2012 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 7-12, 2012. Vilamoura, Algarve, Portugal

Map Merging Using Hough Peak Matching

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

Abstract— **One of the major problems for multi-robot SLAM is that the robots only know their positions in their own local coordinate frames, so fusing map data can be challenging. 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 data sets and real-world experiments with multiple robotic platforms.

I. Introduction

Simultaneous localization and mapping, or SLAM, first proposed by Smith and Cheeseman [1] and later improved by Leonard and Durrant-Whyte [2] and many others since has become a cornerstone of mobile robotics. 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 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. It is more desirable for the relative poses to be determined from the maps themselves to provide maximum flexibility of the solution.

This paper presents a solution to this problem using the Hough transform. It will be shown that it is more efficient to perform the map merging operation in the Hough space because of the unique representation of the obstacles in an occupancy grid map. As a result, there is no need to perform any extensive searching and the map merging algorithm scales linearly with map size.

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 III will outline the proposed methods based on the Hough transforms. Section IV will present the results and finally some conclusions are presented in Section V.

†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

II. Background of Research

Many past approaches to multi-robot SLAM have used a feature-based world representation where maps are represented by distinguishable objects. For example, in [3] a feature-based multiple robot SLAM is proposed which is based on the information filter. In [4] a classic EKF-based solution is proposed. In [5] a feature based visual SLAM using particle filtering is proposed. A feature-based multiple robot localization is proposed in [6] where fuzzy sets are used to represent uncertain position information and fuzzy intersection is used to fuse data. In [7] an algorithm for merging feature-based maps with limited communication is introduced.

A. Map Merging with View Based SLAM

View-based SLAM differs from feature-based SLAM in that no-features are explicitly extracted. Instead, entire scans are matched.

In [8] 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 when there are fewer similar patterns in both maps. The method proposed in [9] is based on the solution proposed in [8], except that individual robots are using Distributed Particle SLAM (DP-SLAM) version 2.0 [10] for filtering. The method proposed in [8] and used in [9] is a highly time consuming algorithm. This is problematic especially for the large scale maps which are a typical problem in indoor environments. In [9] maps are updated by applying a simple rule: the minimum occupancy grid value between the two maps is used for merging. A similar method is proposed in [11] with simulated annealing and hill climbing used to merge maps. This method also becomes ineffective in maps with less overlaps.

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 [12]. The rotation and translation are derived in separate steps similar to our approach. However, 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.

In [13] a multi-robot SLAM algorithm is proposed based on the 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 find their relative initial position. The proposed method finds the initial state of the robots but it can not be applied when robots can not see each other.

In [14] a probabilistic multiple robot view-based SLAM is proposed. The limitation with this method is that the initial pose of the robots is assumed to be known.

In [15], multiple robot SLAM based on topological map merging using both structural and geometrical characteristics of the Voronoi graph is proposed. The topological map is built on the occupied space as opposed to the free space.

B. 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:

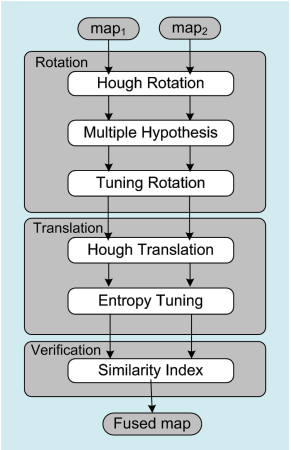
- **Modeling:** A mathematical representation for objects in maps is necessary to compare objects.
- **Partial views:** Map features from different view angles may look different.
- **Geometric relations:** The internal relation of map features is an inherent property that makes each map unique.

Performing this search in Euclidian space is difficult. However, by transforming the map into a different representation, certain image properties can be exploited to make the problem easier to solve. 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.

The basic form of Hough transform, as shown in Fig 1, maps every (x, y) point from the image space into the (θ, ρ) space, known as the Hough space, based on the this relation: $\rho = xcos\theta + ysin\theta$.

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 R_ψ and translation $T = \begin{bmatrix}\delta_x & \delta_y\end{bmatrix}^T$, 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 [16] (the superscript ^{T} denotes the matrix transpose):

$$\theta' = \theta + \psi \text{ and } \rho' = \rho + \begin{bmatrix}\cos(\theta + \psi) & \sin(\theta + \psi)\end{bmatrix} T. \quad (1)$$



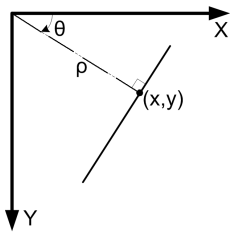


Fig. 1. (**left**) The proposed map fusion algorithm. The two input maps are fused by finding their relative transformation matrix. No prior information is available regarding the relative position of two respective robots. (**right**) Hough modeling.

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, appears as a point with intensity of L in Hough space as

$$\theta = -\frac{1}{(\arctan a)},\ \rho = b\sin\theta \text{ and } \mathcal{H}(\theta, \rho) = L. \quad (2)$$

III. Map Merging with Hough Transforms

Fig. 1 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. According to the diagram of the algorithm, in three steps, first the rotation is found, then the translation, and finally the results are verified. The detail of each step is explained in next sections.

A. Extracting Relative Orientation of Maps

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. The same assumption applies to m_2 where m exists in both maps and Δ_2 exists only in m_2:

$$m_1 \triangleq \{m\} \cup \{\Delta_1\}\ ,\ m_2 \triangleq \{m\} \cup \{\Delta_2\}. \quad (3)$$

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

$$\mathcal{H}_{m_1}(\theta, \rho) = \mathcal{H}_m(\theta, \rho) + \mathcal{H}_{\Delta_1}(\theta, \rho). \quad (4)$$

Assuming that R_ψ and $T = \begin{bmatrix}\delta_x & \delta_y\end{bmatrix}^T$ are transformation elements that can fuse these two maps, then according to **Property 1** and **Property 2**, the Hough image of the transformed m_2 becomes:

$$\mathcal{H}'_{m_2}(\theta, \rho) = \mathcal{H}_m(\theta+\psi, \rho+\begin{bmatrix}\cos(\theta + \psi) & \sin(\theta + \psi)\end{bmatrix} T)+ H_{\Delta_2}(\theta + \psi, \rho + \begin{bmatrix}\cos(\theta + \psi) & \sin(\theta + \psi)\end{bmatrix} T). \quad (5)$$

Ideally, a perfect match requires that $\mathcal{H}_{m_1} = \mathcal{H}'_{m_2}$, but because of the non-overlapping sections Δ_1 and Δ_2, this equality becomes approximate:

$$\mathcal{H}_{m_1} \approx \mathcal{H}'_{m_2}. \quad (6)$$

Let us assume there exists some function $f(\cdot)$, taking an array as its input and outputting a scalar, that is applied to (6) at a fixed angle, θ_o. Therefore the input of $f(\cdot)$ is dependent only on the second input of the Hough image.

$$f(\mathcal{H}_{m_1}\big|_{\theta_o}) \approx f(\mathcal{H}'_{m_2}\big|_{\theta_o}). \tag{7}$$

where $f(\mathcal{H}_{m_1}\big|_{\theta_o})$ means that the input of $f(\cdot)$ is one column of the image $\mathcal{H}_{m_1}$ which corresponds to the angle θ_o. By applying $f(\cdot)$ to (6) the following outcome is desired:

$$h_1(\theta_o) \approx h_2(\theta_o + \psi). \tag{8}$$

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. (8) 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 (8) to all angles the following general relation is achieved:

$$h_1(\theta) \approx h_2(\theta + \psi). \tag{9}$$

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:

$$\Psi_o = h_1(\theta) \circledast h_2(\theta + \psi). \tag{10}$$

where $\circledast$ is the circular cross correlation operation. 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 n rotation candidates defined:

$$\Psi_o = \{\psi_o^i\}_{i=1}^n \tag{11}$$

In the rotation verification process, by another method which is explained in Algorithm 1, only one estimate is calculated, called $\hat{\psi}$. 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:

$$\psi_o = \underset{i}{\operatorname{argmin}}(|\psi_o^i - \hat{\psi}|), \tag{12}$$

which selects the best rotation candidate as the one closest to $\hat{\psi}$.

[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.

To calculate the approximate rotation, 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 in the maps. By comparing these peaks it is possible to find overlaps in both maps. These overlaps can be 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 correspond to the lengths of the line segments.

Algorithm 1 shows the peak association and rotation calculation. Inputs to the algorithm are the local peaks of Hough images of map_1 and map_2 which are denoted by P_k where $k = 1, 2$. K is the number of robots, but here the algorithm is explained for two robots for simplicity. P_k is defined as

$$P_k = \{p_k^i\}_{i=1}^{n_k}, p_k^i = (\theta_k^i, \rho_k^i, v_k^i). \tag{13}$$

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 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:

$$C = \{(i_n, j_n)\}, i = 1, .., n_1, j = 1, .., n_2, n = 1, .., N \tag{14}$$

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

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.

Initially C is empty (line 1). For each peak from P_1, its candidate associated point from P_2 is found by a simple search algorithm (line 2-3). If the difference of peak values is less than a threshold (line 4-5), then points are considered to be associated and added to C (line 6-7). After finding all correspondences, the slopes of the lines are calculated based on **Property 3**. The average difference of slopes provides an approximate estimate for the relative rotation (line 9).

The rotation can be tuned by performing a comparison between the peak points of the Hough images of m_1 and rotated m_2. 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 $\alpha_2 = \psi_o + \Lambda$.

$$\psi = \underset{\theta}{\operatorname{argmin}}\, \mathcal{H}_{\theta=\alpha_1:\alpha_2}(m_1) - \underset{\theta}{\operatorname{argmin}}\, \mathcal{H}_{\theta=\alpha_1:\alpha_2}(m_2), \tag{15}$$

where $\mathcal{H}_{\theta=\alpha_1:\alpha_2}(map)$ 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 2Λ is an arbitrary small interval in which the resolution of the Hough image is higher than 1^o resolution used in the previous step. ψ is the tuned rotation.

Algorithm 1 Multiple hypothesis handling.

Input: P_1, P_2, v_{gate}
Output: $\hat{\psi}$

1: $C \leftarrow \emptyset$
2: **for** $i^\star = 1 \rightarrow |P_1|$ **do**
3: $\quad j^\star \leftarrow \underset{j}{\operatorname{argmin}} |v_1^{i^\star} - v_2^j|$
4: $\quad \delta_v \leftarrow |v_1^{i^\star} - v_2^{j^\star}|$
5: $\quad$ **if** $\delta_v < v_{gate}$ **then**
6: $\quad\quad C \leftarrow C + (i^\star, j^\star)$
7: $\quad$ **end if**
8: **end for**
9: $\hat{\psi} = \frac{1}{|C|} \sum_{n=1}^{|C|} -(\frac{1}{\tan\theta_1^{i_n}} - \frac{1}{\tan\theta_2^{j_n}})$

B. 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 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 (13), 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 (14), 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 $\mathcal{H}'_2$):

$$\rho'_i = \rho_i + \begin{bmatrix}\cos\theta_i & \sin\theta_i\end{bmatrix} T_o \tag{16}$$

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

$$\begin{bmatrix}\rho'_1 - \rho_1 \\ \vdots \\ \rho'_N - \rho_N\end{bmatrix} = \begin{bmatrix}\cos\theta_1 & \sin\theta_1 \\ \vdots & \vdots \\ \cos\theta_N & \sin\theta_N\end{bmatrix} T_o. \tag{17}$$

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

$$T_o = (A^T A)^{-1} A^T B. \tag{18}$$

Algorithm 2 Translation by matching peaks of Hough images

Input: P_1, P'_2, d_{gate}, v_{gate}
Output: T_o

1: $C \leftarrow \emptyset$
2: **for** $\alpha = -90° \rightarrow \alpha = 90°$ **do**
3: $\quad (i^\star, j^\star) = \underset{\mathcal{U}}{\operatorname{argmin}}(|v_1^i - v_2^j|, |\rho_1^i - \rho_2^j|) \Big| \begin{matrix}\theta_1^i = \alpha \\ \theta_2^j = \delta_\alpha\end{matrix}$
4: $\quad \delta_v \leftarrow |v_1^{i^\star} - v_2^{j^\star}|$
5: $\quad \delta_\rho \leftarrow |\rho_1^{i^\star} - \rho_2^{j^\star}|$
6: $\quad$ **if** $\delta_v < v_{gate}$ and $\delta_\rho < d_{gate}$ **then**
7: $\quad\quad C \leftarrow C + (i^\star, j^\star)$
8: $\quad$ **end if**
9: **end for**
10: Calculate A and B from C based on (17).
11: $T_o = (A^T A)^{-1} A^T B$

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

$$\mathbf{H}(map) = -\sum p \log_2(p), \tag{19}$$

where p is the normalized histogram of map. 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. The following relation gives the best translation vector:

$$T = \begin{bmatrix}\delta_x, \delta_y\end{bmatrix}^T = \underset{\mathcal{S}}{\operatorname{argmin}}\{\mathbf{H}(J(map_1, T_{x,y}(map'_2)))\}$$

$$\mathcal{S} = \{(x,y) \subset \mathbb{N}^2 | \delta_{x_o} - \Delta_x < x < \delta_{x_o} + \Delta_x, \\ \delta_{y_o} - \Delta_y < y < \delta_{y_o} + \Delta_y\} \tag{20}$$

where $T_{x,y}(map)$ means that a map is translated according to the values of x and y. $J(m_1, m_2)$ means both maps are in the same coordinate frame according to the coordinates of the first input, m_1. Final probability of each cell of the fused map is generated by additivity of the log odds of the probability of the occupancy grid maps. $\mathcal{S}$ 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.

C. Verification of Results

A performance index, called the similarity index [8] is used and measures the similarity of two maps over some desired region. This index is composed of two components:

$$agr(map_1, map_2) = |\{p = (x, y) | map_1(p) = map_2(p)\}|,$$
$$dis(map_1, map_2) = |\{p = (x, y) | map_1(p) \neq map_2(p)\}|,$$

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

$$V(m_1, m_2) = \frac{agr(m_1, m_2) \times 100\%}{agr(m_1, m_2) + dis(m_1, m_2)}, \qquad (21)$$

where m_1 and m_2 are two input maps and $V(m_1, m_2)$ is the verification index. If $V(m_1, m_2)$ is close to 100%, then the proposed transformation can be accepted.

IV. Experimental Results

To demonstrate the effectiveness of the proposed method, three experiments are presented. The experimental robots are built by CoroWare and each is equipped with encoders and a Hokuyo laser ranger (Fig. 5-a). SLAM on the individual robots is performed with the particle filter [13].

A. RADISH data set, Fort AP Hill

The first experiment is performed on RADISH Fort AP Hill data set [17]. Figs. 2-a and Fig. 2-b show the two maps before fusion. The maps have about 75% overlap and are rotated about 5° relative to one another. The proposed method is used to find the relative rotation between the maps. Fig. 3-a shows $h_1(\theta)$ and $h_2(\theta + \psi)$ in blue and red colors. $h_1(\theta)$ is output of $f(\cdot)$ on map_1 and $h_2(\theta + \psi)$ is 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. 3-b. The result has four local peaks, $\Psi_o = \{-5^\circ, 85^\circ, 175^\circ, -95^\circ\}$. These peaks are considered as potential candidates for ψ_o. As this figure shows, one of four candidate is close to the estimated -5° that mentioned. To compare the effect of other alternatives for $f(\cdot)$, *averaging over non zero elements* of the given input vectors has been examined. Fig. 3-c and Fig. 3-d show $h_1(\theta)$ and $h_2(\theta + \psi)$ and the result of correlation. This time the results have a bigger offset from the required estimated value. The *Entropy* of 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. By applying Algorithm 1, an approximate rotation is extracted as $\hat{\psi} = -15^\circ$ which is close to -5°. Therefore, based on (12) the rest of the experiment continues with $\psi_o = -5^\circ$. After the tuning method ψ becomes -5.5°. The result of the final alignment is shown in Fig. 2-c. Now the next step, which is finding the translation, is started. In this step first Hough image of map_1, m_1 and rotated map_2, m_2' is generated. These two images are shown in Fig. 4-a and 4-b respectively. Using the data association of algorithm 2 and solving (18), the initial estimate for the translation T_o is $[23, 1]^T$. v_{gate} and d_{gate} for all experiment are 10 and 30. Then using the exhaustive search and evaluating the image entropy the translation is tuned to be $[27, 2]^T$. Fig. 4-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. 2-d shows the final alignment of both maps. The verification index, defined in (21) is 94%.

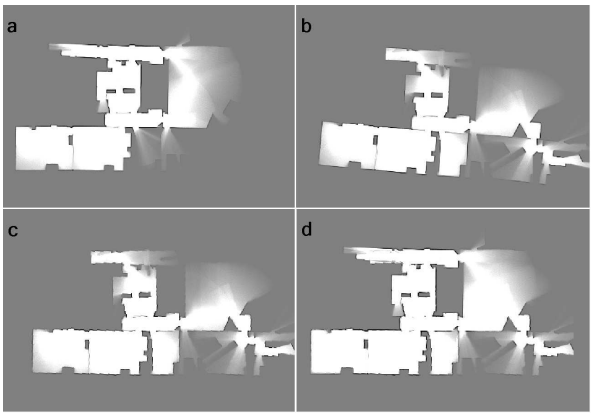


Fig. 2. **a)** map_1, **b)** map_2, **c)** map_2 after tuned rotation, **d)** two maps after transformation.

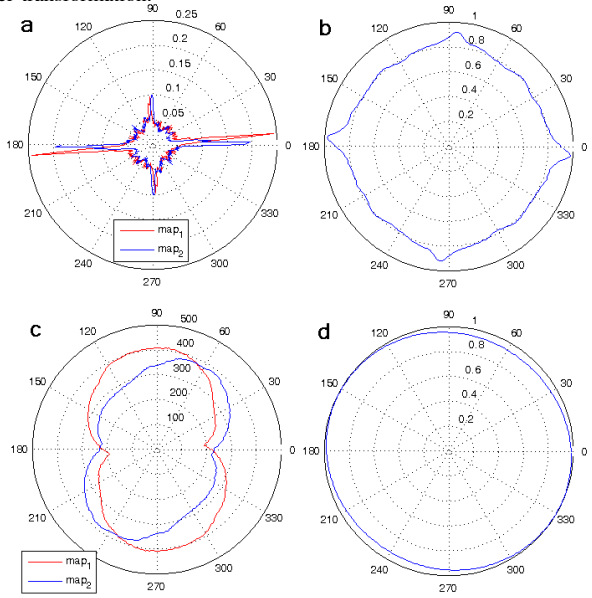


Fig. 3. **a)** $h_1(\theta)$ and $h_2(\theta+\psi)$ when $f(\cdot)$ is $max(\cdot)$, **b)** Result of circular cross correlation. Four local peaks are at -5°, 85°, 175° and -95°, **c)** $h_1(\theta)$ and $h_2(\theta + \psi)$ when $f(\cdot)$ is averaging, **d)** Result of circular cross correlation. Two local peaks are -30° and 150°.

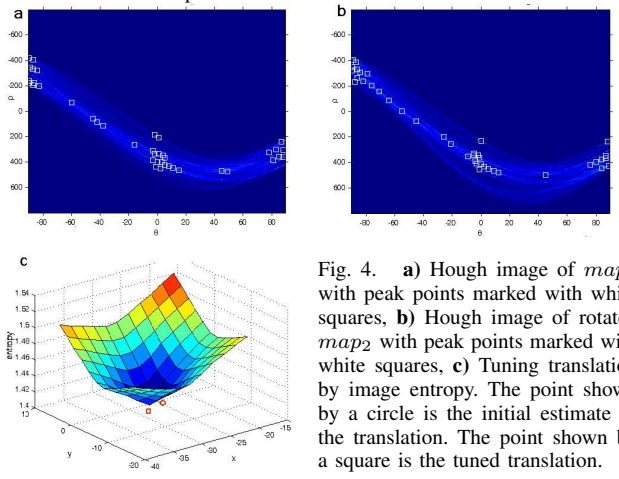


Fig. 4. **a)** Hough image of map_1 with peak points marked with white squares, **b)** Hough image of rotated map_2 with peak points marked with white squares, **c)** Tuning translation by image entropy. The point shown by a circle is the initial estimate of the translation. The point shown by a square is the tuned translation.

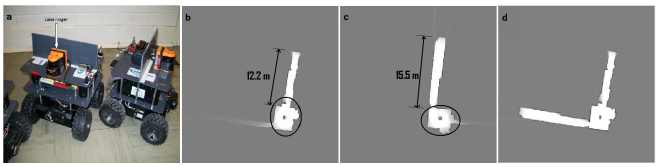


Fig. 5. **a)** experimental robots, **b)** map_1, **c)** map_2, overlaps of two maps are shown by ellipses, **d)** fused maps.

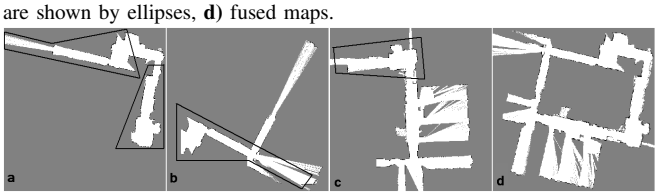


Fig. 6. Three partial maps are fused together to generate a global map. (**a**) is the base map where (**b**) and (**c**) are fused to that. Overlaps are marked with polygons. (**d**) Final fused map which depicts loop closure.

B. Real Experiment with Two Vehicles

A real world experiment is performed with two CoroBot robots (Fig.5-a) in an indoor environment in the basement of the University of New Brunswick. The total area is $81.02m^2$. Fig.5-b and Fig.5-c show the maps of the robots. 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 [12] fails in this experiment due to the small overlap). 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.5-c shows final maps after fusion.

C. 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. 6-a, b and c show three partial maps. Maps of Fig. 6-b and c are fused to Fig. 6-a. Overlaps of Fig. 6-a with other two maps are enclosed inside polygons. The fused global map of the environment is shown in Fig. 6-d.

D. Comparisons

As mentioned, a major benefit of 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 [8] and Map Segmentation [18] methods.

Table I summarizes the comparison of the processing time and verification for all three experiments. As the results show, the proposed method operates faster and the verification index 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 (%)		
Hough peak matching	14	11	14	94	92	94
Map segmentation [18]	105	83	106	95	92	94
ARW map merging [8]	168	150	152	93	88	92

V. CONCLUSION

Multiple-robot map merging in the Hough space has been presented that is fast and robust. 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. Experiments on data sets and collected data show the effectiveness of the method.

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

ACKKNOWLEDGEMENT

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

REFERENCES

[1] R. Smith, M. Self, and P. Cheeseman, "A stochastic map for uncertain spatial relationships," in *Fourth International Symposium of Robotics Research*, 1987, pp. 467–474.

[2] J. Leonard and D.-W. H.F., "Simultaneous map building and localization for an autonomous mobile robot," in *Intelligent Robots and Systems (IROS), Proceedings of the IEEE/RSJ International Conference on*, 1991, pp. 1442–1447.

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

[4] 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.

[5] A. Gil, O. Reinoso, M. Ballesta, and J. Miguel, "Multi-robot visual slam using a rao-blackwellized particle filter," *Robotics and Autonomous Systems*, vol. 58, pp. 68–80, January 2010.

[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] R. Aragues, J. Cortes, and C. Sagues, "Distributed consensus algorithms for merging feature-based maps with limited communication," *Robotics and Autonomous Systems*, vol. 59, pp. 163–180, March 2011.

[8] 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.

[9] C. M. Gifford, R. Webb, J. Bley, D. Leung, M. Calnon, J. Makarewicz, B. Banz, and A. Agah, "A novel low-cost, limited-resource approach to autonomous multi-robot exploration and mapping," *Robotics and Autonomous Systems*, vol. 58, pp. 186–202, February 2010.

[10] A. Eliazar and R. Parr, "Dp-slam 2.0," in *In Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2004.

[11] S. Carpin, A. Birk, and V. Jucikas, "On map merging," *Robotics and Autonomous Systems*, vol. 53, pp. 1–14, 2005.

[12] S. Carpin, "Fast and accurate map merging for multi-robot systems," *Autonomous Robot*, vol. 25, pp. 305–3160, 2008.

[13] 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.

[14] S. Thrun, "A probabilistic on-line mapping algorithm for teams of mobile robots," *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 335–363, 2001.

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

[16] L. Iocchi and D. Nardi, "Hough localization for mobile robots in polygonal environments," *Robotics and Autonomous Systems*, vol. 40, 2002.

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

[18] 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 Conf. on*, 2011.