Multi-robot SLAM with Unknown Initial Correspondence: The Robot Rendezvous Case

Xun S. Zhou and Stergios I. Roumeliotis

Department of Computer Science & Engineering, University of Minnesota, Minneapolis, MN

Abstract— This paper presents a new approach to the multi-robot map-alignment problem that enables teams of robots to build joint maps without initial knowledge of their relative poses. The key contribution of this work is an optimal algorithm for merging (not necessarily overlapping) maps that are created by different robots independently. Relative pose measurements between pairs of robots are processed to compute the coordinate transformation between any two maps. Noise in the robot-to-robot observations, propagated through the map-alignment process, increases the error in the position estimates of the transformed landmarks, and reduces the overall accuracy of the merged map. When there is overlap between the two maps, landmarks that appear twice provide additional information, in the form of constraints, which increases the alignment accuracy. Landmark duplicates are identified through a fast nearest-neighbor matching algorithm. In order to reduce the computational complexity of this search process, a kd-tree is used to represent the landmarks in the original map. The criterion employed for matching any two landmarks is the Mahalanobis distance. As a means of validation, we present experimental results obtained from two robots mapping an area of 4,800 m^2.

I. INTRODUCTION

One of the most challenging problems in autonomous system navigation is reliable localization. Many robotic applications, such as search and rescue, surveillance, planetary exploration etc., require accurate localization within unknown environments. When robots operate in unmapped or GPS-denied areas, achieving accurate localization requires that they create and maintain a map of their surroundings. The problem of building a map while exploring an unknown territory is commonly referred to as Simultaneous Localization and Mapping (SLAM) [1]. In order to increase the accuracy and efficiency when mapping large areas, it is often necessary for two or more robots to participate in this task. This process is known as Multi-robot SLAM or Cooperative SLAM (C-SLAM). While enhancing efficiency, the complexity of SLAM increases when robots cooperate to construct a single, joint map of the area they explore. This problem becomes especially challenging when the coordinate transformation between the initial poses of the robots is unknown. We call this transformation the *initial correspondence*. The main contribution of this paper is an algorithm for merging independently created maps with unknown initial correspondence.

In order to fuse maps created by different robots, the transformation between their coordinate frames needs to be determined. This can be achieved in two ways. The first is to search for landmark matches in the two maps. The most probable transformation is the one that produces the maximum number of landmark correspondences. In this case, the underlying assumption is that the two maps overlap. Alternatively, robot-to-robot measurements can be used for computing the unknown coordinate transformation. When two robots meet and measure their relative distance and bearing, this information can be used to compute the transformation required for merging the two maps. Due to noise in these measurements, the estimated transformation may be inaccurate, which in effect will reduce the quality of the merged map. In our approach, we use a combination of the above methods. We first align the two maps employing the transformation determined by the robot-to-robot measurements. Then, we examine whether there is an overlap between the two maps by searching for landmark matches. If landmark duplicates are identified, this information is used to impose constraints that improve the accuracy of the resulting map.

II. RELATED WORK

Map merging with *unknown* initial robot correspondence and no assumption of map overlap, is a quite challenging problem. In [2], the map merging decision is made in two steps. When the two robots are in communication range but do not know their relative locations, one of the robots receives sensor data from the other robot and attempts to estimate its location by matching the received scan patch against its own map. This is the hypothesis generation step. The next step is to verify a location hypothesis. The two robots try to meet based on the assumed common map. If they fail to meet at the expected location, the hypothesis is rejected. If they succeed, the hypothesis is accepted and their maps are combined permanently.

A Maximum Likelihood approach to the map merging problem with *unknown* initial correspondence is presented by Howard in [3]. In this work manifolds are employed for representing maps created from two-dimensional laser scans. The C-SLAM problem is divided into three sub-problems: incremental localization and mapping, loop closure, and island merging. As a robot moves within an area, incremental additions are made to the map. Loop closure and/or island merging are triggered by mutual observations of the robots that add new constraints between previously unrelated patches. With these constraints, loops are closed and/or previously unrelated maps are merged together. The manifold representation is self-consistent at the cost of increased computational complexity. Additionally, since a Maximum Likelihood estimator processes data in a batch mode, when new measurements

arrive, all data need to be reprocessed, thus further increasing the computational complexity.

Howard's work is closely related to ours. Instead of expensive manifold operations and scan matching, we use an Extended Kalman Filter (EKF) to estimate the robots' and landmarks' positions. Since the EKF is a sequential estimator, previously recorded measurements do not need to be reprocessed, which reduces both memory and computational requirements. Mutual observations are employed to quickly align two robots' local maps. We then search for identical landmarks by landmark matching rather than scan matching. If landmark duplicates are identified, the accuracy of the merged map increases as a result of imposing constraints on their position estimates.

III. PROBLEM FORMULATION

In this section, we address the problem of map merging, that is, aligning the robots' independent maps to create a single global map of the environment. Our approach requires that the robots meet at least once. This can be a random event or can be arranged by the two robots.[1] We solve the map alignment problem by processing mutual *relative distance and bearing measurements*, when the two robots are within sensing range of each other. The transformation between the two maps is computed using this pair of *robot-to-robot* measurements. If the maps created by the two robots overlap, duplicate landmarks can be identified in the merged map.[2] By imposing constraints on the positions of matched landmarks, the quality of the resulting map is significantly improved. Once the two maps are merged, the two robots can continue to cooperatively explore their environment while expressing all new measurements with respect to the common frame of reference. In order to simplify the presentation of the ensuing derivations, we hereafter describe the solution of the two-robot case in a 2D environment. This can be easily extended to the case of larger robot teams.

A. Relative Distance and Bearing Measurements

Consider two robots, R_1 and R_2. Each of them has explored and mapped the area it has travelled through with respect to its initial global frame, $\{G_1\}$ and $\{G_2\}$, respectively. The origins of the global frames are usually set to coincide with the robots' starting locations. The coordinates of R_i with respect to $\{G_i\}$ are $^{G_i}\boldsymbol{X}_{R_i} = [x_i \ y_i \ \phi_i]^T$, $i = 1, 2$. We assume that each robot has a range sensor which can measure the distance ρ and bearing θ towards a target. The measurement of the relative position of robot j with respect to robot i is described by:

$$^i\mathbf{z}_j = \begin{bmatrix} ^i\rho_m \\ ^i\theta_{jm} \end{bmatrix} = \begin{bmatrix} \rho \\ ^i\theta_j \end{bmatrix} + \begin{bmatrix} \eta_{i\rho} \\ \eta_{i\theta_j} \end{bmatrix} \quad i, j = 1, 2$$

[1] A detailed description of the latter case is beyond the scope of this work. The interested reader is referred to [4] for a detailed discussion.

[2] In general, when two robots explore a connected area, their maps will at least partially overlap, e.g., around the point where they meet. In the event that no overlap occurs, the map merging process described in this work is still valid and the robots can perform C-SLAM after they meet, which will increase the accuracy of the merged map.
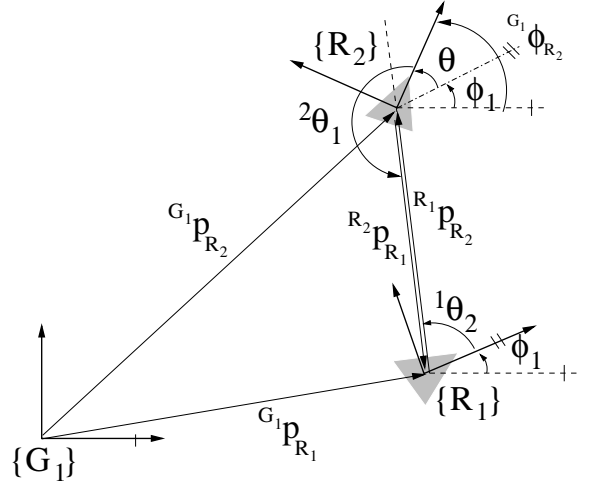

Fig. 1. Relative orientation θ between the two robot frames $\{R_1\}$ and $\{R_2\}$ as a function of the two bearing measurements $^1\theta_2$ and $^2\theta_1$; Position and orientation of robot R_2 expressed with respect to $\{G_1\}$.

where ρ is the distance between the two robots, $^i\theta_j$ is the bearing towards robot R_j, and $\eta_{i\rho}$, $\eta_{i\theta_j}$ are white zero-mean Gaussian noise processes with variances $\sigma_{i\rho}^2$ and $\sigma_{i\theta_j}^2$ respectively.

Since the two distance measurements $^1\rho_m$, and $^2\rho_m$ are independent, a more accurate estimate of the distance ρ between the two robots can be computed as the weighted average of the two individual measurements, i.e.,

$$\rho_m = \sigma_\rho^2 \left(\frac{^1\rho_m}{\sigma_{1\rho}^2} + \frac{^2\rho_m}{\sigma_{2\rho}^2} \right), \quad \frac{1}{\sigma_\rho^2} = \frac{1}{\sigma_{1\rho}^2} + \frac{1}{\sigma_{2\rho}^2} \quad (1)$$

We form the combined measurement vector as

$$\mathbf{Z} = \begin{bmatrix} \rho_m \\ ^1\theta_{2m} \\ ^2\theta_{1m} \end{bmatrix} = \begin{bmatrix} \rho \\ ^1\theta_2 \\ ^2\theta_1 \end{bmatrix} + \begin{bmatrix} \eta_\rho \\ \eta_{1\theta_2} \\ \eta_{2\theta_1} \end{bmatrix} = \mathbf{Z}_t + \boldsymbol{\eta} \quad (2)$$

where $E[\eta_\rho^2] = \sigma_\rho^2$ and $\mathbf{Z}_t$ denotes the vector of the real distance and relative bearings.

As depicted in Fig. 1, the following geometric constraint holds:

$$^{R_1}\mathbf{p}_{R_2} = -^{R_1}_{R_2}\mathbf{C}(\theta) \ ^{R_2}\mathbf{p}_{R_1} \quad (3)$$

with

$$^{R_j}\mathbf{p}_{R_i} = \rho \begin{bmatrix} \cos {}^j\theta_i \\ \sin {}^j\theta_i \end{bmatrix} = \rho \mathbf{C}(^j\theta_i)\mathbf{e}_1, \quad i = 1, 2 \quad (4)$$

where $\mathbf{e}_1$ is the unit vector along the x-axis and $^{R_j}\mathbf{p}_{R_i}$ is the position of robot R_i in robot R_j's local frame, $^{R_1}_{R_2}\mathbf{C}(\theta)$ is the rotational matrix that describes the angular transformation between robot frames $\{R_1\}$ and $\{R_2\}$.

Substituting from Eq. (4) in Eq. (3), the rotational angle θ can be computed as:
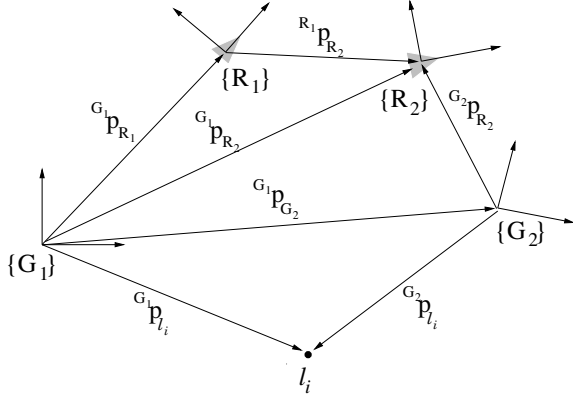
$$\theta = \pi + {}^1\theta_2 - {}^2\theta_1 \quad (5)$$

Fig. 2. Relative position $^{G_1}p_{G_2}$ between the origins of $\{G_2\}$ and $\{G_1\}$ as a function of $^{G_1}p_{R_1}$, $^{R_1}p_{R_2}$, and $^{G_2}p_{R_2}$; Position of robot R_2 expressed with respect to $\{G_1\}$; Position of landmark ℓ_i expressed with respect to $\{G_1\}$.

Once the angle θ (cf. Eq. (5)) and the distance between the robots R_1 and R_2 (cf. Eq. (1)) are computed, the transformation $(^{R_1}\mathbf{p}_{R_2}, {}_{R_2}^{R_1}\mathbf{C}(\theta))$ between $\{R_1\}$ and $\{R_2\}$ is uniquely determined in closed form.

B. Transformation from Global Frame $\{G_2\}$ to $\{G_1\}$

In this section, given the distance and bearing measurements recorded by R_1 and R_2, our goal is to determine the transformation $(^{G_1}\mathbf{p}_{G_2}, {}_{G_2}^{G_1}\mathbf{C}(\phi))$ between the global coordinate frames $\{G_1\}$ and $\{G_2\}$. This will allow us to express the state estimates $^{G_2}\mathbf{X}_2$ of robot R_2 with respect to $\{G_1\}$, i.e., determine the relation

$$^{G_1}\mathbf{X}_2 = \mathbf{h}(^{G_1}\mathbf{X}_1, {}^{G_2}\mathbf{X}_2, \mathbf{Z}) \qquad (6)$$

where $\mathbf{Z}$ is the vector described in Eq. (2).

Initially, each of the two robots performs SLAM independently, i.e., robots R_1 and R_2 estimate the vectors

$$^{G_1}\mathbf{X}_1 = \begin{bmatrix} ^{G_1}\boldsymbol{X}_{R_1}^T & ^{G_1}\boldsymbol{X}_{L_1}^T \dots {}^{G_1}\boldsymbol{X}_{L_i}^T \dots {}^{G_1}\boldsymbol{X}_{L_{n_1}}^T \end{bmatrix},$$

$$^{G_2}\mathbf{X}_2 = \begin{bmatrix} ^{G_2}\boldsymbol{X}_{R_2}^T & ^{G_2}\boldsymbol{X}_{\ell_1}^T \dots {}^{G_2}\boldsymbol{X}_{\ell_j}^T \dots {}^{G_2}\boldsymbol{X}_{\ell_{n_2}}^T \end{bmatrix} \quad (7)$$

respectively, with

$$^{G_1}\boldsymbol{X}_{L_i} = \begin{bmatrix} x_{L_i} \\ y_{L_i} \end{bmatrix} \in M_1, \; ^{G_2}\boldsymbol{X}_{\ell_j} = \begin{bmatrix} x_{\ell_j} \\ y_{\ell_i} \end{bmatrix} \in M_2 \quad (8)$$

where $i = 1 \dots n_1$, $j = 1 \dots n_2$, and $n_1(n_2)$ is the total number of landmarks in the map $M_1(M_2)$ created by robot $R_1(R_2)$.

The rotational transformation matrix $_{G_2}^{G_1}\mathbf{C}$ is readily computed from:

$$\begin{aligned} _{G_2}^{G_1}\mathbf{C}(\phi) &= {}_{R_1}^{G_1}\mathbf{C}(\phi_1) \, _{R_2}^{R_1}\mathbf{C}(\theta) \, _{R_2}^{G_2}\mathbf{C}(\phi_2)^T \\ \Rightarrow \phi &= \phi_1 + \theta - \phi_2 \end{aligned} \qquad (9)$$

Similarly, the rotational transformation of $\{R_2\}$ to $\{G_1\}$ is:

$$\begin{aligned} _{R_2}^{G_1}\mathbf{C}(^{G_1}\phi_{R_2}) &= {}_{R_1}^{G_1}\mathbf{C}(\phi_1) \, _{R_2}^{R_1}\mathbf{C}(\theta) \\ \Rightarrow {}^{G_1}\phi_{R_2} &= \phi_1 + \theta \end{aligned}$$

Substituting for θ from Eq. (5), we have:

$$^{G_1}\phi_{R_2} = \phi_1 + \pi + {}^1\theta_2 - {}^2\theta_1 \qquad (10)$$

where, ϕ_1 is the orientation of robot R_1, and $^j\theta_i$ is the relative bearing measurement of R_j towards R_i.

From Fig. 2, we obtain the following geometric relations:

$$^{G_1}\mathbf{p}_{G_2} = {}^{G_1}\mathbf{p}_{R_1} + {}_{R_1}^{G_1}\mathbf{C}(\phi_1)^{R_1}\mathbf{p}_{R_2} - {}_{G_2}^{G_1}\mathbf{C}(\phi)^{G_2}\mathbf{p}_{R_2} \quad (11)$$

$$^{G_1}\mathbf{p}_{R_2} = {}^{G_1}\mathbf{p}_{R_1} + {}_{R_1}^{G_1}\mathbf{C}(\phi_1)^{R_1}\mathbf{p}_{R_2} \qquad (12)$$

where $^{G_1}\mathbf{p}_{R_1}$ is the position of robot R_1.

As shown in Fig. 2, the position of each of the landmarks $\ell_i \in M_2$ expressed with respect to the global frame $\{G_1\}$ is:

$$^{G_1}\mathbf{p}_{\ell_i} = {}^{G_1}\mathbf{p}_{G_2} + {}_{G_2}^{G_1}\mathbf{C}(\phi) \, ^{G_2}\mathbf{p}_{\ell_i}, \; i = 1 \dots n_2 \quad (13)$$

where ϕ and $^{G_1}\mathbf{p}_{G_2}$ are given by Eqs. (9) and (11), and $^{G_2}\mathbf{p}_{\ell_i}$ is the position of landmark ℓ_i in the map M_2.

The transformation of the state vector $^{G_2}\mathbf{X}_2$, estimated by robot R_2, to $^{G_1}\mathbf{X}_2$ (cf. Eq. (6)) is described by $^{G_1}\mathbf{p}_{R_2}$, $^{G_1}\phi_{R_2}$, and $^{G_1}\mathbf{p}_{\ell_i}$ (Eqs. (12), (10), and (13)). Since the true quantities are not known, measured and estimated quantities are used in this transformation. There relations require (i) an estimate of robot R_1's pose, (ii) an estimate of robot R_2's pose and landmarks' positions in M_2, (iii) the robot-to-robot distance and bearing measurements. The accuracy of the transformation between the two coordinate frames will depend on that of the state estimate in (i) and (ii) and the relative measurements in (iii). In the following section, we show the relations that describe how errors in each of these quantities affect the accuracy of the aligned map.

C. Error Transformation

The transformations described in Section III-B are nonlinear. In order to compute the error in the transformed quantities, we linearize these equations at the estimated quantities. The Jacobians of $^{R_1}\mathbf{p}_{R_2}$, $^{G_1}\phi_{R_2}$, and $^{G_1}\mathbf{p}_{\ell_i}$ (Eqs. (12), (10), (13)) are computed with respect to the augmented state vector $\mathbf{X} = [^{G_1}\mathbf{X}_1^T \; ^{G_2}\mathbf{X}_2^T]^T$ (cf. Eq. (7)) and the measurement vector $\mathbf{Z}$ (cf. Eq. (2)). The errors in $^{R_1}\mathbf{p}_{R_2}$, $^{G_1}\phi_{R_2}$, and $^{G_1}\mathbf{p}_{\ell_i}$, expressed with respect to the global frame $\{G_1\}$, are:

$$^{G_1}\tilde{\mathbf{p}}_{R_2} = \begin{bmatrix} ^{p_{R_2}}\mathbf{H}_1 & {}^{p_{R_2}}\mathbf{H}_2 \end{bmatrix} \begin{bmatrix} ^{G_1}\tilde{\mathbf{X}}_1 \\ \hdashline ^{G_2}\tilde{\mathbf{X}}_2 \end{bmatrix} + {}^{p_{R_2}}\boldsymbol{\Gamma}\,\boldsymbol{\eta} \quad (14)$$

$$^{G_1}\tilde{\phi}_{R_2} = \begin{bmatrix} ^{\phi_{R_2}}\mathbf{H}_1 & {}^{\phi_{R_2}}\mathbf{H}_2 \end{bmatrix} \begin{bmatrix} ^{G_1}\tilde{\mathbf{X}}_1 \\ \hdashline ^{G_2}\tilde{\mathbf{X}}_2 \end{bmatrix} + {}^{\phi_{R_2}}\boldsymbol{\Gamma}\,\boldsymbol{\eta} \quad (15)$$

$$^{G_1}\tilde{\mathbf{p}}_{\ell_i} = \begin{bmatrix} ^{\ell_i}\mathbf{H}_1 & {}^{\ell_i}\mathbf{H}_2 \end{bmatrix} \begin{bmatrix} ^{G_1}\tilde{\mathbf{X}}_1 \\ \hdashline ^{G_2}\tilde{\mathbf{X}}_2 \end{bmatrix} + {}^{\ell_i}\boldsymbol{\Gamma}\,\boldsymbol{\eta} \quad (16)$$

where $i = 1 \dots n_2$, $^{p_{R_2}}\mathbf{H}_1$, $^{p_{R_2}}\mathbf{H}_2$, $^{p_{R_2}}\boldsymbol{\Gamma}$, $^{\phi_{R_2}}\mathbf{H}_1$ $^{\phi_{R_2}}\mathbf{H}_2$, $^{\phi_{R_2}}\boldsymbol{\Gamma}$, $^{\ell_i}\mathbf{H}_1$, $^{\ell_i}\mathbf{H}_2$, $^{\ell_i}\boldsymbol{\Gamma}$ are the corresponding Jacobians which can be found in [5].

D. Transforming the Map

Employing the results of the previous sections, the pose and the feature map of robot R_2 can now be described with respect to frame $\{G_1\}$. Before the two robots meet, two independent Kalman filter estimators, one for each robot, were used for estimating the vectors $^{G_i}\mathbf{X}_i$, with covariance $\mathbf{P}_{ii} = E[^{G_i}\tilde{\mathbf{X}}_i {}^{G_i}\tilde{\mathbf{X}}_i^T]$, $i = 1, 2$. Since $^{G_1}\mathbf{X}_1$ and $^{G_2}\mathbf{X}_2$ are initially independent, the covariance of the augmented state vector is block diagonal.

$$\mathbf{P} = E\left[\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T\right] = \begin{bmatrix} \mathbf{P}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{P}_{22} \end{bmatrix}$$

In order to align the two maps, we express $^{G_2}\mathbf{X}_2$ in the same frame as $^{G_1}\mathbf{X}_1$ (Eq. (6) or equivalently Eqs. (12), (10), and (13)). The new augmented state vector is:

$$\mathbf{X}' = \begin{bmatrix} ^{G_1}\mathbf{X}_1^T & ^{G_1}\mathbf{X}_2^T \end{bmatrix}^T \tag{17}$$

while the errors in that are described as:

$$\begin{aligned} \tilde{\mathbf{X}}' &= \mathbf{H}\tilde{\mathbf{X}} + \mathbf{\Gamma}\boldsymbol{\eta} \\ &= \begin{bmatrix} \mathbf{I}_{m\times m} & \mathbf{0}_{m\times n} \\ \mathbf{H}_1 & \mathbf{H}_2 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{X}}_1 \\ \tilde{\mathbf{X}}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{m\times 3} \\ \mathbf{\Gamma}_2 \end{bmatrix} \boldsymbol{\eta} \end{aligned} \tag{18}$$

where $m = 3 + 2n_1$, $n = 3 + 2n_2$, and

$$\mathbf{H}_1 = \begin{bmatrix} ^{p_{R_2}}\mathbf{H}_1 \\ ^{\phi_{R_2}}\mathbf{H}_1 \\ ^{p_{\ell_1}}\mathbf{H}_1 \\ \vdots \\ ^{p_{\ell_{n_2}}}\mathbf{H}_1 \end{bmatrix}, \quad \mathbf{H}_2 = \begin{bmatrix} ^{p_{R_2}}\mathbf{H}_2 \\ ^{\phi_{R_2}}\mathbf{H}_2 \\ ^{p_{\ell_1}}\mathbf{H}_2 \\ \vdots \\ ^{p_{\ell_{n_2}}}\mathbf{H}_2 \end{bmatrix}, \quad \mathbf{\Gamma}_2 = \begin{bmatrix} ^{R_2}\mathbf{\Gamma} \\ ^{\phi_{R_2}}\mathbf{\Gamma} \\ ^{\ell_1}\mathbf{\Gamma} \\ \vdots \\ ^{\ell_{n_2}}\mathbf{\Gamma} \end{bmatrix}$$

Finally, the covariance of the new system is computed as:

$$\begin{aligned} \mathbf{P}' &= E\left[\tilde{\mathbf{X}}'\tilde{\mathbf{X}}'^T\right] = \mathbf{H}\mathbf{P}\mathbf{H}^T + \mathbf{\Gamma}\mathbf{R}\mathbf{\Gamma}^T \\ &= \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{11}\mathbf{H}_1^T \\ \mathbf{H}_1\mathbf{P}_{11} & \mathbf{H}_1\mathbf{P}_{11}\mathbf{H}_1^T + \mathbf{H}_2\mathbf{P}_{22}\mathbf{H}_2^T + \mathbf{\Gamma}_2\mathbf{R}\mathbf{\Gamma}_2^T \end{bmatrix} \end{aligned} \tag{19}$$

This new merged map can now be used by the two robots for C-SLAM.

E. Detecting and Combining Duplicate Landmarks (Sequential Nearest Neighbor Test)

In this section, we describe a method for determining duplicate landmarks and use the landmark matches to update the *merged* map. It is very probable that the areas the two robots covered have common regions before rendezvous. This means that a number of landmarks might appear as duplicates in the new state vector that resulted by aligning and merging the state estimates of the two robots. Employing this information can significantly improve the accuracy of the final map. Specifically, if all duplicate landmark pairings are known, these can be used to reduce both the alignment errors and the size of the state vector. We hereafter consider the most challenging case, where none of the landmarks can be uniquely identified.

A straightforward approach for finding consistent matches is simply to search through all possible feature-pairing combinations. Without loss of generality, we assume that $n_1 \geq n_2$,

and consider the case where each of the landmarks of map M_2 can be matched to *only* one landmark of map M_1 or to the *null* landmark (i.e., no match). That is, there are $n_1 + 2 - k$ possible matches for the k-th landmark of M_2, $k = 1 \ldots n_2$ which leads to a total of $(n_1 + 1)!/(n_1 - n_2 + 1)!$ feature matching hypotheses.[3] For even modest values of n_1 and considering the most common case where n_1 and n_2 are of the same order of magnitude, this is a prohibitively large number of candidate matchings that cannot be computed in real-time.

Two well known methods for searching for identical landmarks are the Nearest Neighbor (NN) and Joint Compatibility Branch and Bound (JCBB) algorithms [6]. NN simply pairs only the closest two landmarks. If for each landmark in M_2, we search through all landmarks in M_1 for its nearest neighbor, then the total complexity is $O(n_1 n_2)$. JCBB finds the largest number of jointly compatible pairings by searching the interpretation tree [7] with branch and bound heuristics. The algorithm is more robust but has a much higher computational cost. The JCBB algorithm is employed in [8] in the context of single robot SLAM. Specifically, a set of local maps is constructed in order to map a large indoor environment using sonar data. Once a new sub-map is joined with the global map, duplicate landmarks are matched and fused to update the new global map. Although the problem considered in [8] is different from ours (single vs. multi robot SLAM), the same approach can be applied in our case.

Considering the trade-off between robustness and complexity, we have selected to use a modified version of the NN algorithm. NN was deemed sufficient for the purposes of the work presented here, because the positioning uncertainty of the landmarks in the vicinity of the rendezvous location, expressed with respect to frame $\{R_1\}$, is relatively small. These landmarks, i.e., the ones located close to the area where the two robots meet, have a higher probability of being within the intersection of the two maps and are the ones that can be reliably matched within the first few iterations of our algorithm.

1) Mahalanobis Distance Hypothesis Test: At this point, we present the details of our approach for testing for duplicate landmarks and updating the map. After the map transformation, the new state vector is $\mathbf{X}' = \begin{bmatrix} ^{G_1}\mathbf{X}_{R_1}^T & ^{G_1}\mathbf{X}_{L_1}^T & \cdots & ^{G_1}\mathbf{X}_{L_{n_1}}^T & ^{G_1}\mathbf{X}_{R_2}^T & ^{G_1}\mathbf{X}_{\ell_1}^T & \cdots & ^{G_1}\mathbf{X}_{\ell_{n_2}}^T \end{bmatrix}^T$. If L_i and ℓ_j refer to the same landmark, then the difference between the position coordinates of the two landmarks should be zero. We denote the implicit measurement as $\mathbf{Z}_{ij}$, and the corresponding estimate as $\hat{\mathbf{Z}}_{ij}$. Using this implicit measurement, we can formulate a test for duplicate landmarks based

---

[3]If we additionally allow for landmarks of map M_2 to be matched to more than one landmark of map M_1, the total number of hypotheses becomes $(n_1 + 1)^{n_2}$.

on a Mahalanobis distance.

$$
\begin{aligned}
\mathbf{Z}_{ij} &= {}^{G_1}\boldsymbol{X}_{L_i} - {}^{G_1}\boldsymbol{X}_{\ell_j} = \mathbf{0}_{2\times1} \\
\hat{\mathbf{Z}}_{ij} &= {}^{G_1}\hat{\boldsymbol{X}}_{L_i} - {}^{G_1}\hat{\boldsymbol{X}}_{\ell_j} \\
\mathbf{r}_{ij} &= \mathbf{Z}_{ij} - \hat{\mathbf{Z}}_{ij} = -\hat{\mathbf{Z}}_{ij} \\
\mathbf{H}_{ij} &= \begin{bmatrix} \mathbf{0}_{2\times3} & \mathbf{0}_{2\times2(i-1)} & \mathbf{I}_{2\times2} & \mathbf{0}_{2\times2(n_1-i)} \\ \mathbf{0}_{2\times3} & \mathbf{0}_{2\times2(j-1)} & -\mathbf{I}_{2\times2} & \mathbf{0}_{2\times(n_2-j)} \end{bmatrix} \\
\mathbf{S}_{ij} &= \mathbf{H}_{ij}\mathbf{P}^{'}\mathbf{H}_{ij}^{T} \\
D_{ij} &= \mathbf{r}_{ij}^{T}\mathbf{S}_{ij}^{-1}\mathbf{r}_{ij} \ , \ \ i=1\ldots n_1 \ , \ \ j=1\ldots n_2
\end{aligned}
$$

where D_{ij} is the Mahalanobis distance.

If $\min_j(D_{ij})$ is less than a threshold γ, then we consider that L_i and ℓ_j represent the same landmark. In this case, a constraint is imposed via a Kalman filter update using the implicit measurement of $\mathbf{Z}_{ij}$. The remaining equations of the Kalman filter are:

$$
\mathbf{X}^{'}_{+} = \mathbf{X}^{'} + \mathbf{K}_{ij}\mathbf{r}_{ij}, \qquad \mathbf{P}^{'}_{+} = \mathbf{P}^{'} - \mathbf{K}_{ij}\mathbf{S}_{ij}\mathbf{K}_{ij}^{T} \quad (20)
$$

with $\mathbf{K}_{ij} = \mathbf{P}^{'}\mathbf{H}_{ij}^{T}\mathbf{S}_{ij}^{-1}$. After this update, the duplicate landmark ℓ_j is eliminated, i.e., the corresponding row of the state vector $\mathbf{X}^{'}_{+}$, and row and column of the covariance matrix $\mathbf{P}^{'}_{+}$ are deleted. This process of searching for duplicates is repeated for all landmarks ℓ_j originally in the map M_2.

2) Matching Robustness - Effect of the Landmark Distance from the Rendezvous Point: The NN match works well only when all landmarks' position errors are smaller than the distance between any two landmarks. In this case, it is unlikely to have a false match. However, this is not to be expected when merging two large maps. Specifically, the error in the rotation ${}^{R_1}_{R_2}\mathbf{C}(\theta)$ will amplify a landmark's position error with the distance from the landmark to robot R_2. A simple way to verify this is to consider the position error ${}^{R_1}\tilde{\mathbf{p}}_{\ell_j}$ induced in the position estimate of a landmark ${}^{R_1}\mathbf{p}_{\ell_j}$ due to the orientation alignment error $\tilde{\theta}$. It is

$$
\begin{aligned}
{}^{R_1}\mathbf{p}_{\ell_j} &= {}^{R_1}\mathbf{p}_{R_2} + {}^{R_1}_{R_2}\mathbf{C}(\theta)^{R_2}\mathbf{p}_{\ell_j} \Rightarrow \\
{}^{R_1}\tilde{\mathbf{p}}_{\ell_j} &= \tilde{\mathbf{v}} + \mathbf{J}^{R_1}_{R_2}\mathbf{C}(\theta)^{R_2}\mathbf{p}_{\ell_j}\tilde{\theta} \ , \ \ \mathbf{J} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}
\end{aligned}
$$

where $\tilde{\mathbf{v}}$ is the error term due to errors in the estimates of ${}^{R_1}\mathbf{p}_{R_2}$ and ${}^{R_2}\mathbf{p}_{\ell_j}$. Computing the trace of the covariance matrix for the error in ${}^{R_1}\mathbf{p}_{\ell_j}$, we have

$$
\mathrm{tr}(E[{}^{R_1}\tilde{\mathbf{p}}_{\ell_j}{}^{R_1}\tilde{\mathbf{p}}_{\ell_j}^{T}]) = \mathrm{tr}(E[\tilde{\mathbf{v}}\tilde{\mathbf{v}}^{T}]) + \sigma_{\theta}^{2}{}^{R_2}\mathbf{p}_{\ell_j}^{T}{}^{R_2}\mathbf{p}_{\ell_j} \quad (21)
$$

where σ_{θ}^{2} is the variance of the map orientation alignment error $\tilde{\theta}$. As evident, for comparable values of $\mathrm{tr}(E[\tilde{\mathbf{v}}\tilde{\mathbf{v}}^{T}])$, the uncertainty in the position of a landmark $\ell_j \in M_2$ increases with its distance from the point the two robots meet.

In the NN algorithm, the decision for matching two landmarks from the two robots' maps is irrevocable. Thus, any false matching that occurs during the elimination process, will cause the merged map to be inconsistent, making the order in which the landmarks are matched crucial. Based on the preceding analysis, the best order is to consider potential matches starting from the current location of robot R_2. Since,

in most cases, the error in the landmarks' position estimates, with respect to R_2, will be significantly smaller in the vicinity of R_2, correct matches are most likely to be found there. As new matches are confirmed and the state vector is updated sequentially, the position errors of landmarks further away will also be reduced. Thereafter, it is also more likely to find correct matches in distant areas by employing a sequential update procedure.

3) Computational Considerations: In order to reduce the computational complexity, we use a kd-tree [9], [10] to efficiently search for the nearest landmark L_i to a given landmark ℓ_j. The landmarks originally in M_1 are used to build a two-dimensional kd-tree. A kd-tree is a binary tree that partitions k dimensional points into enclosing bounding hyperrectangles. In addition to the landmark coordinate $\mathbf{p}_{L_i}$, each node in the tree contains two pointers, which are either null or point to a child node in the kd-tree, and a discriminator, d, which takes integer values in $[0, \ k-1]$. The splitting hyperplane is a plane which passes through the point $\mathbf{p}_{L_i}$ in the node and perpendicular to the direction specified by the discriminator. Let d be the value of the discriminator. Then a point belongs to the left subtree of $\mathbf{p}_{L_i}$ if and only if its d-th component is less than the d-th component of $\mathbf{p}_{L_i}$, otherwise it belongs to the right subtree. Fig. 3 provides an example of a two-dimensional kd-tree.

Searching for the nearest neighbor of a given landmark ℓ_j in the kd-tree is depth first. It first finds the leaf node that contains ℓ_j. This is done in a similar manner as in a binary search tree. However, the landmark L_i in the leaf node is not necessarily the nearest neighbor. The distance r between the two landmarks is computed. The nearest neighbor can only lie within the hypersphere centered at ℓ_j with a radius r. Then it backs up to search only the nodes that have intersection with the hypersphere. In general, during a nearest-neighbor search only a few nodes need to be inspected. The details of this searching algorithm can be found in [9].

The expected complexity of searching for the nearest $L_i \in M_1$ to each landmark $\ell_j \in M_2$ is $O(\log n_1)$, where n_1 is the total number of landmarks in M_1. Since all landmarks in M_2 need to be tested against the ones in M_1, this process will require a total of $O(n_2 \log n_1)$ operations. Building the kd-tree has complexity $O(n_1 \log n_1)$. After each EKF update, the kd-tree needs to be rebuilt. This results in $O(n_2 \log n_1 + m n_1 \log n_1)$ operations, where m is the number of duplicate landmarks identified. Since in most cases $m \ll n_2$, using kd-tree search is significantly cheaper than $O(n_1 n_2)$. Algorithm 1 summarizes the steps to detect and combine duplicate landmarks. The whole process of C-SLAM is outlined in Algorithm 2.

IV. EXPERIMENTAL RESULTS

We have experimentally tested the multi-robot SLAM and map alignment approach described in this paper within an area of size $\sim 4,800$ m^2. The experiments took place in the main building of our department. Two Pioneer 3-DX robots were used in this experiment. Each robot carried a SICK

Algorithm 1 Detecting and Combining Duplicate Landmarks

1: Sort landmarks $\ell_j \in M_2$ in *ascending* order by their distance to current position of R_2.
2: Build kd-tree of landmarks in $L_i \in M_1$
3: **for** each landmark $\ell_j \in M_2$ the sorted order **do**
4:     Search the nearest neighbor $L_i \in M_1$ using the kd-tree
5:     Compute the Mahalanobios distance D_{ij}
6:     **if** $D_{ij} < \gamma$ **then**
7:         Perform a KF update [Eq. (20)]
8:         Rebuild kd-tree
9:         Remove duplicate landmark from M_2
10:     **end if**
11: **end for**

Algorithm 2 Multi-robot SLAM with Rendezvous

1: N robots perform SLAM (individually)
2: **while** 1 **do**
3:     **if** any two robots meet and their maps are not merged before **then**
4:         Each robot takes a relative distance and bearing measurement
5:         Determine the transformation between the two robots' maps [Eqs. (9), (11)]
6:         Transform robot R_2's state to R_1's global frame [Eqs. (12), (10), (13)]
7:         Compute new state vector [Eq. (17)]
8:         Compute new covariance matrix [Eq. (19)]
9:         Detect and combine duplicate landmarks using Algorithm 1
10:     **end if**
11:     Robots with merged maps perform C-SLAM
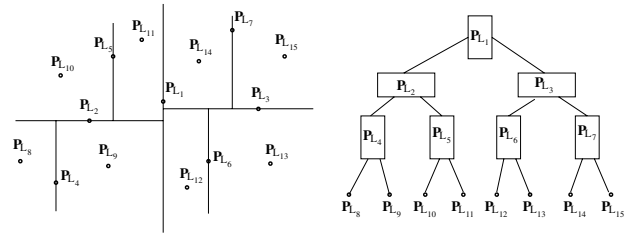12:   Robots without merged maps perform SLAM
13: **end while**

Fig. 3. An example of a kd-tree. Tall rectangles correspond to the splitting hyperplane perpendicular to the x axis. Fat rectangles correspond to the splitting hyperplane perpendicular to the y axis.

LMS200 laser range-finder and a PointGrey color camera with a RemoteReality NetVision360 omnidirectional lens. A colored cylinder was placed on top of each omnidirectional camera (Fig. 4). From the laser range data, corner features were extracted that served as landmarks in the map. Inter-robot measurements were acquired through the omnidirectional camera by means of tracking a colored cylinder mounted on the robots. A view from the omnidirectional cameral is shown in Fig. 5.

The following algorithm is employed to compute the distance and bearing between two robots. The radius and height of the cylinder placed on top of each robot are known. The idea is to match the cylinder in image space to the real cylinder in the three-dimensional space. To do this, we employ a Least-Squares estimation algorithm. We minimize the error between the measured positions of the edge pixels and the predicted positions given a distance and bearing estimate with the camera projection model and the cylinder model. The projection model of our omnidirectional camera can be found in [11]. To start this process, we first locate the upper edge

of the cylinder in the image as shown in Fig. 6. Then, using the center pixel of the upper edge, an approximate estimate of the distance and bearing is determined. The final estimates of these measurements are computed by iteratively minimizing the error between the measured positions of the edge pixels and the predicted positions using the Least-Squares algorithm.

In this experiment, the two robots start their mapping tasks at two different corners in the building (approximately 85 m apart). The relative pose between the two robots is unknown. Initially, each robot explores the floor independently. Their maps are shown in Figs. 7 and 8. When the two robots meet, they measure their relative distance and bearing. Subsequently, the two maps are merged into a single one by the coordinate transformation computed using their relative measurements. Fig. 9 depicts the 3-σ ellipses of the landmarks' positioning uncertainty. It can be seen that as the distance of a landmark from the current location of robot R_2 increases, its position uncertainty in the new map also increases. Note that in Fig. 9, the rendezvous point is approximately at location (10, 0) m. The largest landmark positioning uncertainty (3-σ) is approximately 20 m for landmarks located at the lower right corner of the building map.

To reduce this error, landmark duplicates are identified by employing the Nearest Neighbor test algorithm described in this paper. The *merged* map is updated by imposing co-location constraints on these landmark matches. In this experiment, the number of landmarks detect by robots R_1 and R_2 is 95 and 60 respectively. On average, each landmark match using the kd-tree takes 4 msec on a Pentium 4 2.4 GHz PC. The total computation cost of our algorithm for detecting and combining duplicate landmarks is 0.4 sec. Compared to testing all pairs of landmarks in M_1 and M_2 that takes 6 sec, our algorithm is significantly more efficient. In this process 31 landmarks were identified as duplicates in the maps M_1 and M_2. The final landmark locations are depicted in Fig. 10. Note that the 3-σ ellipses of the landmark positions are now reduced to less than 1 m. Comparing the resulting map to the actual floor plan, we observed that all landmarks were matched correctly. Once the merging process is complete, the two robots continue to perform SLAM now as a single system in a centralized fashion (C-SLAM). The final map of the entire floor is shown in Fig. 11. This experiment demonstrates that our algorithm can successfully align two maps created by two
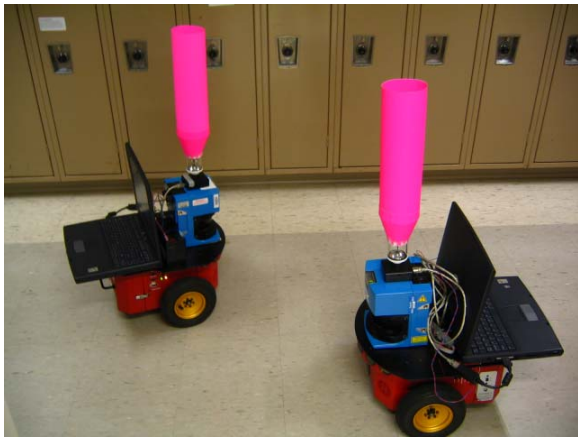
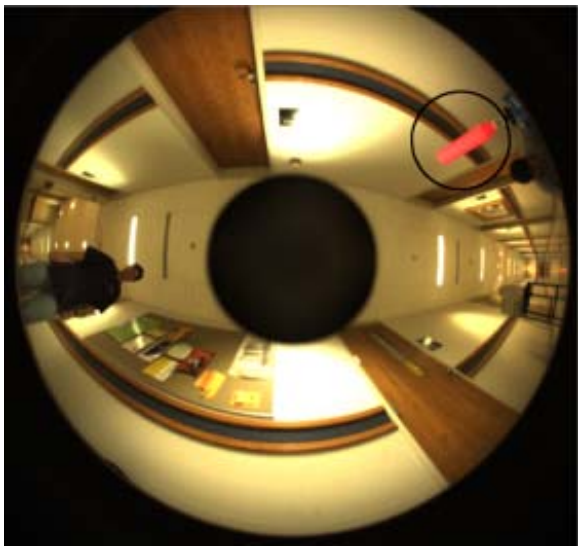Fig. 4. The Two Pioneer 3-DX robots used in the experiment.

Fig. 6. A zoom-in view of the cylinder as seen in the omnidirectional camera.

Fig. 5. A view of the second robot as seen in the omnidirectional camera of the first one.
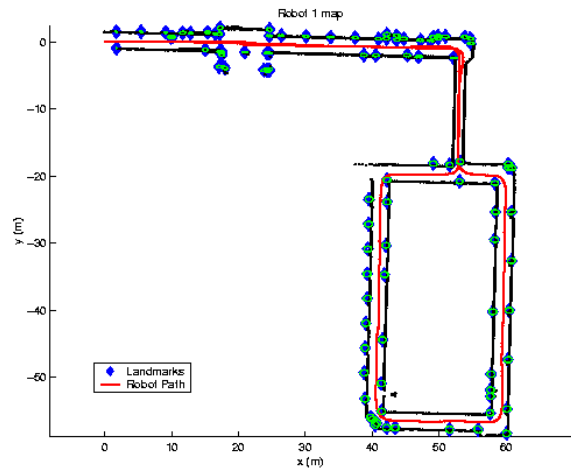
Fig. 7. The map created by robot 1 before map merging.

robots independently, correctly detect and combine duplicate landmarks, and, hence, improve the accuracy of the mapping task.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented an efficient algorithm for solving the multi-robot *map alignment* problem. Relative, robot-to-robot, *distance* and *bearing* measurements are used to compute the coordinate transformation between two maps created independently by two robots. Subsequently, *landmark duplicates* are identified and *co-location constraints* are imposed on the matched landmarks which significantly improve the accuracy of the resulting map. A Nearest Neighbor (NN) test is used in this process based on a kd-tree that reduces the *computational complexity* of merging two maps of approximately equal sizes ($n_1 \simeq n_2 \simeq n$) from $O(n^2)$ to $O(n \log n)$. In order to increase the *robustness* of the matching process, an analysis of the position error of the landmarks as a function
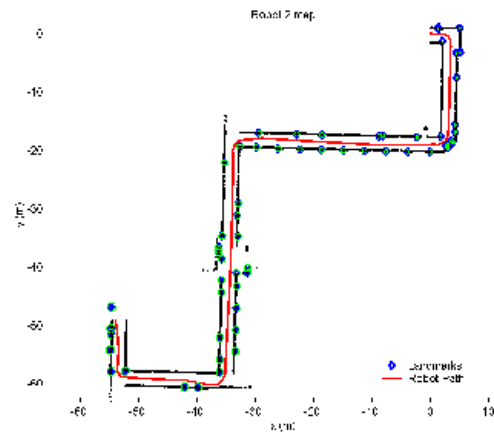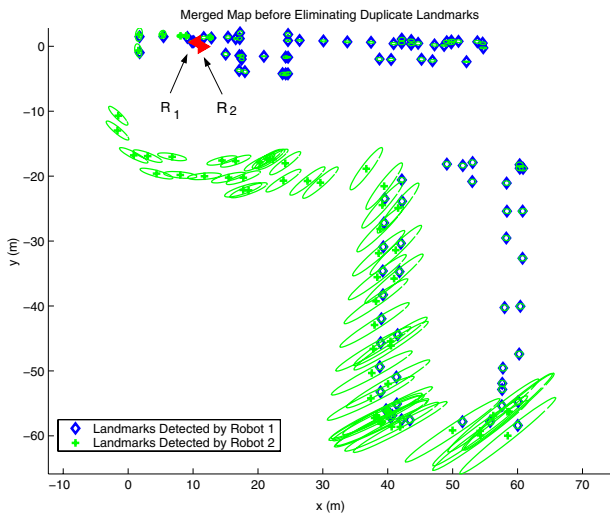

Fig. 8. The map created by robot 2 before map merging.

Fig. 9. The approximate map computed when using only the translation and rotation measured from the mutual observation of the two robots. The 3-σ position uncertainty ellipses are also shown.
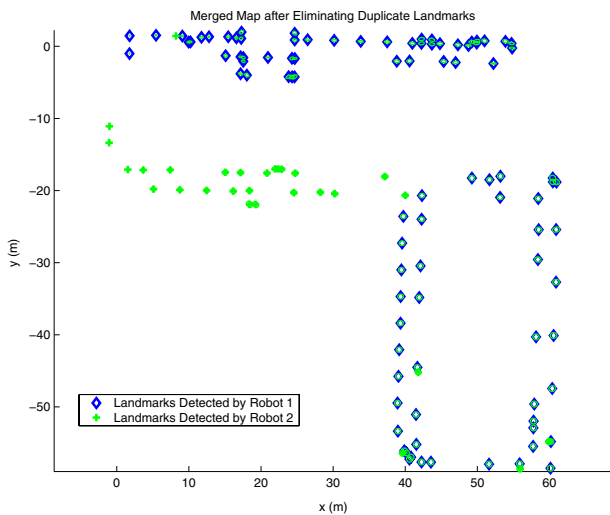

Fig. 11. The expanded map computed after the merging and while the two robots continue to perform SLAM and explore new areas.

Fig. 10. The final merged map after matching identical landmarks.

of the distance from the rendezvous point was presented. It was shown that the position error for the landmarks whose coordinates were transformed increases with their distance to the robot-meeting location. This property is employed for determining the *order* in which a NN-based algorithm should search to find reliable landmark matches in the aligned map.

Our algorithm makes no assumptions regarding the initial robot poses, thus increasing flexibility in the robots' deployment. New robots can join the mapping task at any time and/or any place in the environment of interest. Moreover, this algorithm can be applied to any feature-based map both indoors and outdoors. Whereas the experiments presented in this paper were carried out using only two robots, we are currently working on 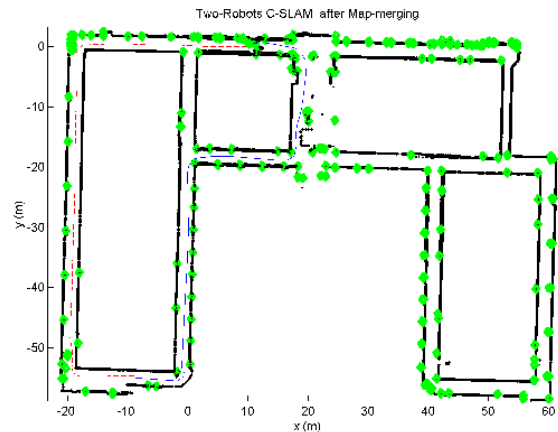C-SLAM implementations for larger robot teams. A future extension of this work will be to focus on reliable map-merging without rendezvous.

REFERENCES

[1] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," *Autonomous Robot Vehicles*, pp. 167–193, 1990.
[2] K. Konolige, D. Fox, B. Limketkai, J. Ko, and B. Stewart, "Map merging for distributed robot navigation," in *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, Nevada, Oct. 27-31 2003, pp. 212–217.
[3] A. Howard, "Multi-robot mapping using manifold representations," in *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, New Orleans, LA, Apr. 26 - May. 1 2004, pp. 4198–4203.
[4] N. Roy and G. Dudek, "Collaborative robot exploration and rendezvous: Algorithms, performance bounds and observations," *Autonomous Robots*, vol. 11, pp. 117–136, 2001.
[5] X. S. Zhou and S. I. Roumeliotis, "Multi robot SLAM map alignment with rendezvous," Department of Computer Science and Engineering, Unversity of Minnesota, Tech. Rep., 2005. [Online]. Available: www.cs.umn.edu/~zhou/paper/rendTechRep.pdf
[6] J. Neira and J. D. Tardós, "Data association in stochastic mapping using the joint compatibility test," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, December 2001.
[7] W. E. L. Grimson, *Object recognition by computer : the role of geometric constraints*. Cambridge, Mass. : MIT Press, 1990.
[8] J. D. Tardós, J. Neira, P. M. Newman, and J. J. Leonard, "Robust mapping and localization in indoor environments using sonar data," *The International Journal of Robotics Research*, vol. 21, no. 4, April 2002.
[9] A. Moore, "An introductory tutorial on kd-trees," Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. No. 209, Computer Laboratory, University of Cambridge, 1991.
[10] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, September 1975.
[11] S. K. Nayar, "Omnidirectional video camera," in *Proceedings of DARPA Image Understanding Workshop*, New Orleans, LA, May. 11-14 1997.