$$

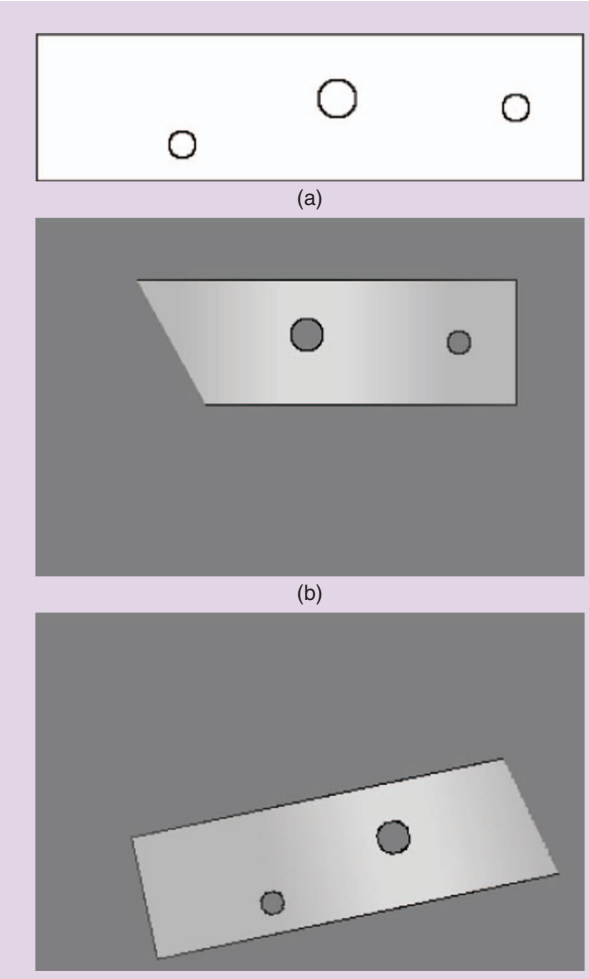

Figure 4. (a) A simulated environment to show the process of map merging. (b) The first OGM covers the right part of the environment. (c) The second OGM covers the left part of the environment.

The uncertainty in the relative rotation must be accounted for in the subsequent calculation of the relative translation. This is performed using Algorithm 1. This means that after finding the rotation, its uncertainty on the map is propagated using Algorithm 1, assuming zero translation and $\sigma_{xx}^2 = \sigma_{yy}^2 = \sigma_{xy}^2 = 0$. Now, the process of finding the relative translation can be done knowing that uncertainty of the rotation is already incorporated into the rotated map. Figure 5(b) shows the aligned map after applying the uncertain rotation alignment. More details about the rotation alignment process can be found in [30].

Building the Probabilistic GVD

Finding the relative translation between maps is much more challenging than finding the relative rotation. The search for overlapping parts of maps can be slow. As a result, we use a novel probabilistic topological representation of the map as described in this section. The proposed solution is based on the idea that the search for overlaps in the topological space, which represents salient information, is easier and consequently faster, than in the metric space. The PGVD is found for each of the two maps, m_1 and m_2. This process is completed in two steps.

1) Find the GVD efficiently using mathematical morphological operations on the binary OGM.
2) Compute the associated probabilities of each cell in the GVD based on the actual probabilities in the OGM.

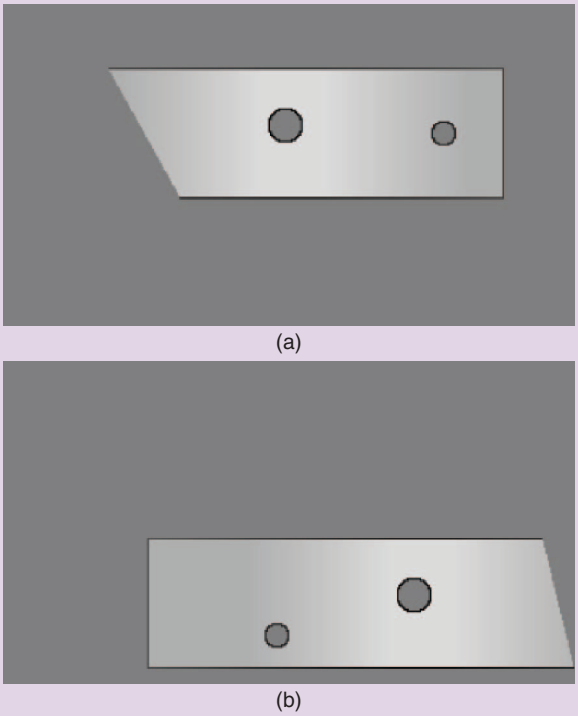

Figure 5. (a) The first OGM. (b) The second OGM after being rotated. The relative rotation is calculated by comparing the Radon images of the maps.

Finding the GVD Using Mathematical Morphology

Mathematical morphology defines the application of set operations on binary images using convolution between the image and defined masks. It has been used extensively in computer vision and image processing. A set of basic operators is defined in mathematical morphology such as erosion, dilation, opening, closing, skeleton, and hit-or-miss transform with different properties. The most important property is that they are translation invariant.

The GVD of the binary map is generated using eight D-type hit-and-miss transform masks [18]. Each mask is designed for a particular situation to guarantee the connectedness (using the connectivity-eight model). The GVD is represented as a matrix, $S = [s_{i,j}]$, with the same size as the map, $M \times N$, defined as

$$[s_{i,j}]_{\,i=1..M,j=1..N} = \begin{cases} 1 & \text{if } m(i,j) \in \text{GVD} \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

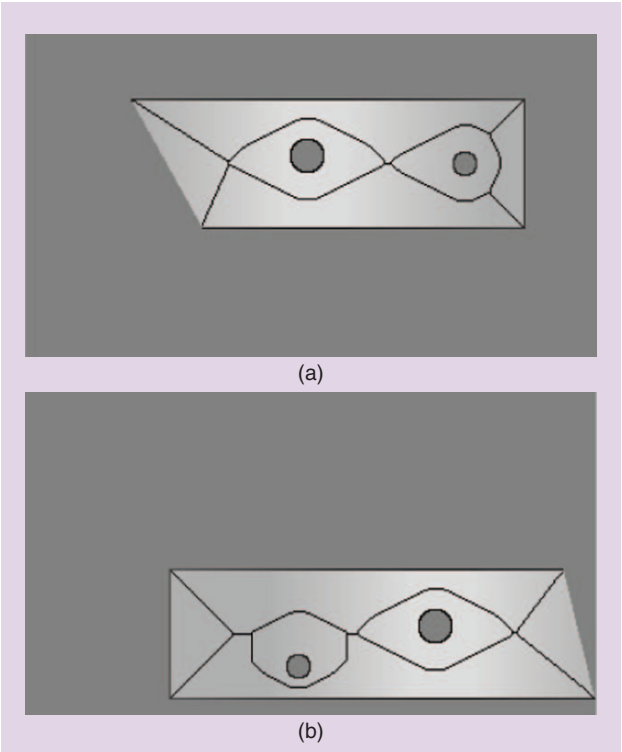

Figure 6. The GVD of (a) the first and (b) the second map.

Figure 6(a) and (b) shows the GVDs of the two aligned maps. The probabilistic GVDs will be based on these GVDs.

Finding the Probabilistic GVD

The skeleton developed by the proposed probabilistic GVD approach does reflect the uncertain nature of the OGM. While the GVD structure is generated from a binary version of the OGM, the probabilistic information in the OGM should be reflected in the GVD structure. A new structure, the PGVD, is built that combines the structure of the GVD with the probabilistic nature of the OGM.

Definition 1: Contact Points. The contact points of a cell in the GVD are all the occupied cells in m that have the same distance from the cell as the closest occupied cell. The probabilities of the contact points in the OGM are used to build the PGVD.

Definition 2: Probabilistic GVD. Each cell in the GVD is represented by a binary RV representing the probability that it has two or more occupied contact points based on their probabilities of occupancy in the OGMs.

Consider that a GVD, S, contains s cells, $G_i, i = 1..s$. Each cell in the GVD has an associated set of n_i contact points in the binary map m.

The PGVD, S^p, has the same structure as the GVD, except that each of the s cells is represented as a binary RV, $G_i^p, i = 1..s$ where $p(G_i^p = 1) = p(G_i^p)$ is the probability that cell G_i^p has been correctly placed in the GVD. Each cell in the PGVD has an associated set of contact points $C_i = \{c_1, c_2, ..., c_{n_i}\}$, $i = 1..s$, where these contact points are at the same locations as the contact points of G_i except they are in the OGM so they have associated probabilities of occupancy. Each contact point is a binary RV where $p(c_j = 1) = p(c_j) \forall j = 1..n_i$ is the probability that the contact point is occupied taken directly from the OGM.

A cell belongs in the GVD if at least two of the contact points are occupied. For each cell in the PGVD G_i^p, we must determine the probability that it has at least two occupied contact points based on their probabilities of occupancy.

To do so, another RV $N_{G_i^p}$ is defined that represents the probability distribution of the number of occupied contact points of G_i^p. $p(N_{G_i^p} = k)$ is the probability that cell G_i^p contains k occupied contact points. The $p(N_{G_i^p} = k)$ is defined as a function of n_i, the number of contact points in the OGM, and k, the number of contact points that are actually occupied

$$P(N_{G_i^p} = k) = f(n_i, k). \tag{10}$$

The function $f(n_i, k)$ can be defined recursively as

$$f(n_i, k) = p(c_{n_i}) f(n_i - 1, k - 1) + (1 - p(c_{n_i})) f(n_i - 1, k). \tag{11}$$

Intuitively, this equation says that the probability of having k out of n_i contact points occupied is equal to the probability that contact point c_{n_i} is occupied and that k–1 of the remaining $n_i - 1$ contact points are occupied, plus the probability that c_{n_i} is not occupied and k of the remaining $n_i - 1$ contact points are occupied.

The base cases for the recursion are cases where all of the cells must be occupied or none of the cells must be occupied

$$f(n_i, n_i) = \prod_{j=1}^{n_i} p(c_j), \qquad f(n_i, 0) = \prod_{j=1}^{n_i} (1 - p(c_j)). \tag{12}$$

By induction it can be proven that $f(n_i, k)$ is a valid probability density function that produces all possible combinations of contact points and sums to 1 [1]. Now, $p(G_i^p)$ is given by

$$\begin{aligned} p(G_i^p) &= p(N_{G_i^p} \geq 2) = \textstyle\sum_{k=2}^{n_1} p(CN_i = k) \\ &= 1 - f(n_i, 0) - f(n_i, 1). \end{aligned} \tag{13}$$

The PGVD is now defined as

$$S^p = \bigcup\nolimits_{i=1..s} G_i^p. \tag{14}$$

The output from this block will be a PGVD S_k^p for each input map m_k with $k = \{1, 2\}$.

Figure 7(a) and (b) shows the PGVD of the maps. In contrast with Figure 6, where the GVD cells are deterministic, these cells are now probabilistic. The probabilities of the cells of the PGVD are represented by their grayscale intensity, with darker cells having a higher probability of being true GVD cells.

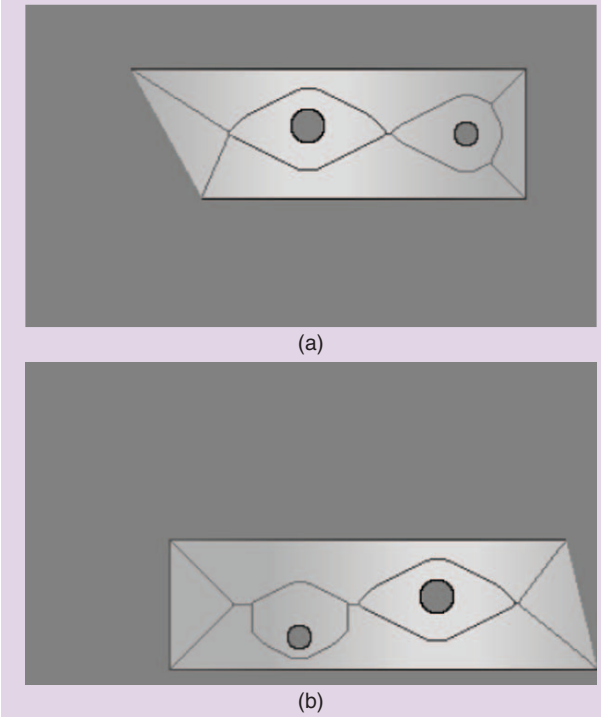

Figure 7. The PGVD of (a) the first and (b) the second map.

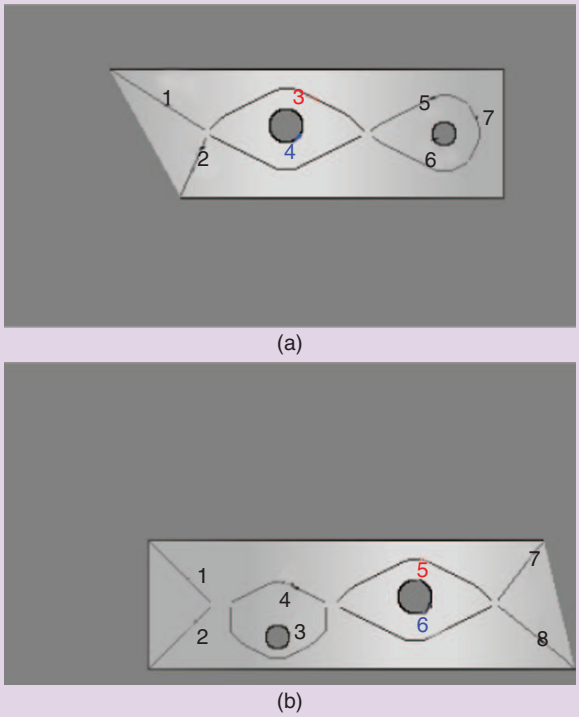

Figure 8. The PEM of (a) the first and (b) the second map.

Graph Representation

The PGVD is then processed to be represented as a graph with edges and vertices. A vertex can be identified as any cell in the PGVD with more than two adjacent occupied cells. The set of all vertices is defined as V.

To identify edges in the graph, a dilation mask is applied to each vertex in V. The result of the dilation operation is a new map, D, that contains the dilated vertices. The edge matrix, E, is found as $S^p - D$. This operation is performed on each of the two GVDs to produce two edge maps: $E_k, k = \{1, 2\}$.

The probabilistic edge matrices (PEMs), E_1^p and E_2^p, have the same structure as the edge matrices, E_1 and E_2, but the cells are the probabilistic ones extracted from the PGVD. These PEMs are now used to find the translational transformation between the two maps m_1 and m_2. The edges with short lengths are removed to avoid processing short edges that have a higher chance of producing false matches.

Figure 8(a) and (b) shows the PEMs for each map where the short edges have been removed. The edges of each map are marked with numbers. The first map has seven edges, and the second map has eight edges. Figure 9(a) and (b) shows two enlarged edges of the first map, number 3 and number 7. The grayscale intensity of each cell in these edges represents the probability of that cell in the PGVD (the probability that it is a valid GVD cell). Edge number 3 is located in the area of the original map, which has high certainty, so the probability of the cells of this edge being in the GVD is high. However, edge number 7 is located at the end corner of the map, which is less certain. Therefore, the probability of the cells in this edge is lower.

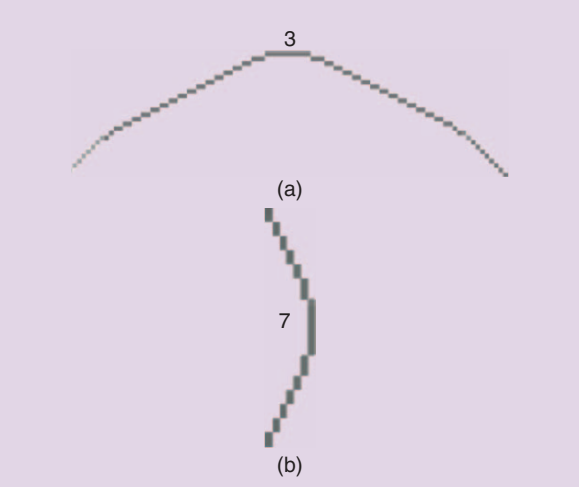

Figure 9. The two edges from Figure 8(a). (a) Edge number 3. (b) Edge number 7.

Edge Matching for Translation Alignment

The edges of each probabilistic edge matrix are matched using a 2-D cross-correlation. To speed up the cross-correlation, each edge from each edge matrix is represented as a submatrix of E where its size is such that it is as small as possible while still containing the entire edge. For a given edge matrix,

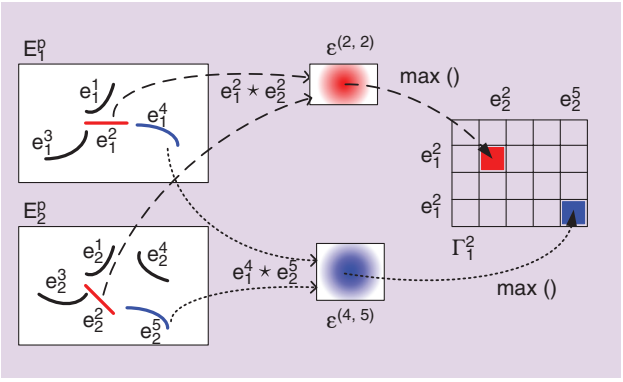

Figure 10. The formation of the Γ matrix from the PEMs. All edges from the PEMs are cross-matched. The peak of each match is stored at the corresponding location in Γ.

E_k^p with N_k edges, each edge, $e_k^{i_k}, i_k = 1..N_k$ is given by ($m_i \times n_i$ is the size of the submatrix containing e_k^{ik})

$$e_k^{i_k} = E_k^p(pr_i{:}pr_i + m_i, pc_i{:}pc_i + n_i). \tag{15}$$

Here, pr_i and pc_i point to the start of the submatrix. To do matching, each edge of E_1^p is cross-correlated with each edge of E_2^p.

The 2-D cross-correlation of two matrices $e_1^{i_1}$ and $e_2^{i_2}$ with the respective sizes $(m_1 \times n_1)$ and $(m_2 \times n_2)$ is given as

$$\mathcal{E}^{(i_1,i_2)} = e_1^{i_1} \star e_2^{i_2}, \tag{16}$$

where $\mathcal{E}^{(i_1,i_2)}$, the cross-correlation matrix of edge $e_1^{i_1}$ and edge $e_2^{i_2}$, has size $(m_1 + m_2 - 1) \times (n_1 + n_2 - 1)$. The operator $\star$ denotes the 2-D cross-correlation operation.

The maximum value in the $\mathcal{E}^{(i_1,i_2)}$ matrix quantifies the best match between $e_1^{i_1}$ and $e_2^{i_2}$ based on all possible combinations of translations between the two edge matrices. This value is computed using (16) for every combination of edges: $i_1 = 0..N_1$ and $i_2 = 0..N_2$.

We can then define the similarity matrix, Γ_1^2, between E_1^p and E_2^p as

$$\Gamma_1^2 = [\gamma_{i_1 i_2}]_{i_1=1..N_1; i_2=1..N_2}, \tag{17}$$

$$\gamma_{i_1 i_2} = \max(\mathcal{E}^{(i_1,i_2)}). \tag{18}$$

An overview of the formation of the similarity matrix, Γ, is shown in Figure 10. First, all probabilistic edges from the PEMs are cross-matched based on (16). For instance, the two edges shown in blue are matched to generate the correlation matrix in blue, $\mathcal{E}^{(4,5)}$. The maximum value of $\mathcal{E}^{(4,5)}$, is stored at the corresponding location in Γ based on (17) and (18).

The best candidate for a match corresponds to the maximum value in the similarity matrix Γ_1^2

$$[i_1^*, i_2^*] = \underset{i_1,i_2}{\operatorname{argmax}}(\Gamma_1^2[i_1, i_2]), \tag{19}$$

where i_1^* and i_2^* are the indices of the most likely similar edges from the two maps.

The relative translation is calculated by resolving the translation vector that matched these two edges using the following relation [30]:

$$T = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \mu_x(e_1^{i_1^*}) - \mu_x(e_2^{i_2^*}) \\ \mu_y(e_1^{i_1^*}) - \mu_y(e_2^{i_2^*}) \end{bmatrix}, \tag{20}$$

where the $\mu_x(\cdot)$ and $\mu_y(\cdot)$ functions return the mean values of the elements of the input matrix evaluated along the x and y axes, respectively.

The final transformed map is defined as

$$m'_2 = T_{x,y,0}(T_{0,0,\psi}(\text{map}_2)) = T_{x,y,\psi}(\text{map}_2). \tag{21}$$

For the PEMs in Figure 8, the Γ matrix defined in (17) is

$$\Gamma_1^2 = \begin{bmatrix} 0.77 & 0.26 & 3.26 & 2.35 & 4.71 & 3.23 & 0.35 & 2.31 \\ 0.28 & 0.55 & 1.61 & 0.60 & 0.87 & 0.87 & 1.37 & 0.32 \\ 3.51 & 2.44 & 7.38 & 13.72 & \mathbf{52.13} & 19.63 & 2.30 & 5.13 \\ 0.71 & 0.71 & 10.94 & 10.15 & 19.63 & \mathbf{52.08} & 0.91 & 1.68 \\ 0.49 & 0.93 & 5.33 & 11.86 & 17.59 & 15.85 & 1.17 & 1.13 \\ 0.77 & 0.49 & 9.90 & 7.32 & 13.32 & 18.86 & 0.62 & 1.79 \\ 0.48 & 0.48 & 2.74 & 0.55 & 0.75 & 0.75 & 1.18 & 0.56 \end{bmatrix}. \tag{22}$$

There are seven rows corresponding to the seven edges in the first map and eight columns corresponding to the eight edges in the second map.

From (19), the peak of the matrix is at $i^* = 3$ and $j^* = 5$. This means that edge 3 from the first map is the best match with edge 5 from the second map (Figure 8, where matching pairs of edges have identification numbers with the same color). Moreover, edge 4 from the first map is a good match with edge 6 from the second map. However, both matching results will generate similar translation vectors. Using (20), the translation is $T = [-80, -70]^T$, which is the translation required to fuse the maps. In general, if there are multiple good match candidates, the translation resulting from each of the matches can be used or the results can be averaged. In the case in which there are no matching edges in the maps, the map-fusion process cannot be completed. The existence of the solution and verification of the matches are explained further in the "Existence of a Solution" section.

It should be emphasized that the associated probabilities for each cell in the edges of the PGVD play an important role in determining the values from the cross-correlation. The edges with cells with a higher probability of being true GVD cells will result in higher cross-correlation values.

Existence of a Solution

False matches can be identified by analyzing the results of the 2-D cross-correlation. Assume that there is an edge with l_1 cells within $e_1^{i_1^*}$ and an edge with l_2 cells within $e_2^{i_2^*}$. Assume that the best match in the 2-D cross-correlation results in L matching cells. Define the subset of cells in $e_k^{i_k}$ that were matched as $\epsilon_k^{i_k}$. We can define the average confidence for the matched edge as

$$\alpha_k^{i_k} = \frac{1}{L} \sum_{l=1..L} p(\epsilon_k^{i_k}). \tag{23}$$

By expanding (16), $\max(\mathcal{E})$ becomes $\alpha_1^{i_1} \alpha_1^{i_2} L$.

Given two edges with lengths l_1 and l_2 with unknown mutual matching cells, the best matching or maximum value in the correlation matrix should satisfy the following to be considered a candidate match:

$$\max(\mathcal{E}^{(i_1,i_2)}) > \rho \alpha_1^{i_1} \alpha_1^{i_2} \min(l_1, l_2), \tag{24}$$

where ρ is a desired matching percentage. If there is no element in the Γ_1^2 matrix that satisfies (24), then the matching process fails. For each pair of the edges, i_1 and i_2, the following relation should hold to accept them as a valid match:

$$\frac{\max(\mathcal{E}^{(i_1,i_2)})}{\rho \alpha_1^{i_1} \alpha_1^{i_2} \min(l_1, l_2)} > 1. \tag{25}$$

For our example, $\alpha_1^3 = 0.7402$ and $\alpha_2^5 = 0.7381$. Assuming that the desired ρ is 95%, then

$$\begin{aligned} \rho \alpha_1^3 \alpha_1^5 \min(l_3, l_5) &= (0.95)(0.7402)(0.7381) \min(96, 96) \\ &= 49.83 < \max(\mathcal{E}^{(3,5)}) = 52.13, \end{aligned} \tag{26}$$

which means that the match between edge 3 from the first map and edge 5 from the second map is a valid match. This matching criterion can be calculated for each pair of the edges represented in the Γ matrix. For the given example, if the criterion in (25) is calculated for all edges in the Γ matrix, then the following matrix is the result:

$$\begin{bmatrix} 0.089 & 0.030 & 0.217 & 0.160 & 0.225 & 0.152 & 0.035 & 0.191 \\ 0.030 & 0.060 & 0.156 & 0.059 & 0.061 & 0.060 & 0.122 & 0.031 \\ 0.277 & 0.192 & 0.223 & 0.637 & 1.046 & 0.397 & 0.158 & 0.289 \\ 0.056 & 0.056 & 0.327 & 0.467 & 0.399 & 1.045 & 0.062 & 0.093 \\ 0.053 & 0.101 & 0.329 & 0.758 & 0.778 & 0.692 & 0.119 & 0.088 \\ 0.085 & 0.053 & 0.627 & 0.473 & 0.605 & 0.846 & 0.060 & 0.141 \\ 0.062 & 0.062 & 0.322 & 0.065 & 0.063 & 0.062 & 0.128 & 0.064 \end{bmatrix}, \tag{27}$$

in which only the elements greater than one are valid matches.

The aligned maps after applying the uncertain rotation are used to find the relative translation. After finding the translation, Algorithm 1 is used again to propagate the uncertainty of the translation, assuming zero rotation and $\sigma_{\psi\psi}^2 = 0$. We now have a fully probabilistic representation of the required transformation and can proceed to fuse the maps.

Map Fusion with the Entropy Filter

After finding the relative transformation between the two maps, the probabilities are combined and filtered to produce the final map. The data received in m'_2 is akin to a batch of sensor data and should be incorporated by using the additive property of the log odds representation of occupancy originally defined in (7)

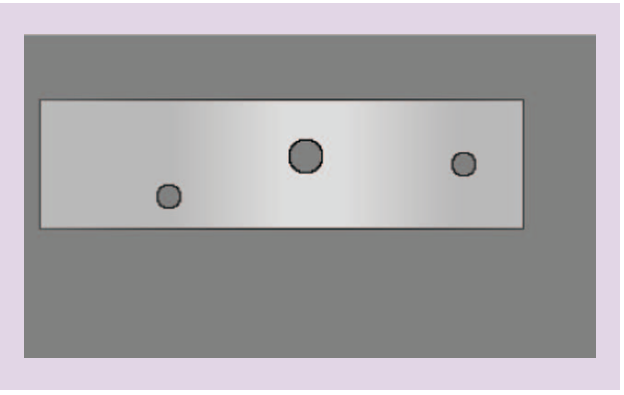

Figure 11. The final fused maps at the output of the entropy filter.

$$l_{\text{fused}}(i,j) = l_1(i,j) + l'_2(i,j), \tag{28}$$

for all $i = 1..N$, $j = i..M$. The probabilities can then be recovered from (7). The entropy filter is applied to the fused map, m_{fused} [31]. The entropy filter compares the original map, m_1, and the fused map, m_{fused}, and rejects updates that result in higher entropy.

For the case of a discrete binary RV, such as each cell of the OGMs, $m(i,j)$ with $p(m(i,j)) = p_{ij}$, the entropy can be described by

$$H(m(i,j)) = -p_{ij} \log p_{ij} - (1 - p_{ij}) \log(1 - p_{ij}). \tag{29}$$

Mutual information, I_{ij}, is defined as the reduction in entropy at location (i,j) between the original map, m_1, and the fused map, m_{fused}

$$\mathrm{I}_{ij} = \mathrm{H}(m_1(i,j)) - \mathrm{H}(m_{\text{fused}}(i,j)). \tag{30}$$

The final map, m_{final}, is defined as

$$m_{\text{final}}(i,j) = \begin{cases} m_{\text{fused}}(i,j) & I_{ij} \geq 0 \\ m_1(i,j) & I_{ij} < 0 \end{cases}, \tag{31}$$

where only values from m_{fused} that result in positive information are kept. It should be noted that, in the case of subsequent map fusion operations, the original maps that contain no information from others should be used. This is to avoid having overconfidence by fusing maps repeatedly. Figure 11 shows the final fused map after the proposed entropy filter.

Experimental Results

To demonstrate the effectiveness of the proposed method, four experiments are presented. The first one is performed on a standard online data set. In the presentation of the results for this experiment, most of the details of the proposed algorithm are explained. The second experiment is a

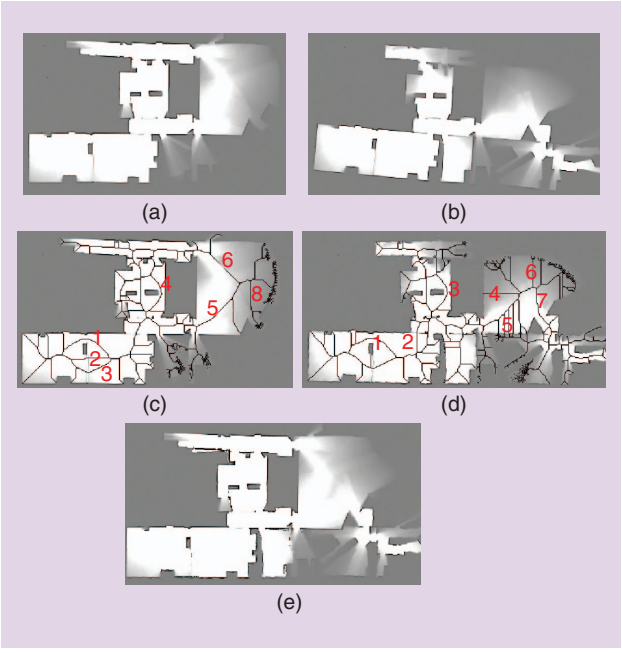

Figure 12. (a) The map$_1$, (b) map$_2$, (c) PGVD of map$_1$, (d) PGVD of rotated map$_2$, and (e) fused map.

real-world experiment performed with two CoroBots in an indoor environment in the basement of the University of New Brunswick (UNB). The third experiment shows the map merging algorithm with three maps of the Department of Electrical and Computer Engineering of UNB. Finally, the last experiment is performed in a simulated MATLAB environment with more obstacles to show the performance of the algorithm for more complex environments.

The three robots used are built by CoroWare, Inc., and each is equipped with high-speed Phidget Encoders and a Hokuyo UTM-30LX laser ranger, as shown in Figure 3(b). SLAM on the individual robots is performed with the particle-filtering algorithm presented in [8].

Experiment 1: RADISH Data Set

The first experiment is performed on the RADISH Fort AP Hill data set [32]. Figure 12(a) and (b) shows two input maps before alignment. Figure 12(c) and (d) shows the PGVDs of the maps after being aligned. The alignment is 5.5°. The marked edges are the filtered edges for cross-matching. The Γ_1^2 matrix is shown in (34). From (19), $i^* = 4$ and $j^* = 3$, which means the fourth edge of the first map has the highest similarity to the third edge of the second map. These two edges are used to calculate the translation vector from (20). The final translation vector is $T = [-25, -1]^T$, and the rotation is $\psi = 5.5^o$.

From (23), $\alpha_1^4 = 0.86$ and $\alpha_2^3 = 0.89$ are the average confidences of the matched edges. The lengths of these two edges were 44 and 45 cells. The matching percentage, ρ, was chosen to be 95%. According to (24),

$$(0.95)(0.86)(0.89)\min(44,45) = 31.99 < 32.9, \quad (32)$$

therefore the match is valid. Note that the match of e_1^1 with e_2^1, given in $\Gamma_1^2(1,1)$, is disqualified because the average confidences of those edges were higher: $\alpha_1^1 = 0.91$, $\alpha_2^1 = 0.92$, and the lengths of the edges are 56 and 62

$$(0.95)(0.91)(0.92)\min(56,62) = 44.54 > 31.6. \quad (33)$$

$$\Gamma_1^2 = \begin{bmatrix} \mathbf{31.6} & 2.0 & 6.4 & 6.1 & 1.9 & 0.8 & 5.4 \\ 12.9 & 1.9 & 2.0 & 11.7 & 0.9 & 0.5 & 1.6 \\ 11.9 & 2.8 & 5.4 & 3.0 & 1.4 & 0.5 & 2.5 \\ 10.7 & 6.7 & \mathbf{32.9} & 1.0 & 5.3 & 1.8 & 8.1 \\ 9.1 & 1.9 & 8.9 & 2.3 & 0.9 & 0.5 & 1.6 \\ 5.9 & 0.4 & 2.1 & 0.4 & 0.4 & 0.2 & 1.9 \\ 3.2 & 9.9 & 3.9 & 0.5 & 11.6 & 4.0 & 6.9 \\ 0.9 & 7.2 & 2.9 & 0.3 & 10.2 & 3.5 & 4.1 \end{bmatrix}. \quad (34)$$

To show the benefit of the probabilistic approach, Γ_{det}, the deterministic Γ, is shown in (35). In this case, matching is performed directly on the GVD instead of the PGVD. The edges that were on the periphery of the explored area in the GVD are now matched quite well when, in reality, those edges do not represent well the structure of the discovered map. As a result, the correct match is much less clear. By comparing those elements from Γ_{det}, shown in bold, with the same elements in Γ_1^2, it is obvious how the probabilistic structure can eliminate false or weak matches. Figure 12(h) shows the final aligned maps.

$$\Gamma_{\text{det}} = \begin{bmatrix} \mathbf{38} & 2 & 9 & 10 & 2 & 2 & 9 \\ 17 & 2 & 2 & \mathbf{30} & 2 & 1 & 2 \\ 16 & 3 & 6 & 5 & 3 & 2 & 3 \\ 11 & 9 & \mathbf{40} & 1 & 10 & 8 & 12 \\ 14 & 2 & 10 & 3 & 2 & 1 & 2 \\ 14 & 1 & 7 & 1 & 1 & 1 & 8 \\ 6 & \mathbf{29} & 9 & 1 & \mathbf{29} & \mathbf{29} & 25 \\ 3 & \mathbf{29} & 9 & 1 & \mathbf{32} & \mathbf{32} & 21 \end{bmatrix}. \quad (35)$$

Experiment 2: Two CoroBots

In this experiment, two robots are used. Figure 13(a) and (b) shows the OGMs built by the robots. The total area is 81.02 m^2, and the trajectories were 22.5 and 25.9 m. In both maps, there are nonoverlapping corridors with almost the same size (15.5 and 12.2 m). However, the algorithm is capable of rejecting them as matching and finding the transformation based on the overlap. Figure 13(c) and (d) shows the PGVD of the aligned maps with selected edges for matching marked with numbers. The edges marked with number 2 in Figure 13(c) and number 3 in Figure 13(d) are used to calculate the translation. Finally, Figure 13(e) shows the final fused map.

Experiment 3: Three CoroBots

This experiment is performed in a larger environment, with an area of approximately 600 m^2, and three agents are involved. The trajectories of the robots are approximately 60, 35, and 55 m. By merging the maps, loop closure happens successfully. Figure 14(a)–(c) shows the three local maps. Maps of Figure 14(b) and (c) are fused to Figure 14(a). The

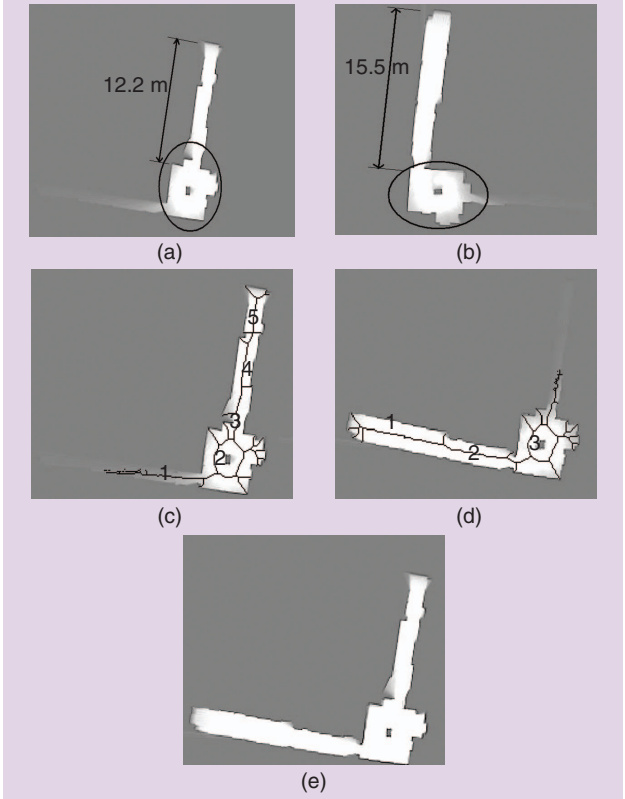

Figure 13. (a) The map_1, (b) map_2, (c) PGVD and PEM of map_1, (d) PGVD and PEM of rotated map_2, and (e) fused maps.

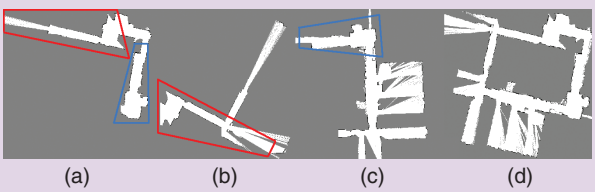

Figure 14. Three partial maps are fused together to generate a global map. (a) is the base map to which (b) and (c) are fused. The overlaps are marked with polygons. (d) The final fused map that depicts loop closure.

overlapping region is indicated in the figure. The fused global map of the environment is shown in Figure 14(d).

Experiment 4: Simulated Highly Cluttered Environment

This experiment is performed in a simulated environment, which includes 30 blocks with different sizes and orientations. Two local maps are generated that have a 50% overlap. Map merging using the proposed probabilistic approach successfully fuses the local maps shown in Figure 15(a) and (b). Figure 15(c) and (d) demonstrates the PEMs of the aligned maps, which are probabilistic GVDs with the short edges removed. The first PEM has 44 edges, and the second one has 45 edges. A few of the matching pairs of the edges are highlighted with

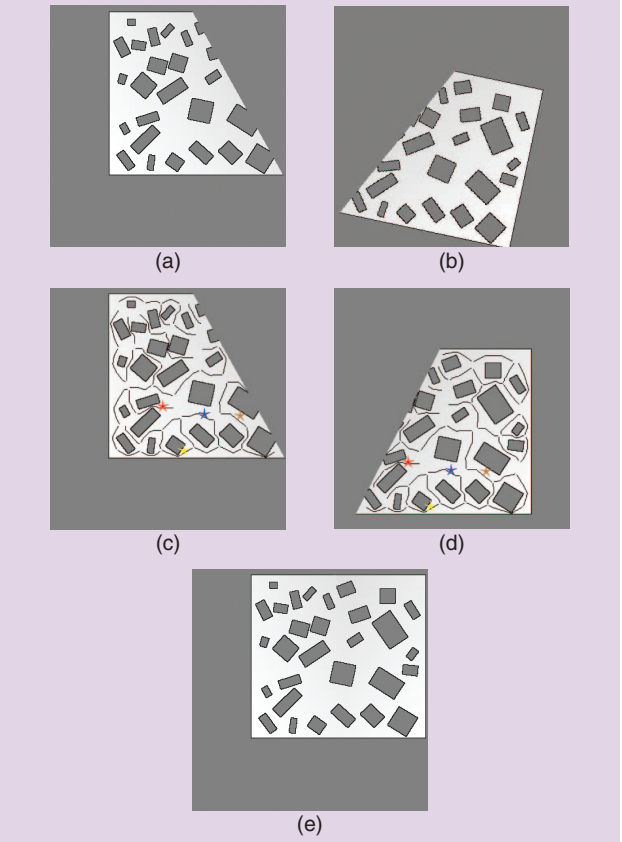

Figure 15. The experiment on a simulated, highly cluttered environment. (a) The first OGM. (b) The second OGM. (c) The PEM of the first map. (d) The PEM of the second map after applying the Radon transform. A few of the matching edges are marked with colored stars. (e) The final fused map using the proposed PGVD and entropy filter.

stars of the same color. Finally, Figure 15(e) shows the final maps generated using the proposed entropy filter.

Comparison

As mentioned, a major benefit of edge matching for map fusion is the low processing time requirement. The proposed method is compared with adaptive random walk (ARW) map merging [4] and map segmentation [30] methods.

Table 1 summarizes the comparison of the processing times and verification for the data set and real-world experiments. To make an accurate comparison, all algorithms and computations were implemented and performed on a Core2Duo 2.66-GHz laptop. As the results show, the proposed method operates at least eight times faster, and the verification index [4] shows the accuracy of the results. The comparison results for the simulation are not provided since the other two methods are not effective for such large and complex environments.

Conclusion and Future Work

A probabilistic multiple-robot SLAM algorithm has been presented that is fast and robust. The probabilistic GVD is used to fuse maps and account for uncertainties in the occupancy

Table 1. The processing time and efficiency of three experiments.

Experiment	①	②	③	①	②	③
Method	**Processing (s)**			**Verification (%)**		
PGVD based	12	10	13	95	91	92
Map segmentation [30]	105	83	106	95	92	94
ARW map merging [4]	168	150	152	93	88	92

①: the Radish data set, ②: the two CoroBots, and ③: the three CoroBots.

grid map. The proposed method has been shown to preferentially fuse areas of the maps that have low uncertainty. Experiments and comparison with other established methods show that it is effective and 9–14 times faster.

In future work, it would be interesting to use the structure of the GVD to verify the accuracy of the map matching. If different pairs of edges are matched with high accuracy, then the structure of the GVD could allow us to determine that both sets of edges correspond to the same transformation, increasing the likelihood of correct matching.

Acknowledgment

This research is supported by the Natural Sciences and Engineering Research Council of Canada and the Canada Foundation for Innovation. We would also like to thank Jacqueline Paull for her contributions.

References

[1] S. Saeedi, L. Paull, M. Trentini, M. Seto, and H. Li, "Efficient map merging using a probabilistic generalized Voronoi diagram," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Systems*, 2012, pp. 4419–4424.

[2] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping (SLAM): Part I the essential algorithms," *IEEE Robot. Autom. Mag.*, vol. 13, no. 3, pp. 108–117, 2006.

[3] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: MIT Press, 2005.

[4] A. Birk and S. Carpin, "Merging occupancy grid maps from multiple robots," *Proc. IEEE: Special Issue Multi-Robot Syst.*, vol. 94, no. 7, pp. 1384–1387, 2006.

[5] J. L. Gould, "Honey bee cognition," *Cognition*, vol. 37, pp. 83–103, Nov. 1990.

[6] T. Guilford, S. Roberts, D. Biro, and I. Rezek, "Positional entropy during pigeon homing II: Navigational interpretation of Bayesian latent state models," *J. Theor. Biol.*, vol. 227, no. 1, pp. 25–38, 2004.

[7] H. Bunke, "Graph matching: Theoretical foundations, algorithms, and applications," in *Proc. Int. Conf. Vision Interface*, 2000, pp. 82–88.

[8] A. Howard, "Multi-robot simultaneous localization and mapping using particle filters," *Int. J. Robot. Res.*, vol. 25, no. 12, pp. 1243–1256, 2006.

[9] R. Smith, M. Self, and P. Cheeseman, "A stochastic map for uncertain spatial relationships," in *Proc. 4th Int. Symp. Robotics Research*, 1987, pp. 467–474.

[10] P. Dario and R. Chatila, *Multi-Robot SLAM with Sparse Extended Information Filters* (Springer Tracts in Advanced Robotics vol. 15). Berlin, Heidelberg, Germany: Springer, pp. 254–266, 2005.

[11] X. S. Zhou and S. I. Roumeliotis, "Multi-robot SLAM with unknown initial correspondence: The robot rendezvous case," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Systems*, 2006, pp. 1785–1792.

[12] A. Gil, O. Reinoso, M. Ballesta, and J. Miguel, "Multi-robot visual SLAM using a Rao-Blackwellized particle filter," *Robot. Auton. Syst.*, vol. 58, no. 1, pp. 68–80, 2010.

[13] D. Hahnel, W. Burgard, D. Fox, and S. Thrun, "An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Systems*, 2003, pp. 206–211.

[14] S. Thrun, "A probabilistic on-line mapping algorithm for teams of mobile robots," *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 335–363, 2001.

[15] S. Carpin, "Fast and accurate map merging for multi-robot systems," *Auton. Robot.*, vol. 25, no. 3, pp. 305–316, 2008.

[16] S. Carpin, A. Birk, and V. Jucikas, "On map merging," *Robot. Auton. Syst.*, vol. 53, no. 1, pp. 1–14, Oct. 31, 2005.

[17] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations* (Intelligent Robotics and Autonomous Agents). Cambridge, MA: MIT Press, 2005.

[18] E. R. Davies, *Machine Vision, Theory, Algorithms, Practicalities*, 3rd ed. San Francisco, CA: Morgan Kaufmann, 2005.

[19] H. Blum, "A transformation for extracting new descriptors of shape," in *Models for the Perception of Speech and Visual Form*. Cambridge, MA: MIT Press, 1967, pp. 362–381.

[20] B. Lau, C. Sprunk, and W. Burgard, "Improved updating of Euclidean distance maps and Voronoi diagrams," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Systems*, 2010, pp. 281–286.

[21] P. Beeson, N. K. Jong, and B. Kuipers, "Towards autonomous topological place detection using the extended Voronoi graph," in *Proc. IEEE/RSJ Int. Conf. Robotics Automation*, 2004, pp. 4373–4379.

[22] 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, 2001.

[23] H. Choset, S. Walker, K. Eiamsa-Ard, and J. Burdick, "Sensor-based exploration: Incremental construction of the hierarchical generalized Voronoi graph," *Int. J. Robot. Res.*, vol. 19, no. 2, pp. 125–148, 2000.

[24] J. O. Wallgrum, "Voronoi graph matching for robot localization and mapping," *Trans. Comput. Sci. IX*, 2010, vol. 6290, pp. 76–108.

[25] P. Beeson, J. Modayil, and B. Kuipers, "Factoring the mapping problem: Mobile robot map-building in the hybrid spatial semantic hierarchy," *Int. J. Robot. Res.*, vol. 29, no. 4, pp. 428–459, 2010.

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

[27] S. Friedman, H. Pasula, and D. Fox, "Voronoi random fields: Extracting the topological structure of indoor environments via place labeling," in *Proc. Int. Joint Conf. Artificial Intelligence*, 2007, pp. 2109–2114.

[28] D. Silver, D. Ferguson, A. Moms, and S. Thayer, "Feature extraction for topological mine maps," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Systems*, 2004, pp. 773–779.

[29] W. H. Huang and K. R. Beevers, "Topological map merging," *Int. J. Robot. Res.*, vol. 24, no. 8, pp. 601–613, 2005.

[30] S. Saeedi, L. Paull, M. Trentini, and H. Li, "Multiple robot simultaneous localization and mapping," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Systems*, 2011, pp. 853–858.

[31] D. Fox, W. Burgard, S. Thrun, and A. B. Cremers, "Position estimation for mobile robots in dynamic environments," in *Proc. American Association for Artificial Intelligence*, 1998, pp. 983–988.

[32] A. Howard and N. Roy. (2003). The robotics data set repository (Radish). [Online]. Available: http://cres.usc.edu/radishrepository/

Sajad Saeedi, COBRA Group, University of New Brunswick, Fredericton, Canada. E-mail: sajad.saeedi.g@unb.ca.

Liam Paull, CSAIL Lab, Massachusetts Institute of Technology, Cambridge. E-mail: lpaull@csail.mit.edu.

Michael Trentini, Defence Research and Development Canada-Suffield, Alberta, Canada. E-mail: Mike.Trentini@drdc-rddc.gc.ca.

Mae Seto, Defence Research and Development Canada-Atlantic, Nova Scotia, Canada. E-mail: mae.seto@drdc-rddc.gc.ca.

Howard Li, COBRA Group, University of New Brunswick, Fredericton, Canada. E-mail: howard@unb.ca.