Efficient Map Merging Using a Probabilistic Generalized Voronoi Diagram

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

Abstract— **Simultaneous Localization and Mapping, or SLAM, is required for mobile robots to be able to explore prior unknown space without a global positioning reference. While multiple robots can achieve the exploration task more quickly, this benefit comes with the cost of added complexity. Probabilistic 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 paper, a probabilistic version of the Generalized Voronoi Diagram (GVD), called the PGVD, is used to determine the relative transformation between maps and fuse them. The new method is effective for finding relative transformations quickly and reliably. In addition, the novel approach accounts for all map uncertainties in the fusion process.**

I. Introduction

The ability of an autonomous agent to sense its environment and situate itself within this environment is a cornerstone of mobile robotics. Simultaneous Localization and Mapping (SLAM) is the process of exploring an unknown environment and concurrently generating a consistent map of the world by fusing available sensor data. As such, the pose of the robot can be estimated and the environment map can be built [1].

Extensive literature exists on SLAM for a single mobile platform, for a review see [2]. However, exploring unknown environments with multiple mobile agents has received comparatively less attention and can have significant advantages, including:

- Exploration and mapping can be done more rapidly.
- A distributed system is more robust to failures [3].
- Results are more accurate due to redundancy of data.

With the stated benefits of multiple robot SLAM comes significant challenges in implementation. The two main problems to be overcome are map merging without a global position reference, and map fusion incorporating map uncertainty. Information within the maps must be used to achieve this fusion and scalability with map size is a significant issue.

Many proposed approaches to multiple robot SLAM exist in the literature. Feature-based multiple robot SLAM has been implemented with an information filter [4], Extended Kalman Filter [5] and fuzzy logic [6]. Another approach is to require that robots detect each other at some point during exploration to determine their relative pose [7]. However, this requirement is quite restrictive on the type of exploration algorithms that can be used. In [3], maps are merged using a random walk algorithm that is shown to be time consuming and not able to identify the case where maps have no match.

This paper presents a novel map fusion algorithm that exploits the properties of the Generalized Voronoi Diagram (GVD) to achieve fast and accurate map fusion for large maps. The GVD is built using fast morphological operations. In addition, the uncertainty in the maps is used to build a probabilistic GVD (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, edges are matched using a 2D cross correlation that will preferentially match the areas of the PGVD that have higher confidences. 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.

To summarize, the contributions of this research to multiple robot SLAM include:

- The morphological skeleton-based platform for map fusion,
- A formulation for propagating the uncertainty of the transformation to a transformed map,
- The PGVD structure for occupancy grid maps,
- The 2D cross correlation framework to match edges of the PGVD,
- Applying the entropy filter to account for uncertainty when fusing occupancy grid maps.

It will be shown that the result is that maps are fused quickly and accurately.

The rest of the paper is organized as follows: Section II presents some background and previous literature, Section III presents the proposed uncertainty propagation method, Section IV introduces the proposed method for topological multiple robot SLAM, Section V presents some experimental results from both online data sets and experiments conducted in the lab, and Section VI makes some general conclusions and discusses future work.

II. Background

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 [8]. The Generalized Voronoi Diagram (GVD), described in next sections, is an appropriate graph structure. In the subsequent, applications of multi-robot SLAM that have used graph structures are summarized.

The Generalized Voronoi Diagram (GVD) is the set of all points in the free space of a map that are equidistant from at least the two closest obstacles. 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, one-dimensional form.

In [9], topological SLAM for a single robot is proposed based on the generalized Voronoi diagram. In [10], an Annotated Generalize Voronoi Graph (AGVG) is used for single robot SLAM. The proposed method is based on a matching scheme for solving the data association problem. The 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 [11], multirobot SLAM based on topological map merging using both structural and geometrical characteristics of the Voronoi graph is proposed. The assumption in this work is that a robot will be able to recognize areas of the map that correspond to vertices.

A common problem encountered when using retractions for SLAM is that the deformation retraction is difficult to find in a closed form and heuristic methods for generating it tend be slow. In [12], a higher dimension Generalized Voronoi Graph (GVG) is proposed. The proposed GVG method is used to improve single robot SLAM in [9]. In [13], an Extended Voronoi Graph (EVG) algorithm is proposed to improve the efficiency of building the GVD when the robot is limited with its sensory horizon. In [14] a dynamic version of the GVD is proposed which improves performance near non-convex obstacles. Though this method has good results, it is time consuming. In Blum's method introduced in [15], the GVD is developed using a Maximal Inscribed Circle (MIC) method. All points of the free space for which the MIC contains at least two occupied cells are in the GVD.

In our approach, morphological operations are used to define the deformation retraction to generate the GVD as is explained in Sec. IV-C. The proposed approach is fast enough to be used in real-time and guarantees connectedness of the skeleton [16]. In addition, the probabilistic nature of the proposed PGVD allows areas of the maps that are known with higher certainty to be preferentially matched.

III. Transformation Uncertainty

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" ([2]). There is an uncertainty associated with the relative transformation matrix represented in the form of a covariance matrix. This uncertainty is mainly caused by modeling uncertainties, sensor noise and linearization effects. When a map is transformed, the covariance of the transformation should also be taken into account. The covariance error is propagated using the linearized uncertainty propagation (LUP) formulation. The LUP is a novel map merging method proposed in this paper which takes into account the uncertainty of the relative transformation between the maps during the fusion process. This section addresses the propagation of the relative uncertainty between two local maps given the transformation matrix and its covariance. First the formulation for a single cell is explained, then it is extended to cover all cells. Assume that $P_1 = \begin{bmatrix} x_1 & y_1 \end{bmatrix}^T$ is a point from map m_i that is transformed to another point, $P_2 = \begin{bmatrix} x_2 & y_2 \end{bmatrix}^T$, with a given transformation $f(\cdot)$ defined by:

$$P_2 = R_\psi P_1 + T = f(\psi, T, P_1), \tag{1}$$

where R_ψ is a 2×2 rotation matrix and T is a 2×1 translation vector:

$$R_\psi = \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix}, T = \begin{bmatrix} x_t \\ y_t \end{bmatrix} \tag{2}$$

Assume that the covariance of the transformation matrix is:

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

In general, the nonlinearity of the transformation will cause a non-Gaussian distribution. However, it is possible to approximate the non-Gaussian distribution with a Gaussian distribution using linearization as follows

$$\Sigma = F \Sigma_{tr} F^T \tag{4}$$

where Σ is the covariance after the transformation and F is the Jacobian of $f(\cdot)$ introduced in (1). By applying this approach, Σ becomes

$$\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}, \tag{5}$$

where

$$\begin{aligned} \Sigma_{11} &= \sigma^2_{xx} + 2\sigma^2_{x\psi} r_1 + \sigma^2_{\psi\psi} r_1^2 \\ \Sigma_{12} &= \sigma^2_{xy} + \sigma^2_{x\psi} r_1 + \sigma^2_{y\psi} r_2 + \sigma^2_{\psi\psi} r_1 r_2 \\ \Sigma_{21} &= \sigma^2_{xy} + \sigma^2_{x\psi} r_1 + \sigma^2_{y\psi} r_2 + \sigma^2_{\psi\psi} r_1 r_2 \\ \Sigma_{22} &= \sigma^2_{yy} + 2\sigma^2_{y\psi} r_2 + \sigma^2_{\psi\psi} r_2^2. \end{aligned} \tag{6}$$

In (6), r_1 and r_2 are defined for the simplification of Σ as

$$\begin{aligned} r_1 &= -x_1 \sin\psi + y_1 \cos\psi \\ r_2 &= -x_1 \cos\psi - y_1 \sin\psi. \end{aligned} \tag{7}$$

This relation gives the shape of the Gaussian distribution for every point. If this formulation is applied to all points of map m_i, then the resulting transformed map, m_i', will have points with their mean determined by (1) and the covariance calculated using (5). This is equivalent to the image Gaussian blur filter with varying mean and covariance defined by

$$\begin{aligned} \mu &= R_\psi \otimes m_i \oplus T \\ m_i' &= N(\mu, \Sigma) \circledast m_i, \end{aligned} \tag{8}$$

which means that m_i undergoes a Gaussian blur filter denoted by $\circledast$, where the specification of the filter is defined by N. $N(\cdot,\cdot)$ is a 2D Gaussian distribution with mean and covariance defined in its first and second arguments. The operators $\otimes$ and $\oplus$ are rotation and translation operators applied on m_i. Now m_i' can be fused with its pair map, m_i, using an entropy filter which will be explained later.

IV. Probabilistic Map Merging with the GVD

For simplicity, consider the case of two mobile robots, R_1 and R_2, equipped with laser rangers exploring an environment and building occupancy grid maps (OGM) [2]. In an OGM representation, each cell in map $m_k(i,j)$ 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.

It is convenient to represent the values in the OGM using the log odds representation of occupancy [2]:

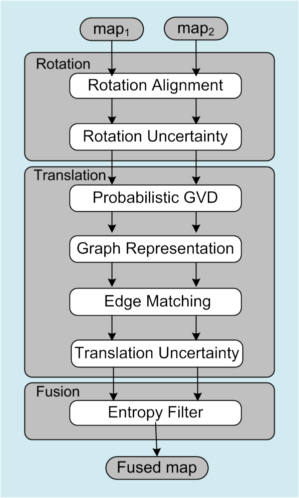
$$l_k(i,j) = \log \frac{p(m_k(i,j))}{1 - p(m_k(i,j))} \tag{9}$$

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) Uncertainty of the transformation should be taken into account in the transformation.
3) The OGM probabilities from map_2 need to be incorporated with the OGM probabilities of map_1.

The relative transformation between the maps is a result of the relative initial poses of the robots. Assuming that there is no access to a global reference, the only way to extract this transformation is by processing the maps themselves.

The proposed probabilistic skeleton method accounts for all issues.



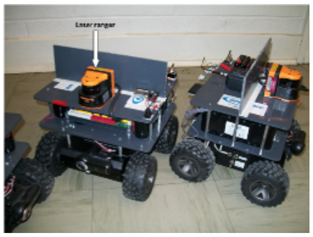


Fig. 1. (**left**) The proposed map fusion algorithm. Two input maps, map_1 and map_2, are fused by finding their relative transformation matrix. No prior information is available regarding the relative position of two respective robots, $robot_1$ and $robot_2$. (**right**) Experimental robots, CoroBots, each equipped with a laser ranger and wheel encoders.

An overview of the elements of the algorithm is shown in Fig. 1. In the first block, **Rotation Alignment**, the relative orientation between the maps is found using Radon transform. The covariance of the orientation is then calculated. The rotated map is oriented given the rotation and its covariance. This is performed in the **Rotation Uncertainty** block using (8) assuming zero translation. Then in the **Probabilistic GVD** block, the GVD is extracted while taking into account the uncertainty of the OGM . The GVD in the **Graph Representation** block is formulated mathematically. Then in the **Edge Matching** block, edges of the graphs are cross matched to find the translation. Based on the matching edges, the covariance of the translation is calculated in **Translation Uncertainty** using (8) assuming zero rotation. This covariance is used to translate the map taking into account the uncertainty of the translation. Finally in the **Entropy Filter** block, the aligned maps are fused by an entropy filter. The subsequent sections will describe each of the block components in detail.

A. Rotation Alignment

In structured environments like urban or indoor settings, the relative rotation between maps can be found easily using the Radon transform as shown by the authors in previous work [17], [18]. The Radon transform is the projection of the image intensity along a radial line oriented at a specific angle. Peak points in the Radon transform will correspond to 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.

$$\psi = \arg\min_{\theta} \mathcal{R}_{\theta=0:180}(map_1) - \arg\min_{\theta} \mathcal{R}_{\theta=0:180}(map_2), \tag{10}$$

where $\mathcal{R}_{\theta=0:180}(m_i)$ is the Radon image of the input map over the specified domain of angles (θ).

At the outputs of this preprocessing block are the two aligned maps, m_1 and m_2 given by:

$$m_1 = map_1 \tag{11}$$

$$m_2 = T_{0,0,\psi}(map_2) \tag{12}$$

where $T_{x,y,\alpha}(m)$ is a transformation matrix representing a translation by (x, y) and a rotation by α. It is assumed that both maps now have the same size: $M \times N$.

B. Rotation Uncertainty

The uncertainty in the relative rotation must be accounted for in the subsequent calculation of the relative translation. This can be done using the general formulation in (8) presented for transformation uncertainty. This means that after finding the rotation, its uncertainty on the map is propagated using (8), 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.

C. 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 probabilistic GVD (PGVD) is found for each of the two maps, m_1 and m_2. In general 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.

1) Finding the GVD using Mathematical Morphology: Mathematical morphology (MM) 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 are defined in MM like 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 [16]. Each mask is designed for a particular situation to guarantee the connectedness. The GVD is represented as a matrix, S, with the same size as the map defined as:

$$S = [s_{i,j}]_{i=1..M,j=1..N} = \begin{cases} 1 & \text{if } m(i,j) \in \text{ GVD} \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

Fig. 2-a depicts a simple environment with the occupied cells and the GVD, S.

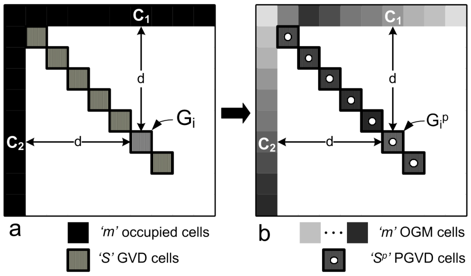


Fig. 2. **a)** The GVD generated from the OGM. **b)** The PGVD contains probabilistic information based on the probabilities in the OGM

2) Finding the Probabilistic GVD: The skeleton developed by this 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, coined the probabilistic GVD (PGVD) is built that combines the structure of the GVD with the probabilistic nature of the OGM and is shown in Fig. 2-b.

Definition 1: 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.

For example in Fig. 2-a, c_1 and c_2 are contact points of G_i.

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 2 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 2 occupied contact points based on their probabilities of occupancy.

To do so, another RV $N_{G_i^p}$ is defined which 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:

$$P(N_{G_i^p} = k) = f(n_i, k). \tag{14}$$

The function $f(n_i, k)$ can be defined recursively using the following:

$$f(n_i, k) = p(c_{n_i})f(n_i - 1, k - 1) + (1 - p(c_{n_i}))f(n_i - 1, k). \tag{15}$$

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

$$f(n_i, n_i) = \prod_{j=1}^{n_i} p(c_j) \tag{16}$$

$$f(n_i, 0) = \prod_{j=1}^{n_i} (1 - p(c_j)) \tag{17}$$

We will proceed to prove that this function $f(n_i, k)$ provides the valid probability density function for the RV $N_{G_i^p}$

Theorem 1:

$$\sum_{k=0}^{n} f(n, k) = 1 \tag{18}$$

Proof: By induction:

Base case: $n = 1$

$$\sum_{k=0}^{1} f(1, k) = f(1, 0) + f(1, 1) = p(c_1) + (1 - p(c_1)) = 1 \tag{19}$$

from (16) and (17).

Induction step:

Assume true for $n - 1$:

$$\sum_{k=0}^{n-1} f(n - 1, k) = 1 \tag{20}$$

Consider n:

$$\begin{aligned} & \sum_{k=0}^{n} f(n, k) \\ = \;& f(n, 0) + f(n, 1) + ... + f(n, n - 1) + f(n, n) \\ = \;& -p(c_n)f(n - 1, 0) + f(n - 1, 0) \\ + \;& p(c_n)f(n - 1, 0) - p(c_n)f(n - 1, 1) + f(n - 1, 1) \\ + \;& p(c_n)f(n - 1, 1) - p(c_n)f(n - 1, 2) + f(n - 1, 2) \\ + \;& p(c_n)f(n - 1, 2) - p(c_n)f(n - 1, 3) + f(n - 1, 3) \\ + \;& ... \\ + \;& p(c_n)f(n - 1, n - 1) \\ = \;& f(n - 1, 0) + f(n - 1, 1) + ... + f(n - 1, n - 1) \\ = \;& \sum_{k=0}^{n-1} f(n - 1, k) = 1 \end{aligned}$$

Induction is complete. ■

Corollary 1: The number of additive terms in $\sum_{k=0}^{n_i} f(n_i, k)$ is 2^{n_i}

Proof: The number of additive terms in $f(n_i, k)$ is

$$\left(\begin{array}{c} n_i \\ k \end{array} \right) = \frac{n_i!}{k!(n_i - k)!} \qquad (21)$$

It is also well known from combinatorial analysis by summing the entries in Pascal's triangle that:

$$\sum_{k=0}^{n_i} \left(\begin{array}{c} n_i \\ k \end{array} \right) = 2^{n_i} \qquad (22)$$

■

Therefore $f(n_i, k)$ is a valid probability density function that produces all possible combinations of contact points and sums to 1.

$p(G_i^p)$ is now given by:

$$\begin{aligned} p(G_i^p) &= p(N_{G_i^p} \geq 2) = \sum_{k=2}^{n_1} p(CN_i = k) \\ &= 1 - f(n_i, 0) - f(n_i, 1) \qquad (23) \end{aligned}$$

The PGVD is now defined as:

$$S^p = \bigcup_{i=1..s} G_i^p \qquad (24)$$

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

This new probabilistic topological representation of the environment will be shown in subsequent sections to have significant advantages.

D. 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. Fig. 3-a shows a sample skeleton with vertices marked with 'v'. The set of all vertex points is defined as V.

To identify edges in the graph, a dilation mask is applied to each vertex in V. The dilation mask is shown in Fig. 3-b. The result of the dilation operation is a new map D (Fig. 3-b) that contains the dilated vertices. The edge matrix E is found as $S^p - D$, as shown in Fig. 3-c. This operation is performed on each of the two GVDs to produce two edge maps: $E_k, k = \{1, 2\}$.

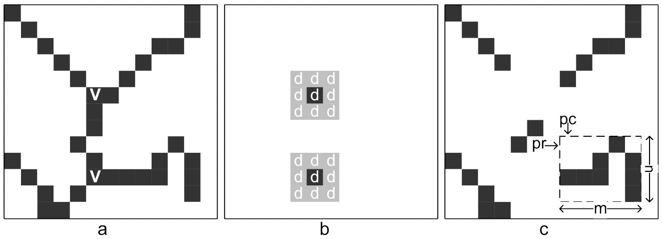


Fig. 3. **a)** A sample GVD with vertices marked by a 'v' **b)** Vertices are dilated by the morphological Dilation operator to extract edges **c)** Edges of the GVD. Cells bounded by the dashed line depict the bounding matrix of an edge.

E. Edge Matching for Translation Alignment

The probabilistic edge matrices (PEM), 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. Edges with short lengths are removed to avoid processing short edges which have a higher chance of producing false positives. The edges of each probabilistic edge matrix are matched using a 2D cross correlation. To speed up the cross correlation, each edge from each edge matrix is represented as a sub-matrix of E where its size is such that it is as small as possible while still containing the entire edge. Fig. 3-c shows an edge bounded by a dashed boundary line.

For a given edge matrix, E_k^p with N_k edges, each edge, $e_k^{i_k}$, $i_k = 1..N_k$ is given by (refer to Fig. 3-c):

$$e_k^i = E_{k(pr_i:pr_i+m_i, pc_i:pc_i+n_i)}^p \qquad (25)$$

To perform matching, each edge of E_1^p is cross correlated with each edge of E_2^p.

The 2D 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 defined as

$$\begin{aligned} \mathcal{E}^{(i_1, i_2)} &= [\epsilon_{i,j}]_{i=0,\ldots,m_1+m_2-1, j=0,\ldots,n_1+n_2-1} \qquad (26) \\ &= e_1^{i_1} \star e_2^{i_2} = \sum_{k=0}^{m-1} \sum_{l=0}^{n-1} e_1^{i_1}(k, l) e_2^{i_2}(k + i, l + j) \end{aligned}$$

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 2D 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 the all possible combinations of translations between the two edge matrices. This value is computed using (27) for every combination of edges: $i_1 = 0..N_1$ and $i_2 = 0..N_2$.

Then we can define the similarity matrix, Γ_1^2, between E_1^p and E_2^p as:

$$\Gamma_1^2 = [\gamma_{i_1 i_2}]_{i_1=1..N_1; i_2=1..N_2} = \max\left(\mathcal{E}^{(i_1, i_2)}\right) \qquad (27)$$

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

$$[i_1^*, i_2^*] = \arg\max(\Gamma_1^2), \qquad (28)$$

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 which matched these two edges using the following relation [17]:

$$T_{x,y,0} = \begin{bmatrix} \delta_x \\ \delta_y \\ 0 \end{bmatrix} = \begin{bmatrix} \mu_x(e_1^{i_1^*}) - \mu_x(e_2^{i_2^*}) \\ \mu_y(e_1^{i_1^*}) - \mu_y(e_2^{i_2^*}) \\ 0 \end{bmatrix} \qquad (29)$$

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.

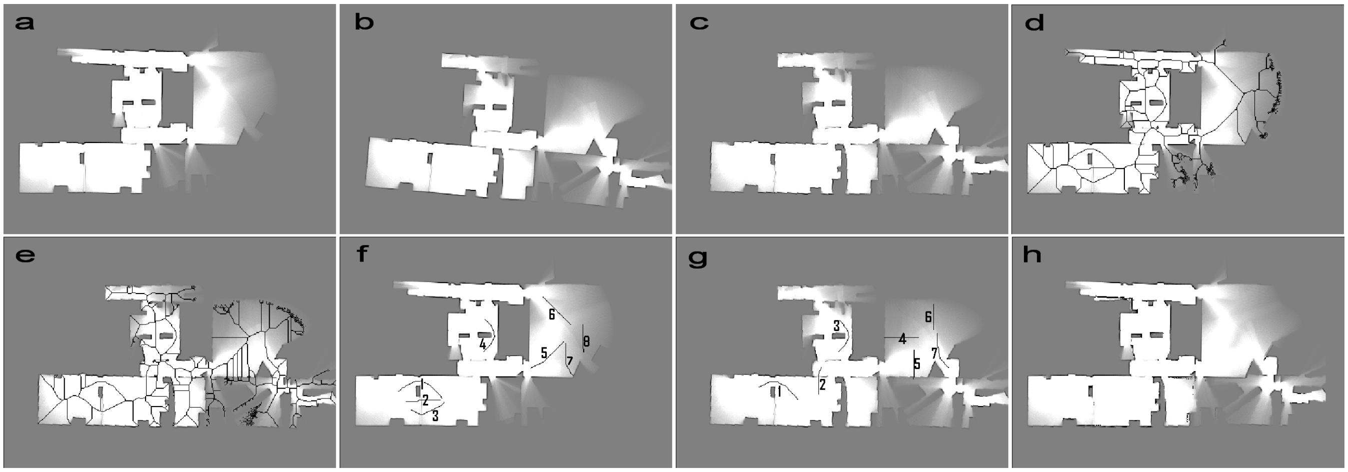


Fig. 4. **a**) map_1, **b**) map_2, **c**) map_2', rotated map_2, **d**) PGVD of map_1, **e**) PGVD of map_2', **f**) PEM of map_1, **g**) PEM of map_2', **h**) Fused map.

The final transformed map is defined as:

$$m_2^T = T_{x,y,0}(T_{0,0,\psi}(map_2)) = T_{x,y,\psi}(map_2) \qquad (30)$$

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. Edges with cells with higher confidence of being true GVD cells will result in higher cross-correlation values.

1) Existence of Solution: False matches can be identified by analyzing the expected results of the 2D 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 2D cross correlation results in L matching cells. Define the subset of cells in $e_k^{i_k}$ that were matched as $\varepsilon_k^{i_k}$. We can define the average confidence for the matched edge according to:

$$\alpha_k^{i_k} = \frac{1}{L}\sum_{l=1..L} p(\varepsilon_k^{i_k}) \qquad (31)$$

By expansion of (27), $\max(\mathcal{E})$ 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 as a candidate match:

$$\max(\mathcal{E}^{(i_1,i_2)}) > \rho\alpha_1^{i_1}\alpha_1^{i_2}\min(l_1, l_2), \qquad (32)$$

where ρ is a desired matching percentage. If there is no element in the Γ_1^2 matrix to satisfy (32), then there is no matching between the two maps.

F. Translation Uncertainty

The transformed map which included the effect of the rotation uncertainty was used to find the translation. After finding the translation, (8) is used again to propagate the uncertainty of the translation, assuming zero rotation and $\sigma^2_{\psi\psi} = 0$. We now have a fully probabilistic representation of the required transformation and can proceed to fuse the maps.

G. 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 that is 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 (9).

$$l_{fused}(i,j) = l_1(i,j) + l_2(i,j) \qquad (33)$$

for all $i = 1..N, j = i..M$. The probabilities can then be recovered from the log odds representation using:

$$p(m_{fused}(i,j)) = 1 - \frac{1}{exp\{l_{fused}(i,j)\}} \qquad (34)$$

The entropy filter is applied to the fused map, m_{fused} [19]. The entropy filter compares the original map, m_1 and the fused map m_{fused} and reject updates that resulted in higher entropy.

For the case of a discrete binary RV, such as each cell of the OGM, $m(i,j)$ with $p(m(i,j)) = p_{ij}$, the entropy can be described by [20]:

$$H(m(i,j)) = -p_{ij}\log p_{ij} - (1-p_{ij})\log(1-p_{ij}) \qquad (35)$$

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}:

$$I_{ij} = H(m_1(i,j)) - H(m_{fused}(i,j)) \qquad (36)$$

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

$$m_{final}(i,j) = \begin{cases} m_{fused}(i,j) & I_{ij} \geq 0 \\ m_1(i,j) & I_{ij} < 0 \end{cases} \qquad (37)$$

where only values from m_{fused} which resulted in positive information are kept.

V. Experimental Results

To demonstrate the effectiveness of the proposed method, three experiments are presented. The first one is performed on a well known data set for multiple robot SLAM. In this experiment all the details of the proposed algorithm are explained. The second experiment is another real world

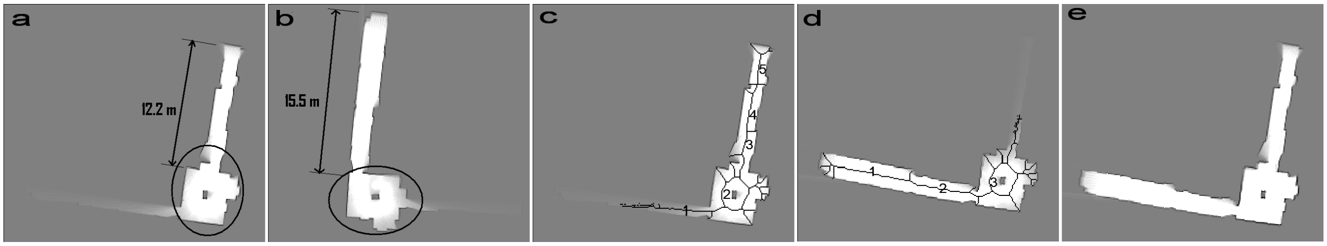


Fig. 5. **a)** map_1, **b)** map_2, **c)**, PGVD and PEM of map_1 **d)**, PGVD and PEM of rotated map_2 **e)** Fused maps.

experiments performed with two CoroBots in an indoor environment in the basement of the University of New Brunswick (UNB). Finally, the last experiment shows the map merging with three maps of the Department of Electrical and Computer Engineering of UNB. 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 Fig. 1. SLAM on the individual robots is performed with the particle filter implementation presented in [7]. The same particle filtering is used to produce the maps from the data set experiment.

A. *Experiment 1: RADISH Dataset*

The first experiment is performed on RADISH Fort AP Hill dataset [21]. The raw laser ranger and encoder data are recorded in a standard Player format [22]. Fig. 4-a and Fig. 4-b show two input maps before alignment, map_1 and map_2. Fig. 4-c shows m_2, map_2 after applying alignment method by Radon transform. The alignment is 5.5^o. Fig. 4-d and Fig. 4-e show the GVDs of m_1 and m_2. Fig. 4-f and Fig. 4-g show filtered PEMs of m_1 and m_2. The Γ_1^2 matrix is shown in (40). From (28), $i^* = 4$ 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 (29). The final transformation vector is $T_{x,y,\psi} = [-25, -1, 5.5^o]^T$.

From (31), $\alpha_1^4 = 0.86$ 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 which represents matching confidence, was chosen to be 95%. According to (32)

$$(0.95)(0.86)(0.89)\min(44, 45) = 31.99 < 32.9, \qquad (38)$$

therefore the match is valid. Note that the match of e_1^1 with e_2^1, represented in $\Gamma_1^2(1,1)$ is disqualified because the associated 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.

$$(0.95)(0.91)(0.92)\min(56, 62) = 44.54 > 31.6. \qquad (39)$$

$$\Gamma_1^2 = \begin{bmatrix} 31.6 & 2.0 & 6.4 & 6.1 & 1.9 & 0.8 & 5.4 \\ 12.9 & 1.9 & 2.0 & 11.7 & 0.9 & 0.5 & 1.6 \\ 11.9 & 2.8 & 5.4 & 3.0 & 1.4 & 0.5 & 2.5 \\ 10.7 & 6.7 & \boxed{32.9} & 1.0 & 5.3 & 1.8 & 8.1 \\ 9.1 & 1.9 & 8.9 & 2.3 & 0.9 & 0.5 & 1.6 \\ 5.9 & 0.4 & 2.1 & 0.4 & 0.4 & 0.2 & 1.9 \\ 3.2 & 9.9 & 3.9 & 0.5 & 11.6 & 4.0 & 6.9 \\ 0.9 & 7.2 & 2.9 & 0.3 & 10.2 & 3.5 & 4.1 \end{bmatrix} \qquad (40)$$

To show the benefit of the probabilistic approach, the Γ_{det} is shown in (41). In this case, matching is performed directly on the GVD instead of the PGVD. 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. Fig. 4-h shows the final aligned maps.

$$\Gamma_{det} \begin{bmatrix} 38 & 2 & 9 & 10 & 2 & 2 & 9 \\ 17 & 2 & 2 & 30 & 2 & 1 & 2 \\ 16 & 3 & 6 & 5 & 3 & 2 & 3 \\ 11 & 9 & \boxed{40} & 1 & 10 & 8 & 12 \\ 14 & 2 & 10 & 3 & 2 & 1 & 2 \\ 14 & 1 & 7 & 1 & 1 & 1 & 8 \\ 6 & 29 & 9 & 1 & 29 & 29 & 25 \\ 3 & 29 & 9 & 1 & 32 & 32 & 21 \end{bmatrix} \qquad (41)$$

B. *Experiment 2: two CoroBots*

In this experiment, two differential-wheeled robots are used. Fig. 5-a and Fig. 5-b show the OGMs built by the robots. The total area is $81.02m^2$ and the trajectories were $22.5m$ and $25.9m$. The area of this experiment is not very large, but it is important because the overlap between two maps is approximately $25.4m^2$ which is 31% of the total area as shown by the ellipses. This shows the effectiveness of the algorithm with small overlaps. In both maps, there are non-overlapping corridors with almost the same size ($15.5m$ and $12.2m$). But the algorithm is capable of rejecting them as matching and finding the transformation based on the overlap. Fig. 5-c and Fig. 5-d depict the PGVD of the aligned maps with selected edges for matching marked with numbers. The edges marked with number 2 in Fig. 5-c and number 3 in Fig. 5-d are used to calculate the translation. Finally, Fig. 5-e shows the final fused map applying the proposed method.

C. *Experiment 3: three CoroBots*

This experiment is performed in a larger environment, with a area of approximately $600m^2$ and three agents are involved. The trajectories of the robots are approximately $60m$, $35m$ and $55m$. By merging the maps, loop closure happens successfully. Fig. 6-a, b and c show the three local maps. Maps of Fig. 6-b and c are fused to Fig. 6-a. The overlapping region is indicated in the figure. The fused global map of the environment is shown in Fig. 6-d.

D. *Comparison*

As mentioned, a major benefit of edge matching for map fusion is the low processing time requirement. The proposed

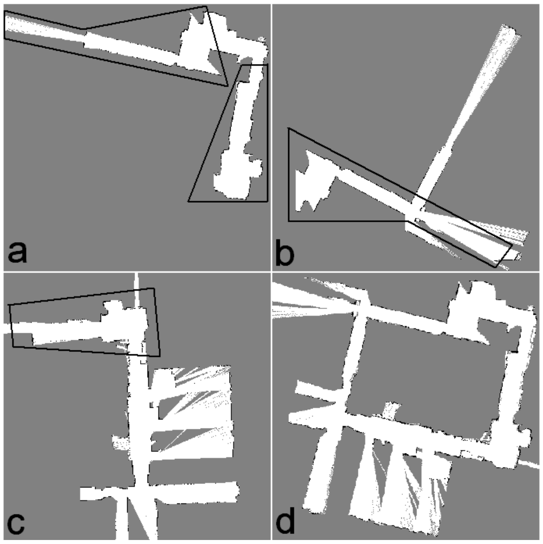


Fig. 6. Three partial maps are fused together to generate a global map. (**a**) is the base map where (**b**) and (**c**) are fused to that. Overlaps are marked with polygons. (**d**) Final fused map which depicts loop closure.

method is compared with Adaptive Random Walk (ARW) map merging [3] and Map Segmentation [17] methods.

ARW is based on search and verification. In ARW, the similarity indices of two maps are calculated given a set of known transformations. The transformation corresponding to the best similarity index is selected as a start point of an adaptive random walk for accurate merging. Map segmentation is based on extracting thin lines from objects and walls of both maps and matching them. Segments are selected for cross correlation matching by a histogram filter.

Table I summarizes the comparison of the processing time and verification for all three experiments. As the results show, the proposed method operates at least eight times faster and the verification index [3] shows the accuracy of the results.

TABLE I

PROCESSING TIME AND EFFICIENCY OF THREE EXPERIMENTS, ① RADISH DATA SET, ② TWO COROBOTS, ③ THREE COROBOTS.

Experiment	①	②	③	①	②	③
Method	Processing (sec)			Verification (%)		
pgvd based	12	10	13	95	91	92
Map segmentation [17]	105	83	106	95	92	94
ARW map merging [3]	168	150	152	93	88	92

VI. CONCLUSION AND FUTURE WORK

A probabilistic multiple robot SLAM algorithm has been presented that is fast and robust. The probabilistic Generalized Voronoi Diagram is used to fuse maps and account for uncertainties in the occupancy 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 to 14 times faster.

In future work, it would be desirable 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.

ACKNOWLEDGEMENT

This research is supported by Natural Sciences and Engineering Research Council of Canada (NSERC) and Canada Foundation for Innovation. The authors would also like to thank Jacqueline Paull for her contributions.

REFERENCES

[1] H. D. Whyte and T. Bailey, "Simultaneous localization and mapping (slam): Part i the essential algorithms," *IEEE Robotics and Automation Magazine*, vol. 13, no. 3, pp. 108–117, 2006.

[2] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, Massachusets, USA: The MIT press, 2005.

[3] A. Birk and S. Carpin, "Merging occupancy grid maps from multiple robots," *it Proceedings of the IEEE: Special Issue on Multi-Robot Systems*, vol. 94, no. 7, pp. 1384–1387, 2006.

[4] S. Thrun and Y. Liu, "Multi-robot slam with sparse extended information filers," *Springer Tracts in Advanced Robotics*, vol. 15, pp. 254–266, 2005.

[5] X. S. Zhou and S. I. Roumeliotis, "Multi-robot slam with unknown initial correspondence: The robot rendezvous case," in *Intelligent Robots and Systems (IROS), Proceedings of the IEEE/RSJ International Conference on*, 2006, pp. 1785–1792.

[6] K. LeBlanc and A. Saffiotti, "Multirobot object localization: A fuzzy fusion approach," *IEEE Transactions on Systems, Man and Cybernetics-Part B*, vol. 39, no. 5, pp. 1259–1276, October 2009.

[7] A. Howard, "Multi-robot simultaneous localization and mapping using particle filters," *International Journal of Robotics Research*, vol. 25, no. 12, pp. 1243–1256, December 2006.

[8] H. Bunke, "Graph matching: Theoretical foundations,algorithms, and applications," in *International Conference on Vision Interface*, 2000, pp. 82–88.

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

[10] J. O. Wallgrum, "Voronoi graph matching for robot localization and mapping," *Transactions on computational science IX*, vol. 6290, pp. 76–108, 2010.

[11] W. H. Huang. and K. R. Beevers, "Topological map merging," *The International Journal of Robotics Research*, vol. 24, no. 8, pp. 601–613, 2005.

[12] H. Choset, S. Walker, K. Eiamsa-Ard, and J. Burdick, "Sensor-based exploration: incremental construction of the hierarchical generalized voronoi graph," *The International Journal of Robotics Research*, vol. 19, no. 2, pp. 125–148, 2000.

[13] P. Beeson, N. K. Jong, and B. Kuipers, "Towards autonomous topological place detection using the extended voronoi graph," in *in Proc. of the IEEE Int. Conference on Robotics and Automation (ICRA)*, 2004.

[14] B. Lau, C. Sprunk, and W. Burgard, "Improved updating of euclidean distance maps and voronoi diagrams," in *in Proc. IEEE Int. Conference on Intelligent Robots and Systems (IROS)*, 2010.

[15] H. Blum, "A transformation for extracting new descriptors of shape," *Models for the Perception of Speech and Visual Form*, pp. 362–381, 1967.

[16] E. R. Davies, *Machine Vision, Theory, Algorithms, Practicalities, 3rd Edition*. Morgan Kaufmann, 2005.

[17] S. Saeedi, L. Paull, M. Trentini, and H. Li, "Multiple robot simultaneous localization and mapping," in *Intelligent Robots and Systems (IROS), Proceedings of the IEEE/RSJ International Conference on*, 2011.

[18] ——, "A neural network-based multiple robot simultaneous localization and mapping," in *Intelligent Robots and Systems (IROS), Proceedings of the IEEE/RSJ International Conference on*, 2011.

[19] D. Fox, W. Burgard, S. Thrun, and A. B. Cremers, "Position estimation for mobile robots in dynamic environments," in *In Proc. of the American Association for Artificial Intelligence (AAAI)*, 1998.

[20] A. Papoulis and S. U. Pillai, *Probability, Random Variables and Stochastic Processes*, 4th ed. McGraw Hill, 2002.

[21] [Online]. Available: http://cres.usc.edu/radishrepository/

[22] R. Vaughan, B. Gerkey, and A. Howard, "The player/stage project: Tools for multi-robot and distributed sensor systems," in *Proceedings of the International Conference on Advanced Robotics (ICAR 2003)*, jul. 2003, pp. 317–323.