Map Merging for Multiple Robots Using Hough Peak Matching

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

Abstract

Navigation in a GPS-denied environment is an essential requirement for increased robotics autonomy. While this is in some sense solved for a single robot, the next challenge is to design algorithms for a team of robots to be able to map and navigate efficiently.

The key requirement for achieving this team autonomy is to provide the robots with a collaborative ability to accurately map an environment. This problem is referred to as cooperative simultaneous localization and mapping (SLAM). In this research, the mapping process is extended to multiple robots with a novel occupancy grid map fusion algorithm. Map fusion is achieved by transforming individual maps into the Hough space where they are represented in an abstract form. Properties of the Hough transform are used to find the common regions in the maps, which are then used to calculate the unknown transformation between the maps.

Results are shown from tests performed on benchmark datasets and real-world experiments with multiple robotic platforms.

Keywords: Simultaneous Localization and Mapping (SLAM), Multiple Robot, Map Merging, Hough Space and Image Entropy.

1 Introduction

Simultaneous localization and mapping, or SLAM, first proposed by Smith and Cheeseman [35] and later improved by Leonard and Durrant-Whyte [27] and many others since has become a cornerstone of mobile robotics. The ability for a mobile platform to build a map and situate itself within that map without any prior knowledge of the environment has meant great improvements in robotics autonomy. In recent years, robotics research has been trending towards multi-agent systems - systems of multiple robotic agents that can coordinate and cooperate to achieve a task. For example, for a common

*COBRA Group at the University of New Brunswick, Fredericton, Canada, http://www.ece.unb.ca/COBRA/, {sajad.saeedi.g, liam.paull, howard}@unb.ca

†Defence Research and Development Canada-Suffield, Alberta, Canada,Mike.Trentini@drdc-rddc.gc.ca

‡Defence Research and Development Canada-Atlantic, Nova Scotia, Canada, mae.seto@drdc-rddc.gc.ca

task such as robotics exploration, it is necessary that robots be able to transmit and fuse their relative world maps in real-time. This ability will allow the agents to make control decisions based upon the global map and result in the task being completed more efficiently.

This is challenging because each agent is building a local map in its own local coordinates. It is cumbersome and sometimes unrealistic to assume that the robots have knowledge of their relative poses either from prior knowledge or from encountering each other in the environment. It is more desirable for the relative poses to be determined from the maps themselves to provide maximum flexibility of the solution.

As will be described, some past approaches to solving this problem have been based on feature-based SLAM, which is problematic because it relies on extracting features from measurements and is prone to information loss. Some approaches have used view-based SLAM, however, they require the robots to meet in the environment [21] or are too computationally expensive to be usable in real-time [2].

This paper presents a novel method for finding these unknown relative transformations using the Hough transform that will be shown to be fast and accurate. Individual robots perform a robust and accurate view-based SLAM using dense laser ranger measurements [16] to generate a consistent local map of the environment. At the multi-robot level, the map merging process is performed in two steps. First, geometric features are extracted from the occupancy grid maps built by the individual robots performing view-based SLAM. Second, these features are matched across maps and used to find the relative transformation. For feature extraction, we propose to use line extraction based on the Hough transform. The advantage of this line extraction method for this task over other methods such as RANSAC and split-merge is that we can more easily and robustly perform the feature association in the Hough space rather than the Euclidean space. Finding the relative rotation and translation of the maps is executed by matching the peaks in the Hough space which correspond to lines in Euclidean space. This parameter estimation problem can be solved with a separate instantiation of a RANSAC-like algorithm, however, we show that it can be done more efficiently and robustly using properties of the Hough transform. Notably, tracking multiple hypotheses, a known deficiency of RANSAC, is important for many environments that contain many perpendicular lines. The result is a robust and efficient algorithm that scales linearly with map size.

Simulation results based on published datasets as well as real world experimental data are used to validate the methods.

The major contributions of this work are listed as follows:

- Multiple hypothesis: generally, in polygonal environments, multiple hypotheses for the rotation alinement exist, due to the similarities in the environment. Processing all these hypotheses requires high computational demand. In this work, a solution based on the Hough transform is proposed which generates only one rotation hypothesis, which helps to save processing resources.

- Peak matching: To the best of our knowledge, this paper is the first paper that applies Hough peak matching to map merging. In the literature, the Hough transform has been applied to localization and scan matching; however, applying

the Hough transform to map merging requires additional problems to be solved, which will be explained later.

- Effectiveness: it is shown expopeerimentally that the available map merging solutions, which are solely based on the correlation of the Hough images, can not produce acceptable results if there is little overlap between the maps. The proposed method overcomes this problem by effectively identifying the overlap between the maps.

The next section will present a more detailed review of the existing literature in the field of multiple robot feature-based and view-based SLAM. Section 3 explains the problem and requirements. Section 4 will outline the proposed methods based on the Hough transforms. Section 5 will present the results. Section 6 will discuss and explain the comparison results and finally some conclusions are presented in Section 7.

2 Background of Research

As mentioned, many past approaches to multi-robot SLAM have used a feature-based world representation where maps are represented by distinguishable objects. For example, in [37] a feature-based multiple robot SLAM is proposed which is based on the information filter. In [42] a classic EKF-based solution is proposed. In [14] a feature based visual SLAM using particle filtering is proposed. A feature-based multiple robot localization is proposed in [26] where fuzzy sets are used to represent uncertain position information and fuzzy intersection is used to fuse data. In [1] an algorithm for merging feature-based maps with limited communication is introduced. In [25] a feature-based map merging for multiple robots under limited communication is proposed where determining and extracting good features remains as future work. While the feature-based paradigm has been shown to be efficient and effective in certain environments, in general it is not. By extracting features, there is possible loss of data. Thrun et al. state that "a lot of information is sacrificed by using features instead of full measurement vector. This lost information makes certain problems more difficult, such as data association problem of determining whether or not the robot just revisited a previously explored location." [38]. Since onboard processing has become so powerful, there is no longer a need to extract features from sensor data and potentially lose useful information in the process. Especially this applies for dense range sensors such as laser rangers.

2.1 View Based SLAM and Map Merging

View-based SLAM differs from feature-based SLAM in that no-features are explicitly extracted [18]. Instead, entire scans are matched using scan matching algorithms [28] and [5]. (Fig. 1)

In [21] a view-based multi-robot SLAM algorithm is proposed using a particle filter. In their method, it is assumed that robots will meet each other at a point. At the meeting point robots know their relative positions and from this point a particle filter is applied to the data in reverse temporal sequence to make a global map. The proposed

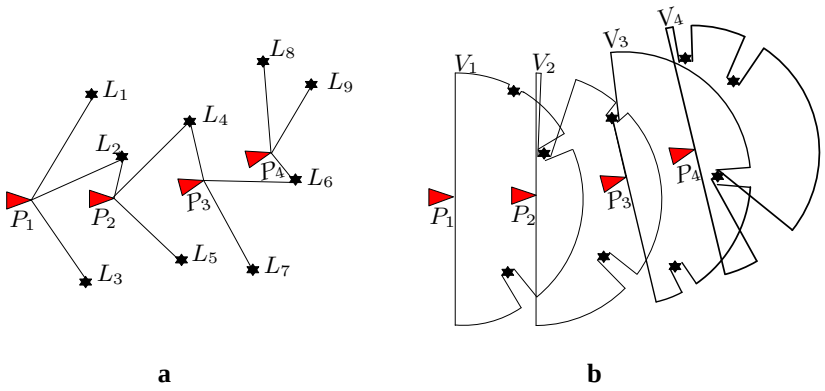

Figure 1: P_1 to P_4 are poses of a robot at four time instances. (a) Feature-based SLAM. L_1 to L_9 are detected features used for localization. (b) View-based SLAM. V_1 to V_4 are corresponding views. No features are extracted, instead views are matched for localization.

method finds the initial state of the robots but it cannot be applied when robots cannot see each other or the relative poses are not known. In [22] a similar method but with a central agent and 80 robots is used. The method proposed in [21] is moderately fast but demands high computational power and memory since it is based on particle filtering.

In [36] a probabilistic multiple robot view-based SLAM is proposed. The limitation with this method is that the initial pose of the robots are assumed to be known approximately prior to the start of the mission.

An effective and fast approach to multiple robot SLAM rests on the concept of the map merging or map fusion. In [2] 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 main drawback of this method is that it usually fails especially when there are fewer similar patterns in both maps. The method proposed in [13] is based on the solution proposed in [2], except that individual robots are using Distributed Particle SLAM (DP-SLAM) version 2.0 [10] for filtering. The method proposed in [2] and used in [13] is a highly time consuming algorithm. This is problematic especially for the large scale maps which are a typical problem in indoor environments. In [13] maps are updated by applying a simple rule: the minimum occupancy grid value between the two maps is used for merging. For example an occupied cell in one map is assumed to be occupied in both maps. A similar method is proposed in [4] with simulated annealing and hill climbing used to merge maps. This method also becomes ineffective in maps with less overlaps.

In [17], multiple robot SLAM based on topological map merging using both structural and geometrical characteristics of the Voronoi graph is proposed. In this case the topological map is built on the occupied space as opposed to the free space. The assumption in this work is that a robot will be able to recognize areas of the map that correspond to vertices. The method in [17] is claimed to be fast. However, a limitation is that the map and pose are not updated.

As will be described, the Hough transform is an alternate way to represent the geometrical data within the map that has some useful advantages. This approach has been used for map merging in the past [3] where the rotation and translation are derived in separate steps similar to our approach. However, as is shown in the experimental results section, the main drawback of their solution is that it fails when there is not enough overlap between the input maps. Specifically, their approach to finding the relative translation depends on a map projection. If there is insufficient shared regions in projections, the proposed translation finding method produces wrong results. Our experimental results show that our method improves upon past methods in this respect.

It should be emphasized that the accuracy of any multiple robot SLAM solution which is based on the map merging, depends on the accuracy of the individual local maps of the robots. Therefore to have a better and consistent global map, it is important to use a robust SLAM solution to generate local maps. In this research individual robots are performing SLAM using an improved grid mapping [16].

2.2 The Hough Transform

Performing map merging requires finding overlaps between maps and this usually requires a search operation. Generally this is a difficult task due to:

- **Modeling:** A mathematical representation for objects in maps is necessary to compare objects.
- **Partial views:** Map features from different view angles may look different.
- **Geometric relations:** The internal relation of map features is an inherent property that makes each map unique.

Performing this search in Euclidean space is difficult. However, by transforming the map into a different representation, certain image properties can be exploited to make the problem easier. The Hough transform turns out to have many such properties. For example, line segments, which are common in most structured environments, are modeled as an intensity point in the Hough space. Relative relations of line segments are represented by the distances between intensity points. Additionally, since the Hough transform is generated over an angle range of 360°, then all views in the Hough space are included and the problem of partial views can be eliminated.

There are a variety of methods to extract geometric information like line segments from maps or laser ranger measurements [29], [19]. The Hough transform [8] and the Radon transform [32] are two classical methods for line extraction which have different applications. The results of the Hough and the Radon transforms on a map are almost identical, although the approach of these two methods are different. The Radon transform is the projection of the image intensity along a radial line oriented at a specific angle, while the Hough transform is a parametric model of each non-background point.

Other methods like Split-Merge [9] and RANSAC [12] are fast and effective for line extracting, but because of interesting properties of the Hough transform, in this work we will primarily use the Hough transform. The details of this construct are presented here.

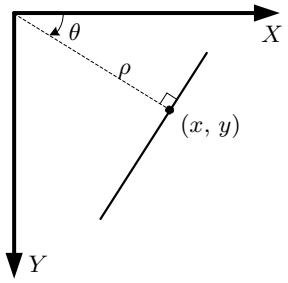

Figure 2: Hough modeling.

The basic form of the Hough transform, as shown in Fig 2, maps every (x, y) point from the image space into the (θ, ρ) space, known as the Hough space, based on the following relation:

$$\rho = xcos\theta + ysin\theta. \tag{1}$$

As a result, each (x, y) point is represented as a sinusoid in the Hough space. The sine waves of a set of points belonging to a line segment all intersect at a point in the Hough space. Line segments in the image space with different lengths are mapped to points with different intensities proportional to the length of the lines. The image generated from the Hough transform of all points of the map is called the Hough image. Since the Hough image is periodic, it is sufficient to consider only a limited range of angles: $\theta \in [-\pi/2, \pi/2)$. Three useful properties of the Hough image are described here:

Property 1: If a map, $m = map(x, y)$ is transformed to $m' = map'(x, y)$ by a rotation matrix R_ψ and a translation vector $T = \begin{bmatrix}\delta_x & \delta_y\end{bmatrix}^T$, then the coordinates of the Hough space of the corresponding Hough images, $\mathcal{H}(\theta, \rho)$ and $\mathcal{H}'(\theta', \rho')$ will have the following relation [23] (the superscript ^{T} denotes the matrix transpose):

$$\theta' = \theta + \psi \tag{2}$$

$$\rho' = \rho + \begin{bmatrix}\cos(\theta + \psi) & \sin(\theta + \psi)\end{bmatrix} T, \tag{3}$$

that is

$$m' = R_\psi \otimes m \oplus T \Rightarrow$$
$$\mathcal{H}'(\theta, \rho) = \mathcal{H}(\theta + \psi, \rho + \begin{bmatrix}\cos(\theta + \psi) & \sin(\theta + \psi)\end{bmatrix} T).$$

The operators $\otimes$ and $\oplus$ are rotation and translation operators applied on m. R_ψ and T are rotation matrix and translation vector. Fig 3 shows this property of Hough image.

Property 2: If $m = m_1 + m_2$ then the superposition relation applies to Hough images: $\mathcal{H}_m(\theta, \rho) = \mathcal{H}_{m_1}(\theta, \rho) + \mathcal{H}_{m_2}(\theta, \rho)$.

Property 3. A line segment with the length of L given by $y = ax + b$ in Cartesian coordinates, results in a point with intensity of L in the Hough space located at the point (θ, ρ) given by:

$$\theta = \arctan a - 90^\circ \tag{4}$$

$$\rho = b \sin\theta \tag{5}$$

$$\mathcal{H}(\theta, \rho) = L. \tag{6}$$

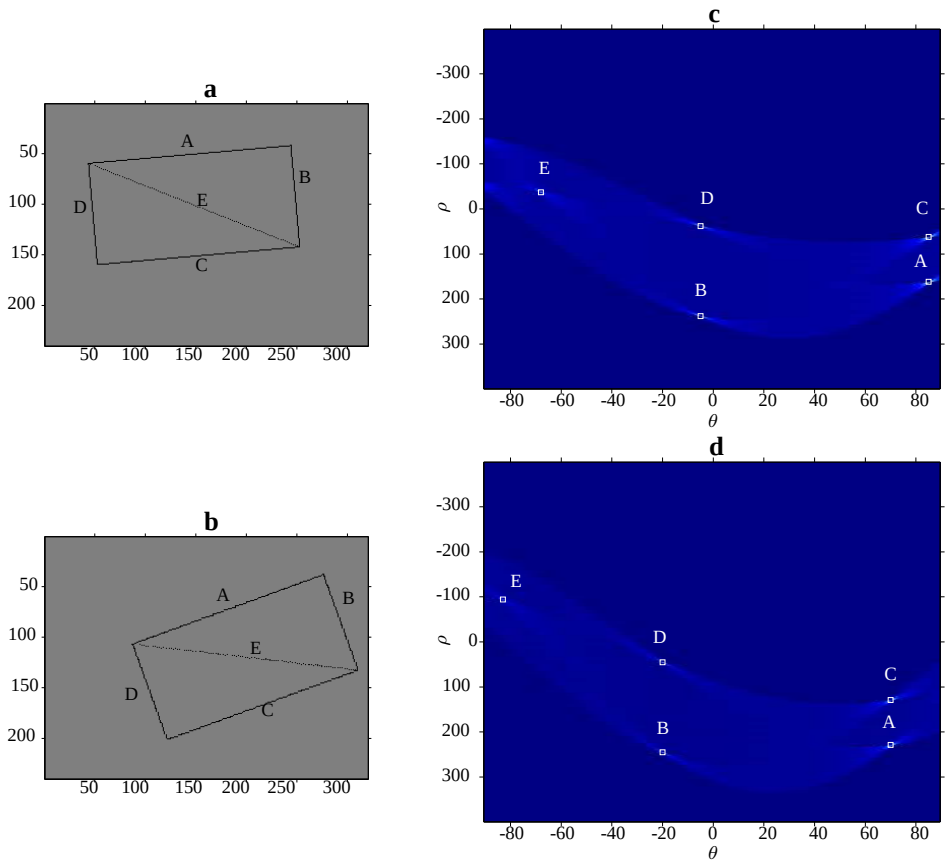

Figure 3: Effect of the transformation of maps on the Hough images according to **Property 1 a)** An example occupancy grid map. **b)** Hough image of the map over 180 degrees. Labeled points correspond to the the line segments. **c)** The map is rotated 15° and translated 50 cells along both axes. **d)** Hough image of the transformed map over 180 degrees

2.2.1 Applications to Robotics Localization and Mapping

In robotic applications, the Hough transform has also been used in scan matching [6] and localization [23], [15], [24]. Censi et al. [6] propose a solution for scan matching, called Hough scan matching (HSM). In HSM, a spectrum function is defined for the Hough image of each laser scan. Then the spectrum functions and their correlations are used to find the transformation between laser scans. The same approach has been adopted by S. Carpin [3] for map merging. However, this approach needs significant overlap between the Hough images to work well. For scan matching, HSM is successful because consecutive scans usually have a lot of overlap; this technique does not produce good results for map merging when little overlap is available. This approach will be further elaborated, and the results will be compared with Carpin's, in

the experimental section.

For mobile robot localization, X. Yun et al. [40] propose a method using the Hough transform. In this solution, the Hough transform is used to identify wall-like features from noisy sound navigation and ranging (SONAR) data. Then these features are used to localize the robot. Because of the noisy SONAR data, this approach is vulnerable to the resolution of the map and the Hough images; however, a neural network approach is used to compensate for this problem.

Iocchi et al. [23], [24] extend the Hough localization into a probabilistic framework. In this framework, an extended Kalman filter is used to estimate the pose of the robot. At each time, the pose is predicted using dead-reckoning and then updated by the Hough map matching method. To perform map matching, the Hough image of the known map of the environment is matched with the Hough image of the local map built locally by the robot's sensor. This method is robust and seems to be computationally efficient, because the Hough image of the environment can be obtained in advance and the Hough image of the local map can be generated quickly due to the small size of the local maps. However, this solution assumes a small error exists for the pose of the robot. This assumption restricts the application of this method to the multiple-robot SLAM problem. In other words, dead-reckoning provides a good initial estimate for the pose of the robot. The solution presented in this work differs from [23] and [24] in which no initial estimate is available.

Grisetti et al. [15] extend the work by Iocchi et al. [23], [24] for the global localization problem, where the robot is not aware of its initial position. The solution presented in this work differs from Grisetti's work for localization in several different respects. First, in global localization, correlation of the map and the sensor measurements is used to calculate the translation and orientation. This solution works only when there is a significant overlap between the map and the measurements which is a valid assumption for localization; however, this assumption does not hold for map merging. In fact in map merging, overlaps have to be identified prior to any processing. Second, Grisetti et al. calculate multiple hypotheses for the rotation, and then for each hypothesis they calculate a separate translation. This increases the computational demand. In our proposed method, multiple hypotheses are handled in such a way that only one candidate for the rotation exists. Third, for each angle of the Hough image, Grisetti et al. use the aggregation of all points of the Hough image within that angle to produce a signal which is a function of the angle only. Then the signal is used for correlation. It is shown that this method, similar to averaging over angles, provides poorer results in map merging. In other words, techniques proposed by Iocchi et al. [23], [24] and Grisetti et al. [15] are more suitable for the localization problem. Our approach improves upon their works in the context of multi-robot mapping in unknown environments without knowing relative positions of the robots.

3 Problem Statement

In this section, formal definitions and requirements for multiple robot SLAM by map merging are presented. First single robot SLAM and occupancy grid map are introduced mathematically, then the map merging problem is discussed.

3.1 Single Robot SLAM

Single robot SLAM is defined as follows. At times $1, 2, .., t$, let $x_{1:t}$ and $z_{1:t}$ denote corresponding sequences of poses and measurements, respectively. The sequence of action signals is given by $u_{1:t-1}$. The goal for SLAM is to calculate the posterior over the map and the trajectory given the action and measurement signals: $p(m, x_{1:t}|z_{1:t}, u_{1:t-1})$.

In Rao-Blackwellized Particle Filtering (RBPF) [18], [16], which is used in this paper to perform single robot SLAM, the posterior is factorized as follows:

$$p(m, x_{1:t}|z_{1:t}, u_{1:t-1}) = p(m|z_{1:t}, x_{1:t})p(x_{1:t}|z_{1:t}, u_{1:t-1}). \tag{7}$$

$p(x_{1:t}|z_{1:t}, u_{1:t-1})$ is calculated by a particle filter using the motion and sensor models. Once $x_{1:t}$ is calculated, $p(m|z_{1:t}, x_{1:t})$ is calculated analytically.

3.2 Occupancy Grid Mapping

Occupancy grid map is a standard method for mapping using ranger measurements. A map m can be represented as a set of N grid cells, attached to each is a binary random variable, showing its occupancy:

$$m = \{m_i\}, i = 1, ..., N. \tag{8}$$

The binary random variable specifies whether a cell is occupied ('1') or free ('0'), then $p(m_i = 1)$ or $p(m_i)$ represents the probability that the cell is occupied.

In an occupancy grid map, it is desired to calculate the posterior over the map given the trajectory of the robot and available measurements as follows [38]

$$p(m|z_{1:t}, x_{1:t}) = \prod_i p(m_i|z_{1:t}, x_{1:t}) \tag{9}$$

The factorization defined in (9) makes updating the map tractable.

Assuming initially a cell i has *unknown* occupancy, $p(m_i) = 0.5$, at time t, if the cell is in the perception field, its value is calculated using following recursive relation:

$$l_{t,i} = l_{t-1,i} + \log \frac{p(m_i|z_t, x_t)}{1 - p(m_i|z_t, x_t)}. \tag{10}$$

where $l_{t,i}$ is referred to as the log odds and is defined as

$$l_{t,i} = \log \frac{p(m_i|z_{1:t}, x_{1:t})}{1 - p(m_i|z_{1:t}, x_{1:t})}, \tag{11}$$

The *log odds* representation helps to avoid numerical instabilities for near zero or one probabilities. The probability of $p(m_i|z_t, x_t)$ is sensor specific and for a laser ranger can be modeled using the approach presented in [38]-Ch.9.

3.3 Map Merging

Let a transformation be composed of a 2×2 rotation matrix R_ψ and a 2×1 translation vector T as follows

$$R_\psi = \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix}, T = \begin{bmatrix} \delta_x \\ \delta_y \end{bmatrix}, \tag{12}$$

Let m_1 and m_2 be two occupancy grid maps. Assuming that m_2 is merged into m_1, the map merging problem is defined as: find a rotation matrix, R_ψ and a translation vector, T which transforms m_2 such that the overlaps of m_1 and the transformed m_2, m_2', fall on each other. To do this it is required to maximize a verification index defined on the merged map:

$$(R_\psi, T) = \underset{\psi,\delta_x,\delta_y}{\operatorname{argmax}} V(m_1, m_2'), \tag{13}$$

where m_2' means that the m_2 is transformed according to R_ψ and T. $V(\cdot)$ is a criterion that evaluates the merging process and will be explained in subsequent sections.

This optimization is not easy to solve analytically and some methods like [4] uses an exhaustive search to find a transformation matrix and then checks (13) to see if it is satisfied. This paper proposes a novel approach that uses the Hough space to find a good estimate of the optimized R_ψ and T very efficiently.

Fig. 4 depicts an example of map merging with three maps. The overlap between map_1 and map_2 is shown with a dashed ellipse and the overlap between map_1 and map_3 is shown by a solid line ellipse. To find the required transformation, the first step is to identify overlaps, then use the overlaps to calculate the alignment and finally verify the results using (13).

The map merging problem can be interpreted as a special case of image registration in computer vision where images are maps with special geometry. However, the problem is that in multiple robot map merging, the percentage of overlap between maps is usually low.

After finding the relative transformation between two maps, m_1 and m_2, the probabilities are combined to produce the final map. The data that is received from the transformed map, 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 (11):

$$l_{t,i}^{fused} = l_{t,i}^{1} + l_{t,i}^{\prime 2}, \tag{14}$$

for all $i = 1..N$. Where $l_{t,i}^{1}$ and $l_{t,i}^{\prime 2}$ are defined in (11). The superindices are used to identify the maps, l^1 is for m_1 and l'^2 is for m_2'. l^{fused} represents the fused map, m_{fused}.

4 Map Merging with Hough Transforms

Fig. 5 shows the algorithm for two robots with unknown relative positions and each with its own local map. Note that this algorithm is scalable and can be used for more than two robots. Maps in this algorithm are assumed to be in the form of occupancy grid maps.

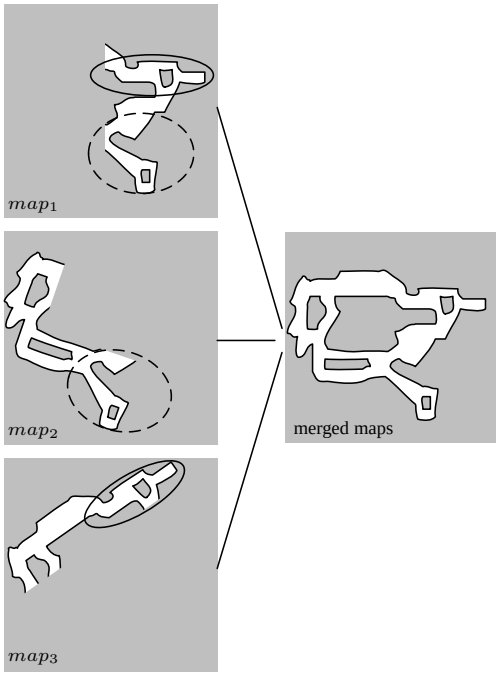

Figure 4: An example of map merging. Three maps are merged to form a complete map of the environment. The overlap between map_1 and map_2 is shown with a dashed ellipse. The overlap between map_1 and map_3 is shown with a solid line ellipse.

According to Fig. 5, the flow diagram of the system, inputs are map_1 and map_2. First, in the **Hough Rotation** block, a function is applied to the Hough images of the inputs to find the relative orientation. This block produces one approximate solution. In the **Multiple Hypothesis** block, multiple accurate estimates are produced. Results of these two block are combined to generate one accurate solution for rotation. Then this result is tuned in the **Tuning Rotation** block. After finding the orientation, in the **Hough Translation** block, an approximate translation is found using oriented Hough images. The approximate translation is adjusted in the **Entropy Tuning** block using image entropy. Finally, in the **Similarity Index** block, results are verified to ensure the accuracy of the map fusion.

For two given maps, $m_1 = map_1$ and $m_2 = map_2$, assume that m_1 is composed of two parts, a set of cells, m, that exists in m_1 and m_2 and a set of cells Δ_1 that exist only in m_1 (Fig .6). The same assumption applies to m_2 where m exists in both maps and Δ_2 exists only in m_2:

$$m_1 \triangleq \{m\} \cup \{\Delta_1\} \tag{15}$$

$$m_2 \triangleq \{m\} \cup \{\Delta_2\}. \tag{16}$$

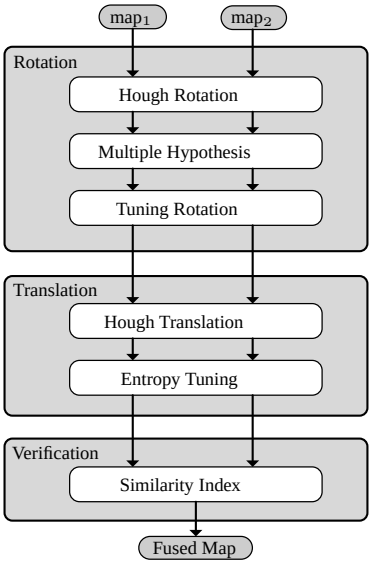

Figure 5: The proposed map fusion algorithm. The two input maps are fused by finding their relative transformation matrix. No prior information is available regarding the relative position of two robots.

According to **Property 2**, the Hough image of m_1 is:

$$\mathcal{H}_{m_1}(\theta,\rho) = \mathcal{H}_m(\theta,\rho) + \mathcal{H}_{\Delta_1}(\theta,\rho). \tag{17}$$

Assume that R_ψ and $T = \begin{bmatrix}\delta_x & \delta_y\end{bmatrix}^T$ are transformation elements that can fuse these two maps. According to **Property 1** and **Property 2**, the Hough image of m_2', the transformed m_2, becomes:

$$\begin{aligned}\mathcal{H}_{m_2'}(\theta,\rho) = \mathcal{H}_m(\theta+\psi, \rho + \begin{bmatrix}\cos(\theta+\psi) & \sin(\theta+\psi)\end{bmatrix} T) + \\ \mathcal{H}_{\Delta_2}(\theta+\psi, \rho + \begin{bmatrix}\cos(\theta+\psi) & \sin(\theta+\psi)\end{bmatrix} T).\end{aligned} \tag{18}$$

If there is a method to identify the overlaps between maps, $\{m\}$, then the transformation is calculated by the fact that the Hough transform of $\{m\}$ from map_1 should be the same as the the Hough transform of $\{m\}$ from the transformed map_2, m_2':

$$\mathcal{H}_m(\theta,\rho) = \mathcal{H}_m(\theta+\psi, \rho + \begin{bmatrix}\cos(\theta+\psi) & \sin(\theta+\psi)\end{bmatrix} T). \tag{19}$$

The main tasks which are addressed in the following sections are:

- how overlaps can be identified and
- how the transformation is calculated from (19).

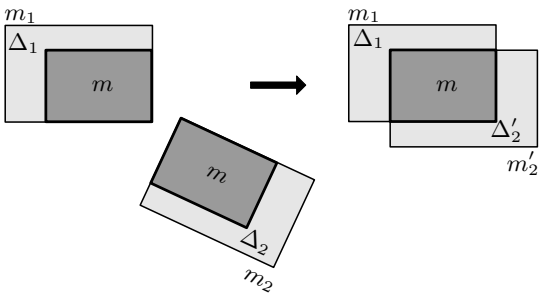

Figure 6: For two given maps, m_1 and m_2, m is the overlap of the maps. Δ_1 and Δ_2 are non overlapping parts of each map.

4.1 Extracting Relative Orientation of Maps

To calculate the transformation, it is required to find overlaps of maps. To find overlaps between maps, they should be compared with each other. One way to compare maps is to extract features from maps and then match features against each other. In general extracting map features is not easy, however, in structured environments, where maps are composed of line segments with different sizes, features can be extracted using the Hough transform. As explained, if the Hough transform is applied to an occupancy grid map, line segments appear as local peaks and the relation of local peaks in the Hough space, corresponds to the geometry of the real world. In fact, peak points are treated as features extracted from maps.

Let us assume that $\mathcal{H}_1$ and $\mathcal{H}_2$ are the Hough images of map_1 and map_2. According to **Property 3**, the internal relations of the local peaks of $\mathcal{H}_1$ and $\mathcal{H}_2$ correspond to special geometric shapes (composed of line segments) in the maps. By comparing these peaks overlaps are identified in both maps (see Fig. 3 as an example where there is a full overlap). These overlaps are used to find the rotation. The only limitation with this approach is that if there are similar patterns in both maps, the extracted overlaps may not be completely accurate.

In order to find a correct solution, the key is to correctly associate the relative peaks from the two Hough transforms. This is achieved using the values of the peaks, which can be mapped to the lengths of the line segments in the original maps.

Algorithm 1 shows the peak association and rotation calculation. Inputs to the algorithm are the Hough images of map_1 and map_2, denoted by $\mathcal{H}_{m_k}$ where $k = 1, 2$. k is the number of robots, but here the algorithm is explained for two robots for simplicity.

v_{gate} is a parameter used to reject outlier peaks. If the difference of values of two peaks is more than v_{gate} then they are not considered as associated points.

The local peaks of Hough images of map_1 and map_2 are denoted by P_k where $k = 1, 2$ (line 1-2). P_k is defined as

$$P_k = \{p_k^i\}_{i=1}^{n_k}, p_k^i = (\theta_k^i, \rho_k^i, v_k^i). \tag{20}$$

P_k is a set of n_k points, each with triplets $(\theta_k^i, \rho_k^i, v_k^i)$. The first two arguments are the

Algorithm 1 Relative Rotation.

Input: $\mathcal{H}_{m_1}, \mathcal{H}_{m_2}, v_{gate}$
Output: ψ
► Hough Rotation Block
1: $P_1 \leftarrow$ local peaks of $\mathcal{H}_{m_1}$
2: $P_2 \leftarrow$ local peaks of $\mathcal{H}_{m_2}$
3: $C \leftarrow \emptyset$
4: **for** $i^\star = 1 \rightarrow |P_1|$ **do**
5: $\quad j^\star \leftarrow \underset{j}{\operatorname{argmin}} |v_1^{i^\star} - v_2^j|$
6: $\quad \delta_v \leftarrow |v_1^{i^\star} - v_2^{j^\star}|$
7: $\quad$ **if** $\delta_v < v_{gate}$ **then**
8: $\quad\quad C \leftarrow C + (i^\star, j^\star)$
9: $\quad$ **end if**
10: **end for**
11: $\hat{\psi} \leftarrow \frac{1}{|C|} \sum_{n=1}^{|C|} (\tan\theta_1^{i_n} - \tan\theta_2^{j_n})$
► Multiple Hypothesis Block
12: $h_1(\theta) \leftarrow f(\mathcal{H}_{m_1}(\theta, \rho))$
13: $h_2(\theta_o + \psi) \leftarrow f(\mathcal{H}_{m_2}(\theta, \rho))$
14: $\Psi_o = \{\psi_o^i\}_{i=1}^l \leftarrow h_1(\theta) \circledast h_2(\theta + \psi)$
15: $\psi_o \leftarrow \operatorname{argmin}_i(|\psi_o^i - \hat{\psi}|)$
► Rotation Tuning Block
16: $\alpha_1 \leftarrow \psi_o - \Lambda$
17: $\alpha_2 \leftarrow \psi_o + \Lambda$
18: $\delta_\psi \leftarrow \operatorname{argmax}_\theta \max(\mathcal{H}_{m_1}^{\theta=\alpha_1:\alpha_2}) - \operatorname{argmax}_\theta \max(\mathcal{H}_{m_2'}^{\theta=\alpha_1:\alpha_2})$
19: $\psi \leftarrow \psi_o + \delta_\psi$

Hough coordinates and the last is the value of the peak at that location in the Hough image. Set C holds pointers to associated peak points from P_1 and P_2, defined as:

$$C = \{(i_n, j_n)\}, i = 1, ..., n_1, j = 1, ..., n_2, n = 1, ..., N \tag{21}$$

where (i_n, j_n) means $p_1^{i_n}$ is associated with $p_2^{j_n}$. N is the total number of associated points, denoted by subscript n.

Initially C is empty (line 3). For each peak from P_1, its candidate associated point from P_2 is found by a simple search algorithm (line 4-5). If the difference of peak values is less than a threshold (line 6-7), then points are considered to be associated and added to C (line 8). After finding all correspondences, the slopes of the lines are calculated based on **Property 3**. The average difference of slopes provides an estimate for the relative rotation (line 11). Based on experiments this estimate is not very accurate. However, the estimated rotation is used to find a better solution for the rotation.

Let us assume there exists some function $f(\cdot)$, taking an array as its input and outputting a scalar, that is applied to a Hough image at a fixed angle, θ_o. Therefore the

input of $f(\cdot)$ is dependant only on the second input of the Hough image.

$$h(\theta_o) = f(\mathcal{H}_m(\theta, \rho)\,\big|_{\theta=\theta_o}) \tag{22}$$

where $f(\mathcal{H}_m(\theta, \rho)\,\big|_{\theta=\theta_o})$ means that the input of $f(\cdot)$ is one column of the image $\mathcal{H}_m$ which corresponds to the angle θ_o. By applying $f(\cdot)$ to the Hough images of m_1 and m_2, we will have:

$$\begin{aligned} h_1(\theta_o) &= f(\mathcal{H}_{m_1}(\theta, \rho)|_{\theta=\theta_o}) \\ h_2(\theta_o + \psi) &= f(\mathcal{H}_{m_2}(\theta, \rho)|_{\theta=\theta_o+\psi}). \end{aligned} \tag{23}$$

where $h_1(\cdot)$ and $h_2(\cdot)$ are showing the result of applying $f(\cdot)$ on the Hough image at each angle.

$f(\cdot)$ should be designed in such a way that $h_1(\cdot)$ and $h_2(\cdot)$ have highly similar patterns. This is feasible because according to **Property 1** the second argument of the Hough image after transformation is shifted by a specified amount. Therefore any Order Invariant Function (OIF)[1] like *max*, *averaging over non zero elements* or *entropy* would be a good candidate for $f(\cdot)$.

By extending (23) to all angles, $\theta_o = [-\pi/2, \pi/2)$, two function from images are extracted; $h_1(\theta)$ and $h_2(\theta + \psi)$ (line 12-13). To extract ψ, which is the relative rotation between m_1 and m_2, a circular cross correlation is used. The cross correlation must be circular because the Hough image is periodic and boundary effects in periodic signal should be avoided:

$$\Psi_o = h_1(\theta) \circledast h_2(\theta + \psi). \tag{24}$$

where $\circledast$ calculates the circular cross correlation of $h_1(\cdot)$ and $h_2(\cdot)$ and outputs local peaks as candidate rotations. The subscript o denotes that the outcomes of (24) are initial estimates and will be refined later. The location of the peak in the correlation shows an estimate for the relative rotation between the maps.

Intuitively, if *max* is used as the function $f(\cdot)$, it will be equivalent to extracting peak points (which represent long walls) at each angle from both maps and finding the correlation between them.

In circular cross correlation, due to incomplete overlap between maps, process noise and periodicity, other local maxima should be taken into account as potential solutions. Therefore multiple peaks are considered and only one is accepted by a rotation verification process.

Let us assume that there are l rotation candidates defined as (line 14):

$$\Psi_o = \{\psi_o^i\}_{i=1}^l \tag{25}$$

At line 11, only one estimate was calculated, called $\hat{\psi}$. $\hat{\psi}$ is used in a rotation verification process. This estimate is approximate but since members of Ψ_o are highly sparse, it can help to select the best rotation from the set Ψ_o by the following relation (line 15):

$$\psi_o = \underset{i}{\operatorname{argmin}}(|\psi_o^i - \hat{\psi}|), \tag{26}$$

[1] An OIF is a function that reordering its input arguments does not effect the output. For example $f(a_1, a_2, a_3, a_4) = (a_1 + a_2)^2 + (a_3 + a_4)^2$ is not an OIF but $f(a_1, a_2, a_3, a_4) = (a_1 + a_2 + a_3 + a_4)^2$ is.

which means the best rotation candidate is the one closest to the only approximate rotation.

The rotation can be tuned by performing a comparison between the peak points of the Hough images of m_1 and m_2', the rotated m_2 by ψ_o. This means that given an initial start angle, ψ_o, the peak comparison process can be performed around the rotation angle, ψ_o and within the interval (α_1, α_2) where $\alpha_1 = \psi_o - \Lambda$ and $\alpha_2 = \psi_o + \Lambda$ (line 16-19).

$$\begin{aligned} \delta_\psi &= \underset{\theta}{\operatorname{argmax}} \max(\mathcal{H}_{m_1}^{\theta=\alpha_1:\alpha_2}) - \underset{\theta}{\operatorname{argmax}}\, max(\mathcal{H}_{m_2'}^{\theta=\alpha_1:\alpha_2}), \\ \psi &= \psi_o + \delta_\psi \end{aligned} \tag{27}$$

where $\mathcal{H}_{map}^{\theta=\alpha_1:\alpha_2}$ is the Hough image of the input map over the specified domain of angles (θ). This means the Hough image is prepared around ψ_o with the interval of (α_1, α_2), where Λ is an arbitrary small interval in which the resolution of the Hough image is higher than 1^o resolution used in the previous step. $\max$ function operates the same way as $f(\cdot)$ does in (22). δ_ψ is the amount of the tuning angle added to ψ_o and ψ is the tuned rotation.

4.2 Extracting Relative Translation of Maps

Now assume that $\mathcal{H}_1$ and $\mathcal{H}_2'$ are the Hough images of map_1, m_1, and rotated map_2, m_2'. As for Algorithm 1, local peaks of $\mathcal{H}_1$ and $\mathcal{H}_2'$ represent special geometric shapes. By comparing these peaks it is possible to find the overlaps between maps. To account for the geometric relations between the local peaks, the proposed algorithm can be done iteratively like point cloud matching. However, experiments show that if outliers are rejected by properly tuned parameters, one iteration suffices. Another advantage of matching peaks is that the lines and geometric shapes in the maps are represented as intensity points in the Hough images and generally dealing with points is much easier than with shapes.

To establish the correct peak association two types of information from the Hough images are used: the values of the peaks and the distance between peaks with the same value and orientation.

Algorithm 2 shows the peak association and translation calculation. The inputs to the algorithm are the local peaks of the Hough image of map_1, P_1, and the local peaks of the Hough image of the rotated map_2, P_2'. These two sets have the same structure as that introduced in (20), except that the prime superscript on P_2' shows that the peaks are extracted from the Hough image of map_2 after applying the rotation. d_{gate} and v_{gate} are used to reject outliers.

Similar to Algorithm 1, as introduced in (21), set C is defined to hold associated points from P_1 and P_2'.

Initially C is empty (line 1). α spans from -90° to 90° (line 2). At each α, there are limited (or no) local peaks. For each local peak residing at α, associated points from P_2' are found considering the minimum difference in peak values, $|v_1^i - v_2^j|$ and distance $|\rho_1^i - \rho_2^j|$ over the range of $\mathcal{U} = \{(i,j) \subset \mathbb{N}^2 | i = 1, ..., n_1, j = 1, ..., n_2\}$ (line 3). While peaks of map_1 are at α, the algorithm looks for its associated peaks at

δ_α, where δ_α is an interval defined as $[\alpha-\delta, \alpha+\delta]$. The reason to include this interval is to take into account possible inaccuracy of the rotation. For example, for a given peak from P_1 at angle 50°, its associated peak from P_2' might reside at 51° because of inaccurate rotation. δ in this research is considered to be 1°.

The gate values, d_{gate} and v_{gate}, are used to reject outliers. This means that if the difference of peak values of two cells is less than v_{gate} *and* their distance is less than d_{gate}, then the cells are associated. If there exists corresponding peaks, they are added to C (line 4-8).

According to **Property 1**, for each pair of associated points the following relation is established ($\psi = 0$ for $\mathcal{H}_1$ and $\mathcal{H}_2'$):

$$\rho_i' = \rho_i + \begin{bmatrix}\cos\theta_i & \sin\theta_i\end{bmatrix} T_o \tag{28}$$

Considering this relation for all N corresponding points, the following relation can be formulated:

$$\begin{bmatrix}\rho_1' - \rho_1 \\ \vdots \\ \rho_N' - \rho_N\end{bmatrix} = \begin{bmatrix}\cos\theta_1 & \sin\theta_1 \\ \vdots & \vdots \\ \cos\theta_N & \sin\theta_N\end{bmatrix} T_o. \tag{29}$$

This equation can be written in the form of $B = AT_o$ and the least squares error solution for this equation is (line 10-11)

$$T_o = (A^T A)^{-1} A^T B. \tag{30}$$

Algorithm 2 Relative Translation by matching peaks of the Hough images

Input: P_1, P_2', d_{gate}, v_{gate}, Δ_x, Δ_y
Output: T

► Hough Translation Block

1: $C \leftarrow \emptyset$
2: **for** $\alpha = -90^\circ \rightarrow \alpha = 90^\circ$ **do**
3: $(i^\star, j^\star) = \underset{\mathcal{U}}{\operatorname{argmin}}(|v_1^i - v_2^j|, |\rho_1^i - \rho_2^j|) \Big| \begin{matrix}\theta_1^i = \alpha \\ \theta_2^j = \delta_\alpha\end{matrix}$
4: $\delta_v \leftarrow |v_1^{i^\star} - v_2^{j^\star}|$
5: $\delta_\rho \leftarrow |\rho_1^{i^\star} - \rho_2^{j^\star}|$
6: **if** $\delta_v < v_{gate}$ and $\delta_\rho < d_{gate}$ **then**
7: $C \leftarrow C + (i^\star, j^\star)$
8: **end if**
9: **end for**
10: Calculate A and B from C based on (29).
11: $T_o \leftarrow (A^T A)^{-1} A^T B$

► Entropy Tuning Block

12: $\mathbf{tr} \leftarrow \operatorname{argmin}_{\mathrm{S}}\{\mathbf{H}(J(m_1, \mathcal{T}_{x,y}(m_2')))\}$

The calculated translation can be tuned to provide more accurate results. This is done using image entropy. Image entropy is defined as:

$$\mathbf{H}(map) = -\sum p \log_2(p), \tag{31}$$

where p is the normalized histogram of map. Entropy is a statistical measure of the randomness associated with an image and is usually applied to characterization of image texture. A translation is desired which minimizes the entropy of aligned maps. To do this optimization, an exhaustive search in the neighborhood of the approximate translation is done. It is challenging to use guided search algorithms like Genetic Algorithms (GA) or Particle Swarm Optimization (PSO) because in many cases the best solution may occur very close to very poor solutions. The size of the search space is determined by experience based on the performance of the approximate translation method. In addition, if the total allowable time for searching can be determined, the size of the search space can be specified such that the search is completed within the allotted time. Usually a search space with a radius of ten cells around the current approximate solution is acceptable considering the required time constraints and accuracy of the previous result.

The following relation gives the best translation vector (line 12):

$$T = \begin{bmatrix} \delta_x \\ \delta_y \end{bmatrix} = \underset{\mathcal{S}}{\operatorname{argmin}} \{\mathbf{H}(J(m_1, \mathcal{T}_{x,y}(m_2')))\}$$

$$\begin{aligned} \mathcal{S} = \{(x,y) \subset \mathbb{N}^2 | & \delta_{x_o} - \Delta_x < x < \delta_{x_o} + \Delta_x, \\ & \delta_{y_o} - \Delta_y < y < \delta_{y_o} + \Delta_y\} \end{aligned} \tag{32}$$

where m_2' is the rotated m_2 by ψ and $\mathcal{T}_{x,y}(map)$ means that the input map is translated according to the values of x and y. $J(map_1, map_2)$ means both input maps, map_1 and map_2 are fused using (14) and are in the same coordinate frame according to the coordinates of the first input, map_1. $\mathcal{S}$ is the search space, defined as a rectangle centered at $(\delta_{x_o}, \delta_{y_o})$ with dimensions $2\Delta_x \times 2\Delta_y$. T is the tuned translation vector.

4.3 Verification of Results

After finding the transformation matrix, a verification is performed. A performance index, called the similarity index [2] is used and measures the similarity of two maps over some desired region. When there is more overlap between two maps, there will be an extremum in the index. This index is composed of two components:

$$\begin{aligned} agr(map_1, map_2) &= |\{p = (x,y) | map_1(p) = map_2(p)\}|, \\ dis(map_1, map_2) &= |\{p = (x,y) | map_1(p) \neq map_2(p)\}|, \end{aligned} \tag{33}$$

where the operator $|\cdot|$ over a given set returns the cardinality of the set. The function $agr(\cdot)$, the agreement index, is the number of known cells with equal status in both maps (either both occupied or both free). The function $dis(\cdot)$, the disagreement index, is the number of cells which are known in at least one map and have unequal status.

The similarity verification uses the ratio of $agr(\cdot)$ and $dis(\cdot)$ as given by:

$$V(m_1, m_2') = \frac{agr(m_1, m_2') \times 100\%}{agr(m_1, m_2') + dis(m_1, m_2')}, \tag{34}$$

where m_2' is the final transformed m_2 and $V(m_1, m_2')$ is the verification index. If $V(m_1, m_2')$ is larger than a threshold, then the proposed transformation can be accepted. The maximum verification percentage occurs at the exact transformation so it is very likely that the transformation matrix that has been found is correct.

5 Experimental Results

This section presents hardware and software setups and some experimental results. For real world experiments, three differential-wheeled robots are used where one is shown in Fig. 7. The robots are built by CoroWare, Inc. and each is equipped with High Speed Phidget Encoders and a Hokuyo UBG-05LN laser ranger. Data from these sensors are fused using a particle filtering scheme [21]. In the rest of the section, first an experiment on a dataset is explained in detail. Then an experiment is performed in Gazebo with four robots on a simulated environment. Then an experiment with two robots in a real-world is presented and at the end another experiment with three robots is demonstrated.

Figure 7: experimental robots.

5.1 RADISH dataset, Fort AP Hill

The first experiment is performed on RADISH Fort AP Hill dataset [31]. This is an open source and well-known dataset which is used as a benchmark dataset for view-based multiple robot SLAM. The raw laser ranger and encoder data are recorded in standard Player format [39] and fuzed by particle filtering [21].

Figs. 8-a and Fig. 8-b show the two maps before fusion. The maps have about 5° rotation. The proposed method is used to find the relative rotation between the maps.

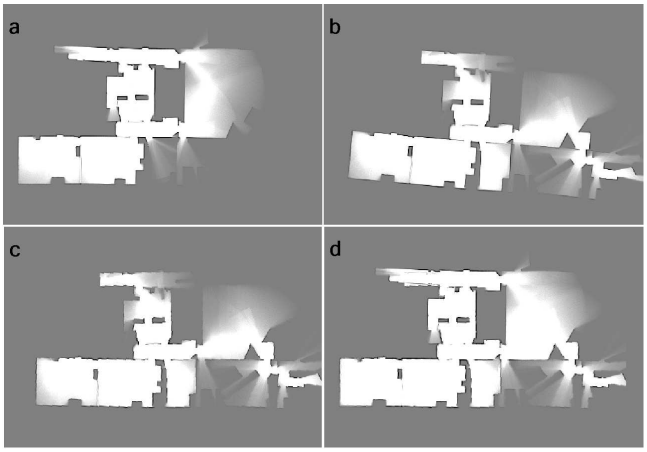

Figure 8: **a)** map_1, **b)** map_2, **c)** map_2 after tuned rotation, **d)** two maps after transformation.

By applying the *Hough Rotation* block of Algorithm 1, an approximate rotation is extracted as $\hat{\psi} = -15°$.

Then the Algorithm 1 continues with *Multiple Hypothesis*. Fig. 9-a shows $h_1(\theta)$ and $h_2(\theta+\psi)$ in blue and red colors. $h_1(\theta)$ is the output of $f(\cdot)$ on map_1 and $h_2(\theta+\psi)$ is the output of $f(\cdot)$ on map_2, where $f(\cdot)$ is $max(\cdot)$ function. To find ψ_o the result of the circular cross correlation has been depicted in Fig. 9-b. The result has four local peaks, $\Psi_o = \{-5°, 85°, 175°, -95°\}$. These peaks are considered as potential candidates for ψ_o. The approximate rotation which is $\hat{\psi} = -15°$ is close to $-5°$. Based on (26) the rest of the experiment continues with $\psi_o = -5°$.

To compare the effect of other alternatives for $f(\cdot)$, *averaging over non zero elements* of the given input vectors has been examined. Fig. 9-c and Fig. 9-d show $h_1(\theta)$ and $h_2(\theta + \psi)$ and the result of correlation. This time the result has bigger offset from the required estimated value. *Entropy* of the input vector has been also evaluated which has an offset of the order of averaging function. $max(\cdot)$ produces better results and is used for other experiments in this paper.

After the tuning method ψ becomes $-5.5°$. The result of the final alignment is shown in Fig. 8-c.

Now the next step which is finding the translation is started. In this step first Hough images of map_1, m_1 and rotated map_2, m_2' are generated. These two images are shown in Fig. 10-a and 10-b respectively. Using the data association of algorithm 2 and solving (30), the initial estimate for the translation T_o is $[23, 1]^T$. (v_{gate} and d_{gate} for all experiments are 10 and 30, set by experiments.) Then using the exhaustive search and evaluating the image entropy the translation is tuned to be $[27, 2]^T$. Fig. 10-c shows the convergence of the image entropy. The initial estimate is shown by a circle and the tuned estimate is shown by a square. Finally Fig. 8-d shows the final alignment of both maps. The verification index, defined in (34) is 94%.

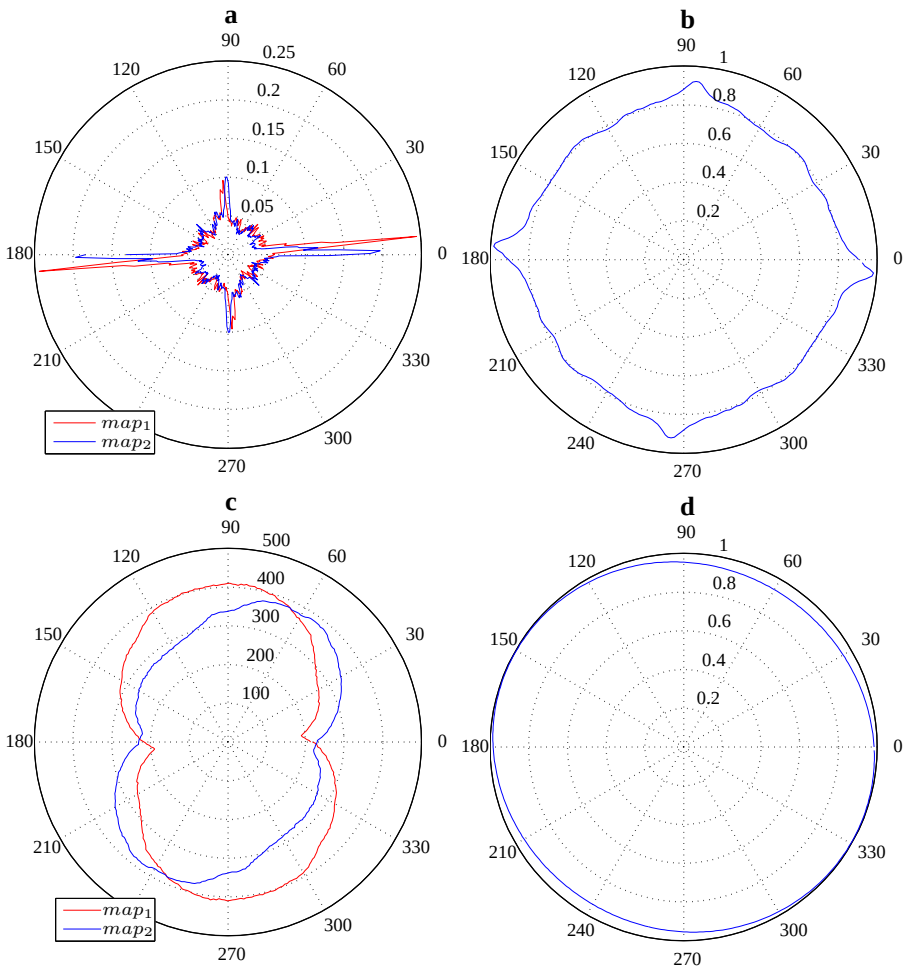

Figure 9: **a)** $h_1(\theta)$ and $h_2(\theta + \psi)$ when $f(\cdot)$ is $max(\cdot)$, **b)** Result of circular cross correlation. Four local peaks are at $-5°$, $85°$, $175°$ and $-95°$, **c)** $h_1(\theta)$ and $h_2(\theta + \psi)$ when $f(\cdot)$ is averaging, **d)** Result of circular cross correlation. Two local peaks are $-30°$ and $150°$.

5.2 Gazebo Simulation with Four Vehicles

This experiment is performed in ROS (Robot Operating System)-Gazebo [33] simulation environment. The simulated world is shown in Fig. 11-a, composed of 30 blocks with different sizes and orientations. Four Erratic robots start to explore and map the world using the gmapping algorithm [16]. As figures 11-b to 11-f show, each robot covers almost half of the world. The amount of overlap between the maps is shown in Table 1. (numbers are approximate.). Fig 11-e shows four final maps fused using the proposed algorithm.

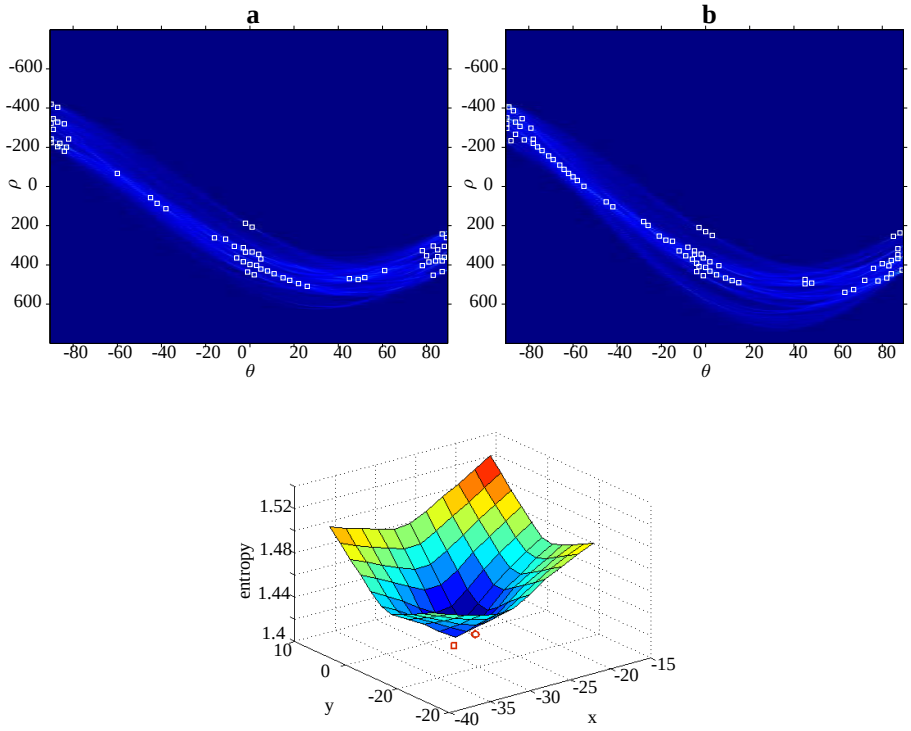

Figure 10: **a)** Hough image of map_1 with peaks points marked with white squares, **b)** Hough image of rotated map_2 with peaks points marked with white squares, **c)** Tuning translation by image entropy. The point shown by a circle is the initial estimate of the translation. The point shown by a square is the tuned translation.

Table 1: Percentage of approximate overlaps between maps.

Robot	R_1	R_2	R_3	R_4
R_1	-	61	5	48
R_2	61	-	52	15
R_3	5	52	-	42
R_4	48	15	42	-

5.3 Real Experiment with Two Vehicles

A real world experiment is performed with two CoroBots (Fig.7) in an indoor environment in the basement of Head Hall, the University of New Brunswick. The total area is $81.02m^2$ and the lengths of robot trajectories are $22.5m$ and $25.9m$. Fig.12-a and Fig.12-b show the maps of the robots. Although the area of this experiment is not very large, it is important for two reasons. First as the maps show, the required rotation is

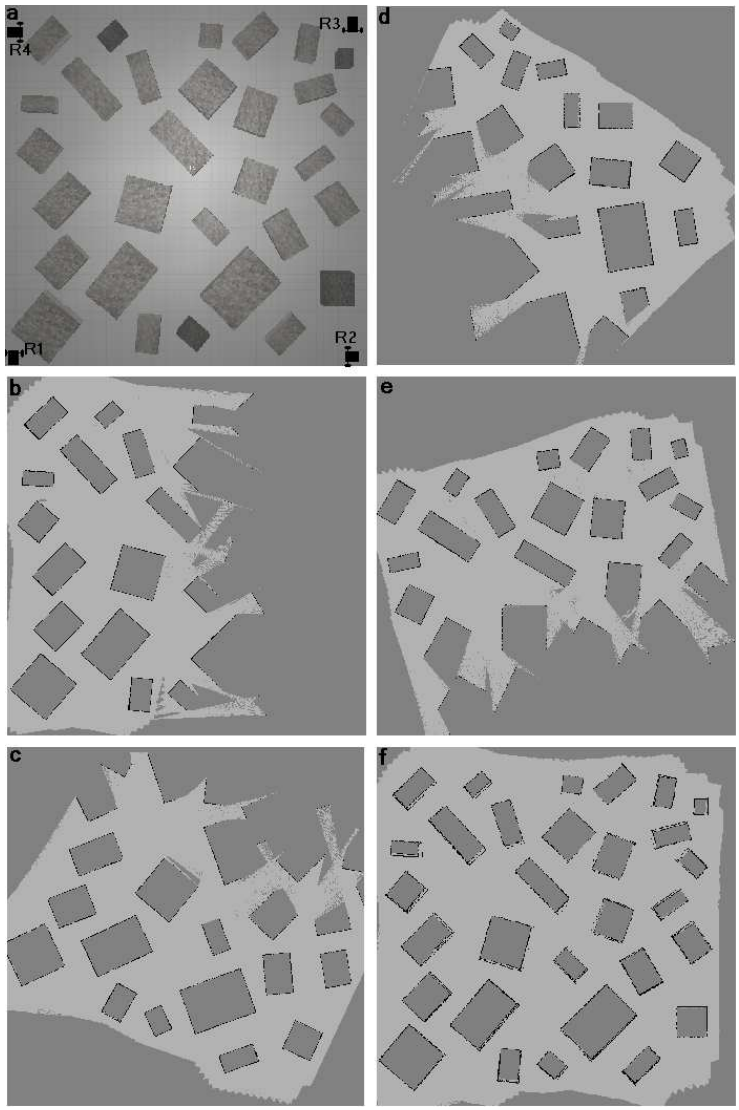

Figure 11: **a)** Simulated Gazebo world. Four Erratic robots map the world producing four partial maps, **b)** map by robot $R1$, **c)** map by robot $R2$. **d)** map of robot $R3$, **e)** map of robot $R4$, **e)** fused four maps to the coordinates of $R1$.

close to 90°. Second the overlap between two maps is approximately $25.4m^2$ which is 31% of the total area and this is important to show the effectiveness of the algorithm with small overlaps (The method proposed in [3] fails in this experiment due to small overlap). The overlaps in two maps are shown by ellipses. 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 outliers and finding the transformation based on the overlap. Fig.12-d shows final maps after fusion.

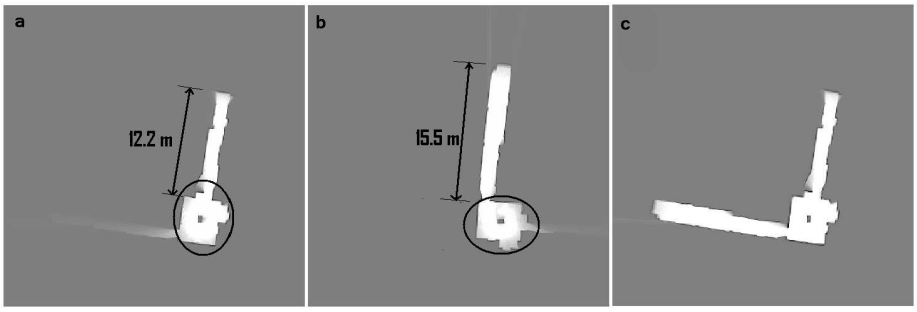

Figure 12: The experiment with two robots. **a)** map_1, **b)** map_2, overlaps of two maps are shown by ellipses, **c)** fused maps.

5.4 Real Experiment with Three Vehicles

This experiment is performed in a larger environment, with a coverage area of approximately $600m^2$ and three agents are involved. Trajectories of robots are approximately $60m$, $35m$ and $55m$. By merging maps, loop closure happens successfully. Fig. 13-a, b and c show three partial maps. Maps of Fig. 13-b and c are fused to Fig. 13-a. Overlaps of Fig. 13-a with other two maps are enclosed inside polygons. The fused global map of the environment is shown in Fig. 13-d.

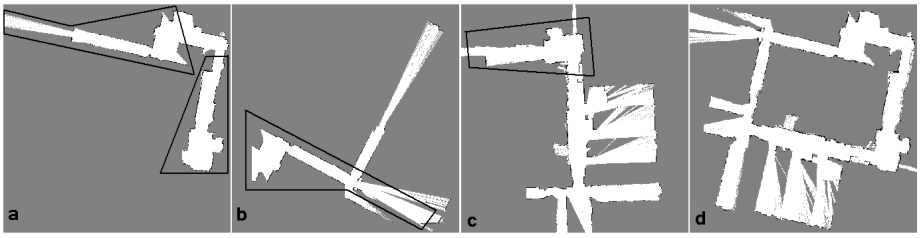

Figure 13: Three partial maps are fused together to generate a global map. (**a**) is the base map where (**b**) and (**c**) are fused to that. Overlaps are marked with polygons. (**d**) Final fused map which depicts loop closure.

6 Discussion

In this section two issues are presented. First tuning algorithm parameters are discussed and then results of the proposed method are compared with other related algorithms.

6.1 Tuning Parameters

Throughout the algorithm, a few parameters are introduced which need to be tuned for good performance. Table I lists these parameters. In this paper, these parameters have been tuned by trial and error however it is possible to use non-derivative optimization methods which is out of the scope of the paper. Repeated experiments show that changing these nominal values for about $\pm 15\%$ does not affect the quality of the output significantly.

Table 2: List of parameters with their nominal values.

Block	Parameter	Value
Hough Rotation	v_{gate}: peak value threshold	10
	Λ: rotation tuning interval	1°
Hough Translation	v_{gate}: peak value threshold	10
	d_{gate}: distance value threshold	30
	Δ_x: translation tuning interval	10
	Δ_y: translation tuning interval	10
Similarity Index	verification threshold	92 %

6.2 Comparisons

As mentioned, a major benefit of the Hough peak matching for map fusion is the low processing time requirement and its robustness. The proposed method is compared with Adaptive Random Walk (ARW) map merging [2], the Hough spectrum method [3], Map Segmentation [34], Fourier-Hough registration [7], and peak matching using RANSAC [12].

ARW is based on search and verification. In ARW, the similarity indices of two maps are calculated given a set of transformations. The set is composed of known rotations and translations. The transformation corresponding to the best similarity index is selected as the start point of an adaptive random walk search to merge the maps more accurately. Finding the initial transformation from the set takes the majority of the processing time.

The Hough spectrum approach is based on correlation of Hough images and there is no assessment on overlaps between maps. In this method, a spectrum function, originally proposed in [6] by A. Censi et al., is defined on the Hough image. The function is applied to both of Hough images of maps, then the correlation of the results is used to find potential rotations. There are multiple candidates which all are processed to find the translation. For each candidate a translation is calculated and at the end, only

Table 3: Normalized processing time and efficiency comparison of four experiments with three other methods, ① Radish dataset, ② Gazebo world with four robots ③ two CoroBots, ④ three CoroBots.

Experiment	①	②	③	④	①	②	③	④
Method	Processing (sec)				Verification (%)			
Hough peak matching	14	12	11	13	94	92	92	94
Map segmentation [34]	105	68	83	106	95	91	92	94
ARW map merging [2]	168	140	150	152	93	90	88	92
Hough spectrum [3]	16	16	fails	17	92	89	fails	90
Fourier-Hough transform	19	17	fails	16	88	89	fails	87
Peak matching using RANSAC	11	8	8	10	77	81	83	80

one rotation and translation are accepted. Carrying over all rotation candidates requires more processing time and computational power and makes the algorithm slow. For each rotation, the translation is found using correlation of projections of the Hough images along x and y directions. For example, if projections of the Hough image of map m_1 are shown by P_{x_1} and P_{y_1} and for the rotated map m_2 by P_{x_2} and P_{y_2}, then the peak of correlation of P_{x_1} and P_{x_2} results in the translation along x and the peak of correlation of P_{y_1} and P_{y_2} generates the translation along y. This approach is effective if projections have enough similar patterns, and this similarity happens only when there is a significant amount of overlap between maps. Therefore, finding the translation fails in maps with less overlaps.

Map segmentation is based on extracting thin lines from objects and walls of both maps and matching them. First, edges are found using Canny edge detection. Then, subsequent lines are extracted by smoothing the Canny edges. Edges are selected for cross correlation matching using a histogram filter.

In the Fourier-Hough registration method [7], [41] and [30], first a 1D Phase correlation is applied to the Hough images of two maps to find the relative orientation. Once the maps are aligned using the calculated orientation angle, a 2D convolution using the Fourier transform is applied to the aligned maps to calculate the translation.

To demonstrate the effectiveness of the proposed algorithm, a simple implementation of the RANdom SAmple Consensus (RANSAC) algorithm [20] was also performed. RANSAC estimates a mathematical model from a set of observed data iteratively. The data is assumed to be noisy and contains inliers and outliers. RANSAC can produce false models if the assumption does not hold. The peak points of the Hough images were used as input data to calculate the transformation between the the corresponding images. Obviously, there is no a priori information about the transformation.

To use RANSAC, first a correspondence set between the peak points of the Hough images must be established [20]. This correspondence is putative, and a subset of it will in fact be mismatches (outliers); however, RANSAC can deal with this situation. To establish the correspondence, for each peak (ρ, θ) in the first image, the match with highest intensity and neighborhood similarity in the second image is selected. This

process is implemented within a square search region centered on (ρ, θ). The size of the square should be large enough to account for the distance of the corresponding points generated by the transformation. Symmetrically, for each peak from the second map, this process is repeated. If one peak in one image is claimed by more than one peak in the other image, then the match with the highest similarity is chosen.

Once the correspondence set is determined, RANSAC is applied to calculate the transformation as well as the modified correspondence within the set (inliers). To calculate the transformation, four corresponding peaks are selected randomly from the correspondence set. A transformation is calculated for these peaks using equations (2) and (3). Then the symmetric fitting error (distance) of all other correspondences with respect to the transformation is calculated. To do this, the second Hough image is transformed to the coordinates of the first Hough image using the transformation (this process is also repeated by transforming the first image to the coordinates of the second image, to generate a symmetric transfer error). Ideally, if the transformation and the correspondences are perfect, the corresponding peak points will fall on each other. However, in reality, most of the peak points will not match; therefore, an error metric must be defined to evaluate the transformation. This error should in some way represent the distance between the corresponding peak points after the transformation. A proper definition of the distance of the peak points is very important to evaluate the transformation accurately. In the Hough space, each peak point has two coordinates, which are ρ (in pixels) and θ (in radians or degrees). Defining a distance with such heterogenous coordinates is a challenging problem. One possibility could be to treat the points in Hough coordinates in polar space and use the Euclidean distance in the polar coordinates. However, since these peak points in Hough space actually correspond to lines in Euclidean space, we define the distance between two peak points as the Euclidean distance between the end-points of the line segments.

Assume after applying the transformation, $P = (\rho, \theta)$ and $P' = (\rho', \theta')$ are two corresponding peaks from the two maps. Assume points p_a and p_b are the end-points of the line segment corresponding to peak P. Similarly, p_c and p_d are defined as the end-points of peak P'. Four distances between the end-points are defined as follows (see Fig. 14)

$$d_1 = ||p_a - p_c||_2, \tag{35}$$
$$d_2 = ||p_a - p_d||_2, \tag{36}$$
$$d_3 = ||p_b - p_c||_2, \tag{37}$$
$$d_4 = ||p_b - p_d||_2, \tag{38}$$

where $||p_i - p_j||_2$ represents the Euclidean distance of two points p_i and p_j. Now the following metric is used to calculate the distance of the line segments

$$D = (d_1 + d_4)(d_2 + d_3). \tag{39}$$

This metric accurately models the distance of two line segments in different configurations. For instance, when the line segments are identical ($p_a = p_c$, $p_b = p_d$, and $p_a \neq p_b$), D is zero. Similar results are achieved by using the minimum of the two main terms in equation (39), $D = min(d_1 + d_4, d_2 + d_3)$.

Note that it is also possible to use the minimum of the distances as the error metric, $D_{min} = min(d_1, d_2, d_3, d_4)$; however, D_{min} does not distinguish between certain configurations of the line segments. For instance, as shown in Fig. 15, D_{min} outputs zero for two line segments with different configuration but $p_a = p_c$, $(d_1 = 0)$. The mean of the distances, $D_{mean} = mean(d_1, d_2, d_3, d_4)$, is another metric, but it does not output zero for two identical line segments, $p_a = p_c$, $p_b = p_d$, and $p_a \neq p_b$.

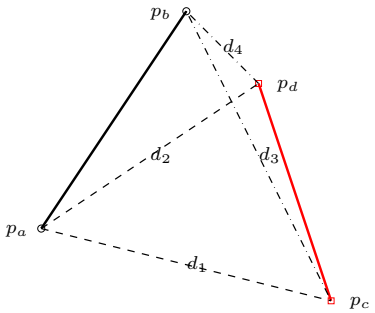

Figure 14: Two line segments are shown in solid black and red. The distance between the two line segments is defined based on the distances of the end-points of the line segments.

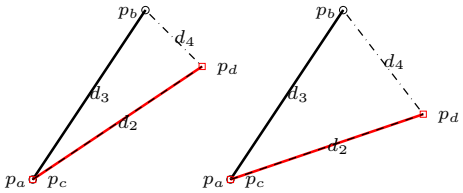

Figure 15: Two line segments are shown in solid black and red in two different configuration. In both cases d_1 is zero. $D_{min} = min(d_1, d_2, d_3, d_4)$ is also zero in both cases.

The process of selecting random peak points and evaluating the transformation continues until for a pre-defined number of iterations or until the error converges. The best transformation is the one that has the lowest error [20].

Table 3 summarizes the comparison of the processing time and verification for all three experiments (results are normalized for one pair of maps). For the third experiment, the Hough spectrum approach fails due to having less overlaps. For the same reason, the Fourier-Hough fails to calculate the transformation. As the results show, the proposed method operates fast and the verification index shows the accuracy of the results.

Experiments with different overlaps show the effectiveness of the proposed algorithm. In the first experiment, RADISH dataset, there was an approximate overlap of 70%. In the Gazebo simulation, the overlap between robots is shown in Table 1. In the

last two real world experiment, the overlap were approximately 31% and 38%.

The time complexity of the proposed algorithm is very close to the RANSAC algorithm; however, RANSAC generates less accurate results. There are a few reasons for this inaccuracy; in the proposed algorithm, the transformation is calculated in two steps: rotation and then translation; however, RANSAC estimates these parameters together. More parameters require many more trials to find the best transformation (RANSAC for each experiment was iterated at least 1500 times.). Moreover, in the proposed algorithm, multiple hypotheses for the rotation angle are considered, unlike RANSAC which has no strategy to deal with multiple hypotheses. Additionally, RANSAC works well if the translation and rotation between the maps is small. For example, in visual odometry, the pose of the camera between two frames does not change substantially and therefore RANSAC can find a reasonably accurate transformation. In a typical map merging problem, the displacement and the rotation is very large.

It is also possible to integrate RANSAC with the proposed algorithm to find an initial estimate and then use the proposed algorithm to refine the results. This technique has been used by other researchers in similar problems such as 3D SLAM [11].

6.3 Shape of World

Most map merging algorithms rely on the orthogonality of the world to find the relative rotation. This is a limiting assumption that makes those algorithms applicable to rectangular shapes with orthogonal lines. An advantage of the proposed algorithm in this work is overcoming this assumption. As mentioned in Algorithm 1, given two maps, the Hough transform is applied to both to generate the Hough images. Then peaks of the Hough images are extracted to find the relative rotation. When the Hough transform is applied, lines in all different angles are extracted. Moreover, in the *Multiple Hypothesis Block*, if there exist multiple hypotheses for rotation, due to the specific shape of the world, it is handled using the rotation verification procedure, as explained. The independence of the proposed algorithm from the orthogonality of the world models has also been verified in the Gazebo experiment, where random blocks are placed in different angles with respect to each other (Fig. 11-a).

6.4 Percentage of Overlap

In general, it is not easy to determine the required overlap between the maps to successfully merge them. This is because maps of different environments have different forms and there is no closed form to represent them mathematically. In this section for a given map, map merging for different amounts of overlaps has been investigated experimentally. The simulated map, shown in Fig. 16, has been divided into two smaller maps with different overlaps, from 10% to 90% (10% overlap means that the map is divided such that 10% of the map is available in both smaller maps). In each of the nine cases, one of the maps is transformed by a predetermined transformation. Then, the proposed algorithm has been applied to the maps. Fig. 17 evaluates map merging for different overlaps. Note that in this case, a ground-truth map exists; therefore, the merged map is compared with the ground-truth map, and the percentage of the matching cells are shown as a similarity score on the y axis. For this highly cluttered map,

when the overlap is more than 50%, the matching is acceptable; however, for a different map, the required overlap might be lower or higher; for example, in the experiment with two robots, the overlap was approximately 31%.

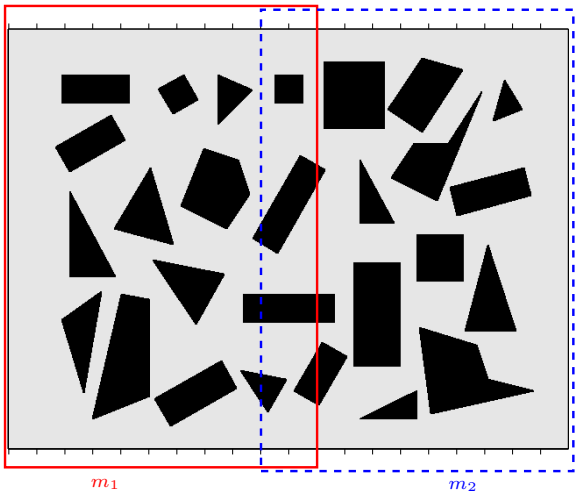

Figure 16: The map has been divided into two smaller maps with known overlap. One of the maps has been transformed by a predetermined transformation. Then the proposed algorithm has been applied to the local maps. Two small maps are shown inside red and blue rectangles. In this case, the overlap is 10%. This process is repeated 9 times with varying overlapping areas.

6.5 Distorted Maps

One of the limitations of any map merging algorithm is handling the error accumulation in the individual maps. Error accumulation in a map usually manifests as a distorted map. For instance, straight walls look bent or curved. In map merging, robots do not have access to raw data (scanning laser measurements, for example) and therefore cannot modify or manipulate the shared maps to remove the distortion directly. In practice, to have a map merging algorithm to work, the maps of the individual robots are self-consistent and accurate.

To evaluate the robustness of map merging against the error accumulation (distortion), an experiment is performed. Two maps from the world shown in Fig. 16 are generated. The experiment is repeated twice with different overlaps, 40% and 70%. In each repetition, similar to the previous discussion, one of the maps is transformed arbitrarily. Then rotational distortion is applied to both maps with respect to the center of the maps. In other words, if a cell is far from the center of the map, it will experience more distortion compared with closer cells. A distorted rotation matrix, shown by R_d,

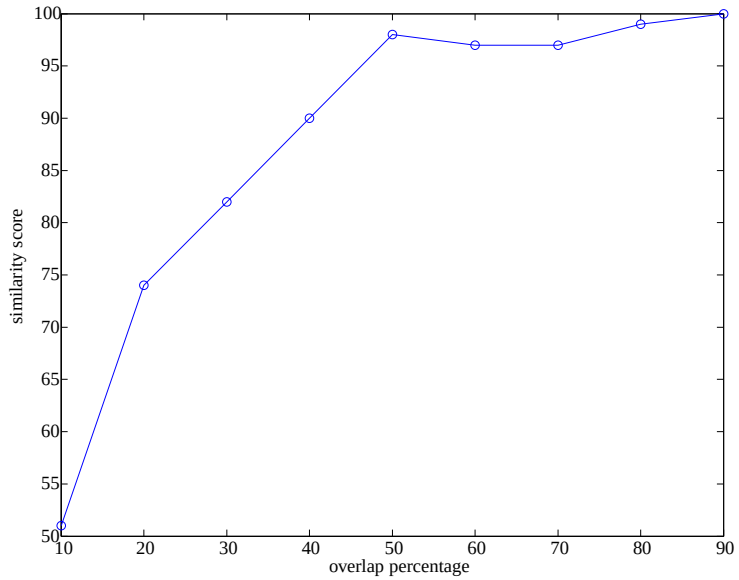

Figure 17: The similarity score for different overlaps. If the overlap is more than 50%, map merging generates acceptable results.

is defined as follows

$$R_d = \begin{bmatrix} cos(d_{ij}\alpha) & sin(d_{ij}\alpha) \\ -sin(d_{ij}\alpha) & cos(d_{ij}\alpha) \end{bmatrix}, \tag{40}$$

where d is the distance of cell i, j from the center of the map, and α is the angle of the distortion. This rotation matrix is applied to both maps. Fig. 18-a shows a map and Fig. 18-b shows the same map, but distorted by $\alpha = 0.04°$. Notice that for this map, the distortion in the center of the image is zeros but on the furthest boundaries of the map it is $16°$, given the size of the map which is 480×640.

For each overlap, the proposed algorithm is applied to the maps for five different distortion angles, α. These values are $0.02°$, $0.04°$, $0.06°$, $0.08°$, and $0.1°$. The distortion on the furthest boundaries of the map is $8°$, $16°$, $24°$, $32°$, and $40°$, respectively. Fig. 19 shows the verification index with respect to different distortions. With the given measure of the distortion, when the maximum distortion is more than $8°$ ($\alpha = 0.02°$), the verification index decreases significantly. Therefore, in order to have map merging operational, the local maps should have very low distortion.

7 Conclusion

Multiple robot map merging in the Hough space has been presented that is fast and robust. The proposed algorithm is designed to work in structured environments. The peaks of the Hough image are used to model the real world and find overlaps between maps. Then the overlaps are used to calculate the relative transformation. Other prop-

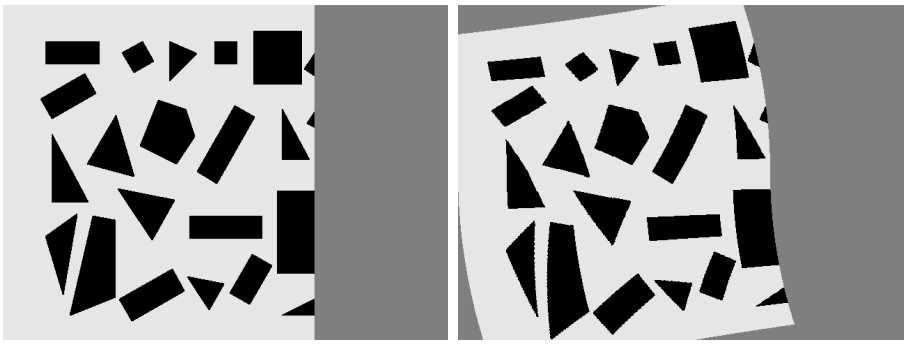

Figure 18: **a)** a map with no distortion, **b)** the same map distorted by $\alpha = 0.04°$. In this case, the maximum distortion, which is $16°$, occurs on the boundaries of the map.

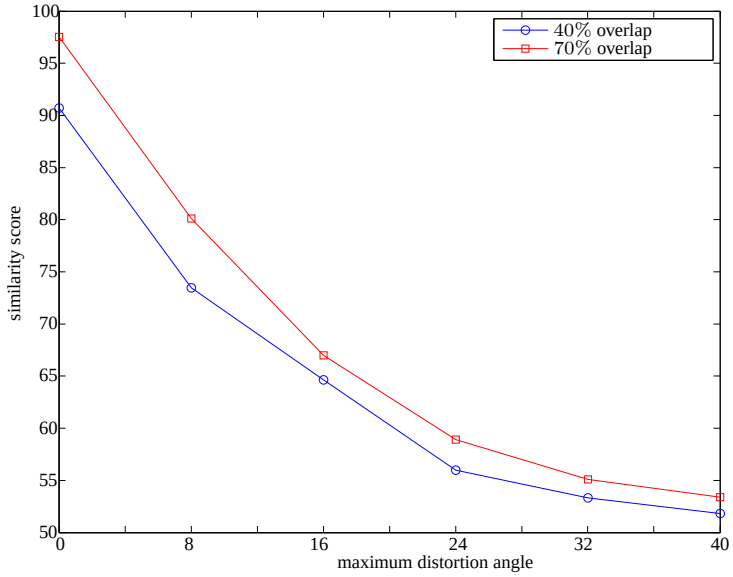

Figure 19: The similarity score for different distortions for two maps with 40% and 70% overlaps. If the distortion increases, map merging fails to generates acceptable results showing that the combination of high distortion and low overlap between the maps is difficult to compensate for.

erties of the Hough transform are used to boost the speed and efficiency of the proposed method. Experiments on datasets and collected data show the effectiveness of the method. Efficiency and processing time comparison with other established methods show that the proposed method is both fast and accurate.

In future work, an efficient iterative peak matching will be investigated to solve for all variables of the transformation matrix at once to further increase the speed.

Also choosing the appropriate time or criteria to fuse maps optimally is another open problem to be investigated.

Acknowledgement

This research is supported by Natural Sciences and Engineering Research Council of Canada (NSERC) and Canada Foundation for Innovation (CFI).

References

[1] Aragues R, Cortes J, Sagues C (2011) Distributed consensus algorithms for merging feature-based maps with limited communication. Robotics and Autonomous Systems 59:163–180

[2] Birk A, Carpin S (2006) Merging occupancy grid maps from multiple robots. Proceedings of the IEEE: Special Issue on Multi-Robot Systems 94(7):1384–1387

[3] Carpin S (2008) Fast and accurate map merging for multi-robot systems. Autonomous Robot 25:305–3160

[4] Carpin S, Birk A, Jucikas V (2005) On map merging. Robotics and Autonomous Systems 53:1–14

[5] Censi A (2008) An ICP variant using a point-to-line metric. In: Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA)

[6] Censi A, Iocchi L, Grisetti G (2005) Scan matching in the Hough domain. In: Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA), pp 2739–2744

[7] Chunhavittayatera S, Chitsobhuk O, Tongprasert K (2006) Image registration using Hough transform and phase correlation. In: Advanced Communication Technology, The 8th International Conference on, pp 973–977

[8] Davies ER (2005) Machine Vision, Theory, Algorithms, Practicalities, 3rd Edition. Morgan Kaufmann

[9] Duda R, Hart P (1972) Pattern Classification and Scene Analysis. John Wiley & Sons, New York

[10] Eliazar A, Parr R (2004) DP-SLAM 2.0. In: Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA)

[11] Endres F, Hess J, Sturm J, Cremers D, Burgard W (2014) 3-D mapping with an RGB-D camera. Robotics, IEEE Transactions on 30(1):177–187

[12] Fischler MA, Bolles RC (1981) Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Communications of the ACM 24(6):381–395

[13] Gifford CM, Webb R, Bley J, Leung D, Calnon M, Makarewicz J, Banz B, Agah A (2010) A novel low-cost, limited-resource approach to autonomous multi-robot exploration and mapping. Robotics and Autonomous Systems 58:186–202

[14] Gil A, Reinoso O, Ballesta M, Miguel J (2010) Multi-robot visual SLAM using a Rao-Blackwellized particle filter. Robotics and Autonomous Systems 58:68–80

[15] Grisetti G, Iocchi L, Nardi D (2002) Global Hough localization for mobile robots in polygonal environments. In: Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA), vol 1, pp 353–358

[16] Grisetti G, Stachniss C, Burgard W (2007) Improved techniques for grid mapping with Rao-Blackwellized particle filters. IEEE Transactions on Robotics 23:34–46

[17] H Huang W, Beevers KR (2005) Topological map merging. The International Journal of Robotics Research 24(8):601–613

[18] Hahnel D, Burgard W, Fox D, Thrun S (2003) An effcient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements. In: Intelligent Robots and Systems (IROS), Proceedings of the IEEE/RSJ International Conference on, pp 206–211

[19] Harati A, Siegwart R (2007) A new approach to segmentation of 2D range scans into linear regions. In: Intelligent Robots and Systems (IROS), Proceedings of the IEEE/RSJ International Conference on, pp 2083–2088

[20] Hartley RI, Zisserman A (2004) Multiple View Geometry in Computer Vision, 2nd edn. Cambridge University Press, ISBN: 0521540518

[21] Howard A (2006) Multi-robot simultaneous localization and mapping using particle filters. International Journal of Robotics Research 25(12):1243–1256

[22] Howard A, Parker L, Sukhatme G (2006) Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection. International Journal of Robotics Research 25(5-6):431–447

[23] Iocchi L, Nardi D (2002) Hough localization for mobile robots in polygonal environments. Robotics and Autonomous Systems 40:43–58

[24] Iocchi L, Mastrantuono D, Nardi D (2001) A probabilistic approach to Hough localization. In: Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA), vol 4, pp 4250–4255

[25] Konolige K, Fox D, Limketkai B, Ko J, Stewart B (2003) Map merging for distributed robot navigationo. In: Intelligent Robots and Systems (IROS), Proceedings of the IEEE/RSJ International Conference on, pp 212–217

[26] LeBlanc K, Saffiotti A (2009) Multirobot object localization: A fuzzy fusion approach. IEEE Transactions on Systems, Man and Cybernetics-Part B 39(5):1259–1276

[27] Leonard JJ, Durrant-Whyte HF (1991) Simultaneous map building and localization for an autonomous mobile robot. In: Intelligent Robots and Systems (IROS), Proceedings of the IEEE/RSJ International Conference on, pp 1442–1447

[28] Lu F, Milios EE (1994) Robot pose estimation in unknown environments by matching 2D range scans. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp 935–938

[29] Nguyen V, Fox D, Martinellii A, Tomatis N, Siegwart R (2005) A comparison of line extraction algorithms using 2D laser rangefinder for indoor mobile robotics. In: Intelligent Robots and Systems (IROS), Proceedings of the IEEE/RSJ International Conference on, pp 1929–1934

[30] Onishi H, Suzuki H (1996) Detection of rotation and parallel translation using Hough and Fourier transfoms. In: Image Processing, Proceedings of the IEEE International Conference on, pp 827–830

[31] RADISH (2007) The robotics data set repository. URL `http://cres.usc.edu/radishrepository/`

[32] Radon J (1986) On the determination of functions from their integral values along certain manifolds. IEEE Transactions on Medical Imaging 5(4):170–176

[33] ROS (2007) Robotic Operating System. URL `http://www.ros.org/wiki/`

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

[35] Smith R, Self M, Cheeseman P (1987) A stochastic map for uncertain spatial relationships. In: Fourth International Symposium of Robotics Research, pp 467–474

[36] Thrun S (2001) A probabilistic on-line mapping algorithm for teams of mobile robots. The International Journal of Robotics Research 20(5):335–363

[37] Thrun S, Liu Y (2005) Multi-robot SLAM with sparse extended information filers. Springer Tracts in Advanced Robotics 15:254–266

[38] Thrun S, Burgard W, Fox D (2005) Probabilistic Robotics. The MIT press, Cambridge, Massachusets, USA

[39] Vaughan R, Gerkey B, Howard A (2003) The Player/Stage project: Tools for multi-robot and distributed sensor systems. In: Proceedings of the International Conference on Advanced Robotics (ICAR 2003), pp 317–323

[40] Yun X, Latt K, Glennon J (1998) Mobile robot localization using the Hough transform and neural networks. In: Proceedings of the Intelligent Control (ISIC), Held jointly with IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), Intelligent Systems and Semiotics (ISAS), pp 393–400

[41] Zhongke L, Xiaohui Y, Lenan W (2003) Image registration based on Hough transform and phase correlation. In: Neural Network and Signal Processing, Proceedings of the IEEE International Conference on, pp 956–959

[42] Zhou XS, Roumeliotis SI (2006) 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, pp 1785–1792