Vision-Based Coordinated Localization for Mobile Sensor Networks

Junghun Suh, *Student Member, IEEE*, Seungil You, *Student Member, IEEE*, Sungjoon Choi, *Student Member, IEEE*, and Songhwai Oh, *Member, IEEE*

***Abstract*—In this paper, we propose a coordinated localization algorithm for mobile sensor networks with camera sensors to operate under Global Positioning System (GPS) denied areas or indoor environments. Mobile robots are partitioned into two groups. One group moves within the field of views of remaining stationary robots. The moving robots are tracked by stationary robots and their trajectories are used as spatiotemporal features. From these spatiotemporal features, relative poses of robots are computed using multiview geometry and a group of robots is localized with respect to the reference coordinate based on the proposed multirobot localization. Once poses of all robots are recovered, a group of robots moves from one location to another while maintaining the formation of robots for coordinated localization under the proposed multirobot navigation strategy. By taking the advantage of a multiagent system, we can reliably localize robots over time as they perform a group task. In experiment, we demonstrate that the proposed method consistently achieves a localization error rate of 0.37% or less for trajectories of length between $715\ \mathrm{cm}$ and $890\ \mathrm{cm}$ using an inexpensive off-the-shelf robotic platform.**

***Note to Practitioners*—In order for a team of robots to operate indoor, we need to provide precise location information. However, indoor localization is a challenging problem and there is no available solution with an accuracy required by a low-cost robot, which lacks a capability of simultaneous localization and mapping. In this paper, we demonstrate that precise localization is possible for a multirobot system using inexpensive camera sensors and the proposed method is suitable for an inexpensive off-the-shelf robotic platform.**

***Index Terms*—Localization, mobile sensor network, multirobot coordination.**

Manuscript received July 28, 2014; accepted August 30, 2014. Date of publication November 12, 2014; date of current version April 05, 2016. This paper was recommended for publication by Associate Editor W. Shen and Editor S. Sarma upon evaluation of the reviewers' comments. This work was supported in part by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant NRF-2013R1A1A2009348, and in part by the ICT R&D Program of MSIP/IITP (Resilient Cyber-Physical Systems Research, Grant 14-824-09-013). This paper was presented at the IEEE International Conference on Automation Science and Engineering, Seoul, Korea, August 2012.

J. Suh, S. Choi, and S. Oh are with the Department of Electrical and Computer Engineering, ASRI, Seoul National University, Seoul 151-744, Korea (e-mail: junghun.suh@cpslab.snu.ac.kr; sungjoon.choi@cpslab.snu.ac.kr; songhwai.oh@cpslab.snu.ac.kr).

S. You is with the Control and Dynamic Systems, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: syou@caltech.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASE.2014.2362933

I. Introduction

A WIRELESS sensor network has been successfully applied to many areas for monitoring, event detection, and control, including environment monitoring, building comfort control, traffic control, manufacturing and plant automation, and military surveillance applications (see [2] and references therein). However, faced with the uncertain nature of the environment, stationary sensor networks are sometimes inadequate and a mobile sensing technology shows superior performance in terms of its adaptability and high-resolution sampling capability [3].

A mobile sensor network can efficiently acquire information by increasing sensing coverage both in space and time, thereby resulting in robust sensing under the dynamic and uncertain environments. While a mobile sensor network shares the same limitations of wireless sensor networks in terms of its short communication range, limited memory, and limited computational power, it can perform complex tasks, ranging from scouting and reconnaissance to environmental monitoring and surveillance by cooperating with other agents as a group. There is a growing interest in mobile sensor networks and it has received significant attention recently [4]–[8].

In order to perform sensing or coordination using mobile sensor networks, localization of all sensor nodes is of paramount importance. A number of localization algorithms have been proposed for stationary sensor networks, e.g., [9] and [10]. But they are applicable for outdoor environment and precise indoor localization is still a challenging problem [11], [12]. (For more information about various localization methods for wireless sensor networks, see references in [9]–[12].) One promising approach to indoor localization is based on the ultra-wideband (UWB) radio technology [13]. But as stated in [13], the minimum achievable positioning error can be in the order of 10 cm's and it is not accurate enough to control and coordinate a group of robots. In addition, the method requires highly accurate time synchronization. In order to address these issues, UWB based localization is combined with infrared sensors using a team of mobile agents in [14]. However, it requires the deployment of UWB detectors in advance, which is not suitable for mobile sensor networks operating under uncertain or unstructured environments.

Localization using camera sensors has been widely studied in the computer vision community. Taylor *et al.* [15] used controllable light sources to localize sensor nodes in a stationary camera network. A distributed version of camera localization is proposed by Funiak *et al.* [16], in which relative positions of cameras are recovered by tracking a moving object. The sensor placement scheme is presented for the problem of minimizing the localization uncertainty in [17]. They proposed a triangulation-based state estimation method using bearing measurements obtained from two sensors. Meingast *et al.* [18] proposed a multitarget tracking based camera network localization algorithm. The critical concept applied in [18] is the use of spatiotemporal features, an approach taken in this paper. Tracks of moving objects are used as spatiotemporal features (tracks are detected by a multitarget tracking algorithm from [19]). In order to find matching features over a pair of cameras, detection times of spatial features are used as well as spatial features such as Harris corners and scale-invariant feature transform (SIFT) keypoints [20]. Then, the relative position and orientation between cameras are computed using multiview geometry. Since an incorrect matching between spatiotemporal features is extremely rare compared to spatial features, the method provided outstanding performance under a wide baseline and varying lighting conditions.

But the aforementioned methods are designed for stationary camera networks and are not suitable for dynamic mobile sensor networks. In fact, in mobile sensor networks, we can take the advantage of mobility to improve the efficiency of localization. For instance, Zhang *et al.* [21] proposed a method to control the formation of robots for better localization. They estimated the quality of team localization depending on the sensing graph and the shape of formation. A multirobot localization algorithm based on the particle filter method is presented in [22]. They proposed a reciprocal sampling method which selects a small number of particles when performing a localization process. Some authors have considered cooperative localization of multiple robots using bearing measurements. Giguere *et al.* [23] addressed the problem of reconstructing relative positions under the condition of mutual observations between robots. The constraint was later relaxed by adding landmarks in [24]. They used nonlinear observability analysis to derive the number of landmarks needed for full observability of the system and an extended information filter was applied to estimate the states of a team of robots using bearing-only measurements. Ahmad *et al.* [25] applied a cooperative localization approach to robot soccer games. They modeled the problem as a least squares minimization and solved the problem using a graph-based optimization method, given static landmarks at known positions. Tully *et al.* [26] used a leap-frog method for a team of three robots performing cooperative localization, which is similar to the proposed method. In [26], two stationary robots localize the third moving robot from bearing measurements using an extended Kalman filter. After completing a single move, the role of each robot is switched and the process is repeated. In their experiments, robots covered a region of size 20 m $\times$ 30 m and showed a localization error of 1.15 m for a trajectory of length approximately 140 m. However, the experiments were conducted using an expensive hardware platform including three on-board computers, four stereo cameras, and a customized ground vehicle with many sensors. Hence, it is unclear if the approach is suitable for an inexpensive off-the-shelf robotic platform considered in this paper.

We propose a coordinated localization algorithm for mobile sensor networks under Global Positional System (GPS) denied areas or indoor environments using an inexpensive off-the-shelf robotic platform. We take the advantage of mobile sensor networks. In order to localize mobile robots, we first partition robots into two groups: stationary robots and moving robots. We assume each robot carries a camera and two markers.[1] The moving robots move within the field of views (FOVs) of stationary robots. The stationary robots observe the moving robots and record the positions of markers of moving robots. Based on the trajectories of markers, i.e., spatiotemporal features, we localize all the robots using multiview geometry. Localization requires recovering relative positions, i.e., translation and orientation. While the translation between cameras can be recovered only up to a scaling factor in [18], we can recover the exact translation using the known distance between markers in the proposed algorithm.

A multirobot navigation strategy is also developed using the rapidly-exploring random tree (RRT) [27], which moves a group of robots from one location to another while maintaining the formation of robots for coordinated localization. Since the proposed localization algorithm requires a certain configuration of a robot team for good localization, the RRT algorithm is modified to guarantee the configuration condition.

We have implemented the proposed algorithm on a mobile robot platform made from an iRobot Create [28] mobile robot and conducted an extensive set of experiments. From experiments, we have discovered a set of configurations of robots, from which good localization is possible. We then applied these configurations in our coordinated multirobot localization algorithm. Our experimental results show that the proposed method consistently achieves less than 1 cm of localization error for trajectories of length less than 100 cm and a localization error rate of 0.37% or less for longer trajectories with length between 715 cm and 890 cm, making it a promising solution for multirobot localization in GPS denied or unstructured environments.

In order to compare the performance of the proposed method, we have also implemented the leap-frog method [26] using the same robotic platform used in this paper. From experiments, the leap-frog method gives a localization error rate of 5.6% for trajectories with the average length of 820.6 cm. The proposed method achieves a localization error rate which is more than 15 times better than the leap-frog method for trajectories with a similar length.

This paper is structured as follows. Section II provides an overview of the proposed coordinated multirobot localization method. In Section III, we present the multirobot localization algorithm. Multirobot navigation system is described in Section IV. We discuss experimental results in Section V and compare the proposed method to the leap-frog method in Section VI.

II. An Overview of Coordinated Multirobot Localization

Suppose there are N robots and we index each robot from 1 to N. We assume that each robot's state is determined by

[1]For robots moving on a flat surface, a single marker with a known height can be used.

its position and orientation in the reference coordinate system. Then, the goal of the multirobot localization problem is to estimate positions and orientations of all robots over time.

Let $X_i(k) = (P_i(k), R_i(k))$ be the state of robot i at time k with respect to the reference coordinate system, where $P_i(k) \in \mathbb{R}^n$ and $R_i(k) \in SO(3)$ are the position and rotation of robot i at time k, respectively.[2] Then, the configuration of a multirobot system at time k is

$$X(k) = (X_1(k), X_2(k), \ldots, X_N(k)).$$

The multirobot localization problem is to estimate $X(k)$ for all k from sensor data.

Suppose that we have $X(k-1)$ with respect to the reference coordinate system and computed relative positions, $T_{ij}(k)$, and orientations, $R_{ij}(k)$, for a pair of robots i and j at time k. Then, we can easily compute positions and orientations of all robots with respect to a single robot of choice. In order to map new positions of robots in the reference coordinate system, we require that there is at least one robot i such that $X_i(k) = X_i(k-1)$. Then, taking positions with respect to this robot, we can recover the positions and orientations of all robots at time k with respect to the reference coordinate system.

Based on this idea, we develop a coordinated localization algorithm. At each time instance, we fix robot q and move other robots. Then, we compute $T_{ij}(k)$ and $R_{ij}(k)$ for pairs of robots such that the pose of a robot can be computed with respect to robot q. Finally, we compute $X(k)$ based on $X_q(k-1)$. For $k+1$, we fix another robot r and move remaining robots and continue this process. By doing so, we can continuously estimate $X(k)$ for all times.

Now, the remaining issue is how to estimate translations $T_{ij}(k)$ and orientations $R_{ij}(k)$ for pairs of robots. For this task, we make the following assumptions.

- Each robot carries a camera and markers.
- The internal parameters of cameras are known (e.g., focal lengths, principal points, and distortion coefficients).
- Each robot communicates with other robots via wireless communication.
- The clocks of all robots are synchronized.
- Either the distance between a pair of markers on a robot is known or the height of a single marker is known when a robot is moving on a flat surface.
- At least two robots which capture images are stationary.

Robots carrying markers move within the FOVs of stationary robots. Each stationary robot captures an image, detects markers, and localizes positions of markers in its image frame at time k. The marker positions and image capture times are shared with other stationary robots. For a pair of stationary robots i and j, we can compute the relative translation $T_{ij}(k)$ and orientation $R_{ij}(k)$ from a pair of marker trajectories using multiview geometry as discussed in Section III. At time $k+1$, every stationary robot except at least one robot moves and repeats the same process. There is one remaining issue which is that we can only recover the relative positions up to a scaling factor when only images are used. To recover the absolute translation value, we need a known length. To resolve this issue, we assume that the markers on robots are placed at known heights.

Since the minimum number of robots required for the proposed coordinated localization algorithm is three, we will discuss our method using a mobile sensor network of three robots for the ease of exposition in this paper. However, the proposed method can be applied to a multirobot system with a larger number of robots. Furthermore, while a single moving robot is used in our discussion and experiment, the method can be likewise applied to the case with multiple moving robots using the multitarget tracking method of [18].

III. Multirobot Localization Using Multiview Geometry

In this section, we focus on a single step of the coordinated multirobot localization algorithm for localizing stationary robots by tracking a moving robot and present how to control the moving robot using visual data from stationary robots.

A. Planar Homography for Robot Localization

When two cameras view the same 3D scene from different viewpoints, we can construct the geometric relation between two views using the homography if the scene is planar. From the homograpy, we can recover $\{R, T, N\}$ up to a scale factor using the singular value decomposition, where $R \in SO(3)$ is the rotation matrix, $T \in \mathbb{R}^3$ is the translation vector, and N is the unit normal vector of the plane with respect to the first camera frame. From this derivation, we obtain two possible solutions [29]. In this application, corresponding points are located on the plane which is parallel to the ground and the tilted angle of each camera is fixed, so we can compute the normal vector of the plane. Among two solutions, we can find a unique solution since the normal vector which is perpendicular to the plane is available in our case. As explained in [29], from the singular value decomposition of $H^T H$, we obtain an orthogonal matrix $V \in SO(3)$, such that $H^T H = V \sum V^T$, where $V = [v_1, v_2, v_3]$. Let u be a unit-length vector such that $N = v_2 \times u$ and $v_2^T u = 0$ and v_2 is orthogonal to N. Therefore, given v_2 and N, we can solve for u. Once we find u, we can form the new orthonormal basis $\{v_2, u, N\}$ and obtain R and T as follows:

$$R = WU^T \qquad \text{and} \qquad T = d(H - R)N \tag{1}$$

where $U = [v_2, u, N]$ and $W = \left[Hv_2, Hu, \widehat{Hv_2}Hu\right]$, and $\hat{x} \in \mathbb{R}^{3\times 3}$ is a skew-symmetric matrix. When we reconstruct positions of markers in the 3D space, we can find the exact scale factor using the distance between markers.

The homography can be computed from a number of pairs of feature correspondences. We use the spatiotemporal correspondence features by tracking a moving robot in the FOVs of scenes. We first segment a marker on the robot at each time instance from image sequences of each camera. It is performed by applying the maximally stable extremal regions (MSER) detector [30], a blob detection method. A centroid of the marker is used to build a marker image track. Even though the above detection algorithm for extracting marker positions of the moving robot is robust, it may contain outliers, which do not fit the 2D

[2] $SO(3)$ is the special orthogonal group in $\mathbb{R}^3$ (the group of 3D rotations).

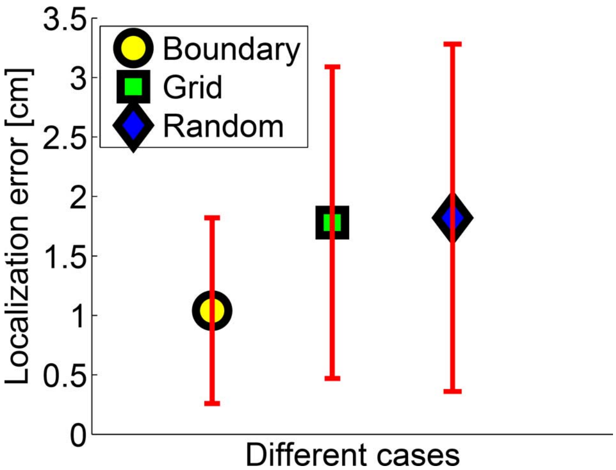
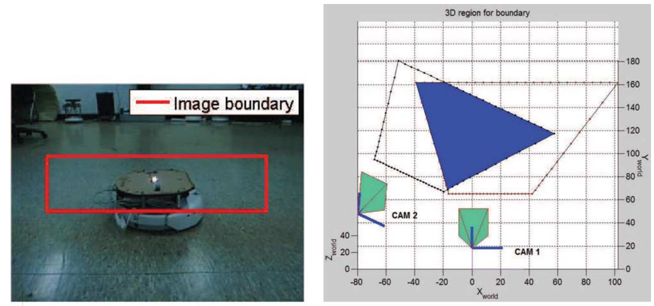


Fig. 1. Average localization errors for three scenarios. The average error is computed from 500 independent runs and the error bar shows one standard deviation from its mean value.



(a) (b)

Fig. 2. (a) An example of a boundary within an image frame. (b) The common FOV of two cameras on the ground plane.

plane, such as debris on the ground or measurement errors. In order to robustly estimate the planar homography, we used the random sampling consensus (RANSAC) algorithm [31].

B. Image-Based Robot Control

The trajectory of a moving robot can influence the quality of localization. Hence, it is desirable to move a robot such that the localization error can be minimized. Since a robot has to be controlled using data from cameras, it can be seen as a visual servo control problem. In visual servo control, namely, the image-based visual servo (IBVS) approach, the control input to the moving robot is computed based on the error generated in the 2D image space [32]. However, in order to compute the control input, visual servoing requires the pseudo inverse of an interaction matrix which represents the relationship between the velocity of the moving robot and the time derivative of the error. Since the interaction matrix has six degrees of freedom, such a process requires at least three feature points at each time. Since we only consider a single marker in the present paper, the visual servoing approach is not applicable and an alternative robot controller is required.

1) Robot Trajectory Design for Better Localization: We have considered three scenarios in order to identify an ideal robot trajectory for minimizing the localization error. The considered cases are: 1) points along the boundary of the common FOV by two cameras; 2) uniformly scattered points inside the FOV; and 3) randomly placed points inside the FOV. For each case, we randomly selected 50 point pairs and performed the proposed localization algorithm. We have repeated the process for 500 times. For each run, we computed the estimation error $\epsilon_i = |\hat{d}_i - d_{true}|$, for $i = 1, 2, \ldots, 500$, where $\hat{d}_i$ is the estimated distance between two robots for the ith run using our localization method and d_{true} is the ground truth distance. Average localization errors are shown in Fig. 1. More detail about the experiment can be found from the longer version of this paper [33]. It can be seen that points along the boundary of the FOV give the lowest localization error. The simulation was conducted using MATLAB based on parameters of cameras used in experiments. A point is projected to the image plane with an additive zero-mean Gaussian noise with a standard deviation of one pixel.

2) Control: From the previous section, we have found that we can lower the localization error using feature points located at the boundary of the FOV. Based on this finding, we design an image-based robot controller which makes the moving robot follow the boundary of the common FOV of stationary robots. One difficulty is that a moving robot can move beyond the FOV due to communication delay. In order to prevent this problem, we first set the boundary within the image frame, as shown in Fig. 2(a). The common FOV of two cameras on the ground plane is shown in Fig. 2(b).

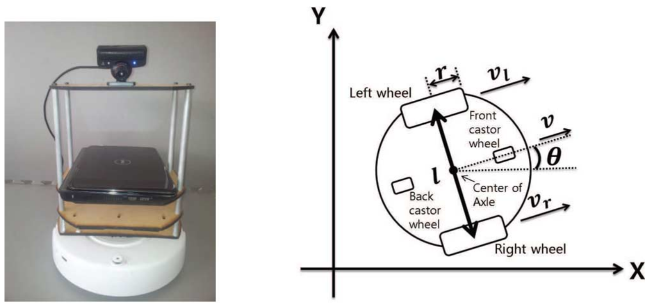
Since the actual heading of a moving robot is not available, it has to be estimated from image data. We estimate the heading direction of the moving robot using a batch least square filter over a finite window from measurements from both cameras. When the moving robot is near the boundary of a camera, the corresponding stationary robot sends a command to the moving robot to rotate by a predefined amount for a short duration. The direction of the rotation is determined by the normal vector of the boundary and the estimated heading direction. The moving robot combines commands from stationary robots and changes its heading for a short duration. While this is an extremely simple controller, we have found it very effective for controlling the robot to move along the boundary of the FOV since we cannot reliably estimate the position and heading of the moving robot using a small number of noisy marker detection results.

IV. Multirobot Navigation System

This section details how we implement the multirobot navigation system including the robot platform used in the experiments and a multirobot navigation method which moves a group of robots from one location to another, while maintaining the formation of robots for coordinated localization.

A. Multirobot System

For our experiments, we used iRobot Create wheeled mobile robots [28] as mobile nodes in our mobile sensor network. The developed mobile platform is shown in Fig. 3(a), which is equipped with a PS3 Eye camera, an ASUS notebook which runs Linux OS, and a white LED which works as a marker. WiFi (IEEE 802.11) is used for communication among robots. Each camera has a resolution of 320 $\times$ 240 pixels and runs at 40 frames per second (fps). As explained in [1], we used a



(a) (b)

Fig. 3. (a) An iRobot Create based mobile robot platform. (b) A two-wheeled differential drive robot.

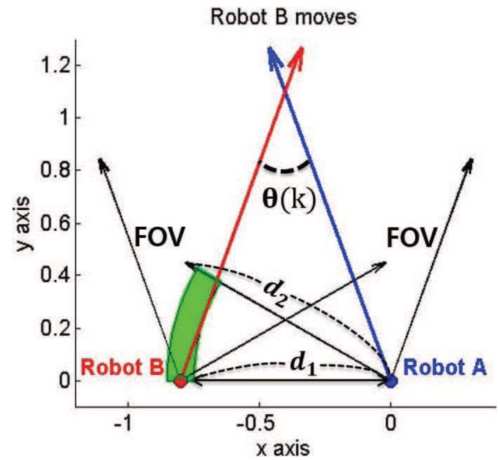


Fig. 4. Blue and red circle represent initial positions of robot A and B and blue and red arrow represent the headings of robots, respectively. Two black arrows with a gray region for each robot represent the field of view of each robot. The region with green color represents $\mathbb{X}_{\text{move}}(k)$.

one-server, two client model for communication. However, unlike [1], in this work, stationary robots consistently check the visibility of the moving robot to prevent the moving robot from going beyond the common FOV as explained in the previous section. The time synchronization operation is implemented as follows. Since each robot can have a different local time, all clocks are synchronized using the clock of the server at each time a stationary robot moves.

We used the two-wheeled differential drive robot dynamics in simulation and experiments. The parameters of the mobile robot are shown in Fig. 3(b). Let l be the distance between the two wheels and θ be its heading. The dynamics of a two-wheeled differential drive robot can be expressed as follows, where v_r and v_l are the right and left wheel velocities, respectively:

$$\begin{aligned} x(t+\Delta t) &= \\ &\begin{cases} x(t) + \frac{2v(t)}{w(t)} \sin(\tilde{w}(t)) \cos(\theta(t) + \tilde{w}(t)), & \text{if } w(t) \neq 0 \\ x(t) + v(t)\Delta t \cos(\theta(t)), & \text{otherwise} \end{cases} \\ y(t+\Delta t) &= \\ &\begin{cases} y(t) + \frac{2v(t)}{w(t)} \sin(\tilde{w}(t)) \sin(\theta(t) + \tilde{w}(t)), & \text{if } w(t) \neq 0 \\ y(t) + v(t)\Delta t \sin(\theta(t)), & \text{otherwise} \end{cases} \\ \theta(t+\Delta t) &= \theta(t) + w(t)\Delta t, \end{aligned} \tag{2}$$

where $\tilde{w}(t) = w(t)\Delta t/2$, $v(t) = v_r(t) + v_l(t)/2$ is the translational velocity, and $w(t) = v_r(t) - v_l(t)/l$ is the angular velocity [34].

We cannot directly specify v and w to reach the specific position, because we cannot compute v_r and v_l from (2). Thus, we consider only two motions by a robot (going-forward and turning) to reduce the modeling error and simplify its control. In order to correctly model the physical mobile platform used in the experiment, we have conducted a number of experiments to measure odometry errors (see [33] for more detail). Based on the experiments, we have obtained a more precise dynamic model of the mobile platform using (2). When a robot moves forward, its heading does not change, hence, the dynamics for the going-forward motion is as follows:

$$\begin{aligned} x(t+\Delta t) &= x(t) + v\cos(\theta(t))\Delta t + \alpha_1 \\ y(t+\Delta t) &= y(t) + v\sin(\theta(t))\Delta t + \alpha_1 \\ \theta(t+\Delta t) &= \theta(t) + \alpha_2, \end{aligned} \tag{3}$$

since $v_r = v_l$ and $w = 0$. The dynamics for turning becomes

$$\begin{aligned} x(t+\Delta t) &= x(t) + \alpha_3 \\ y(t+\Delta t) &= y(t) + \alpha_3 \\ \theta(t+\Delta t) &= \theta(t) + \left(\frac{2v}{l} + \alpha_4\right)\Delta t \end{aligned} \tag{4}$$

since $v_r = -v_l$ and $v = 0$ (i.e., the position of the robot is stationary). Here, $\alpha_1, \alpha_2, \alpha_3$, and α_4 are random variables representing noises including bias terms found from the experiments.

B. Multirobot Navigation

We now consider moving a group of robots from one location to another location while localizing all robots based on the proposed multirobot localization algorithm. We develop a multirobot navigation algorithm based on the rapidly exploring random tree (RRT) [27] which is a sampling-based path planning method. It quickly searches over a nonconvex configuration space by sampling a random point and incrementally builds a navigation tree by extending the tree towards the random point. While an RRT can be readily applied to a single robot, it is not straightforward to apply to a group of robots with constraints. For our multirobot localization method, robots must satisfy a requirement about the configuration of robots, namely, the distance between stationary robots and angles between them for better localization [1].

Let $\mathbf{x}_{\text{team}}(k)$ be the state of two stationary robots, i.e., $\mathbf{x}_{\text{team}}(k) = [\mathbf{x}_A(k), \mathbf{x}_B(k), \theta(k)]$, where $\mathbf{x}_A(k) \in \mathcal{X}$ and $\mathbf{x}_B(k) \in \mathcal{X}$ are states of robots A and B, respectively, at time k, and $\theta(k)$ is the angle between two stationary robots, which is graphically illustrated in Fig. 4. In order for a team of three robots to correctly localize, the following conditions must be satisfied:

$$\begin{aligned} d_1 \leq \|\mathbf{x}_A(k) - \mathbf{x}_B(k)\| &\leq d_2 \\ \theta_1 \leq \theta(k) &\leq \theta_2 \end{aligned} \tag{5}$$

where the parameters d_1, d_2, θ_1, and θ_2 are experimentally determined as discussed in Section V and fixed for the team. In

Fig. 5. Photos from the hallway experiment. A group of robots moves along the straight line (black dotted line). At each step, a robot with an LED marker moves while the remaining two robots localize based on the movement of the robot with LED using the proposed algorithm.

order for a group of robots move, one of the stationary robots has to move and there is a chance that the condition (5) can be violated. For a group of robots to navigate while localizing, the condition (5) has to be satisfied at all times.

Suppose that robot A is stationary and robot B moves forward. Then, the region satisfying the first condition of (5) can be expressed as $\mathbb{X}_{\text{move}}(k)$, the green region in Fig. 4. The second condition of (5) can be easily satisfied by rotating robot B with respect to the heading of robot A. Path planning of a team of robots satisfying the condition (5) is implemented using the RRT. However, since one robot moves and the other robot is stationary, we have to alternatively move one robot at a time while satisfying (5) at each step. The procedure is similar to scheduling steps of a humanoid robot to move from one location to a target location while avoiding obstacles. Results of RRT based path planning for a robot team are shown in Section V.

V. Experimental Results

A. Coordinated Multirobot Localization: Single-Step

Experimental results for a single step of the proposed coordinated multirobot localization algorithm is reported in [33], in which we have demonstrated that there exists a set of multirobot configurations. We have found that we can consistently achieve an error less than 1 cm when the baseline between stationary robots is between 60 cm and 80 cm and the angle between the robots is between 20 ° and 40 °. Since the baseline distance and the angle between robots can be configured in our multirobot localization algorithm for the best performance, the experimental results show how we should configure robots in our coordinated multirobot localization algorithm as we do in the next experiment.

B. Coordinated Multirobot Localization: Multistep

In this experiment, we localize a group of robots as they move from one place to another, as described in Section IV. Based on the single-step experiment, we found a configuration of multirobot system, which gave the best localization performance. Using the configuration, we performed a number of multistep experiments in a room, where the ground truth location information was collected using the Vicon motion capture system. See [33] for the experimental results.

We also conducted the going forward experiments in the hallway to demonstrate its performance over a long distance (see Fig. 5). A robot with a white LED plays the role of the moving group and two robots with a camera forms the stationary group. Table I shows localization results from the experiments in the hallway. For trajectories with length from 715 cm and 890 cm, the achieved localization error is between 1.5 cm and 3.3 cm, making the localization error rate less than 0.37% of the length of the trajectory of the robot. Furthermore, the group of robots can follow the straight line without deviating from the desired path.

TABLE I
Results From the Multistep Experiment in the Hallway

Case	Robot	True	Est.	Error	Err. Rate
1	A	730 cm	728.1 cm	1.9 cm	0.26%
2	A	732 cm	733.5 cm	1.5 cm	0.20%
3	A	715 cm	716.7 cm	1.7 cm	0.23%
4	A	890 cm	886.7 cm	3.3 cm	0.37%
5	A	857 cm	854.5 cm	2.5 cm	0.29%

Next, we tested if a group of robots can make turns to avoid obstacles. Fig. 6 shows snapshots of a multirobot system making left and right turns. We first generated a desired path with right or left turns for the multirobot system, satisfying the condition (5) in Section IV-B. Then, a multirobot system follows the given trajectory while localizing all robots. For each turn, four independent trials were performed and the results are shown in Fig. 7(a) and (b) (see [33] for more experimental results). Fig. 7(c) shows localization errors of each robot as a function of time.

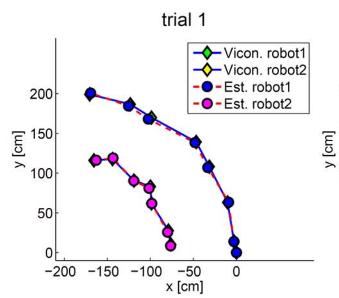
Finally, we show the results from multirobot navigation. Snapshots from the experiment are shown in Fig. 8. Given a path found by the RRT-based multirobot navigation algorithm, a multirobot system moves cooperatively while performing localization. Fig. 8(a) shows the planned path of the stationary robot found by the RRT-based multirobot path planning algorithm to navigate to the goal location. Purple and green circles represent the planned positions of robot A and B, respectively. Blue and red arrows represent their respective headings. Fig. 8(b) shows snapshots from the experiment showing the turn made by the multirobot system. Black lines are the actual trajectories of robots following the planned path. Fig. 9 shows results from obstacle avoidance experiments using the RRT-based multirobot navigation algorithm and they are taken from different overhead cameras. As shown in the figure, robots can safely navigate to reach the goal location while avoiding obstacles.

Localization under the GPS denied or unstructured indoor environment is a challenging problem. But the experimental results show that our algorithm can provide a promising solution to this challenging localization problem.

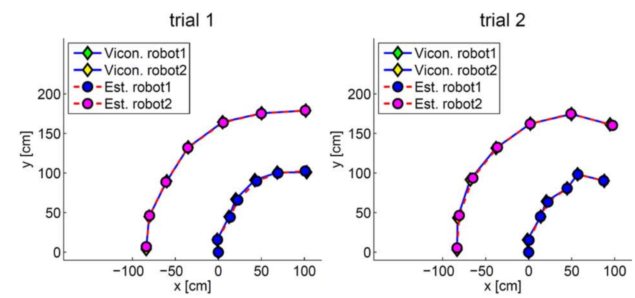
(a)

(b)

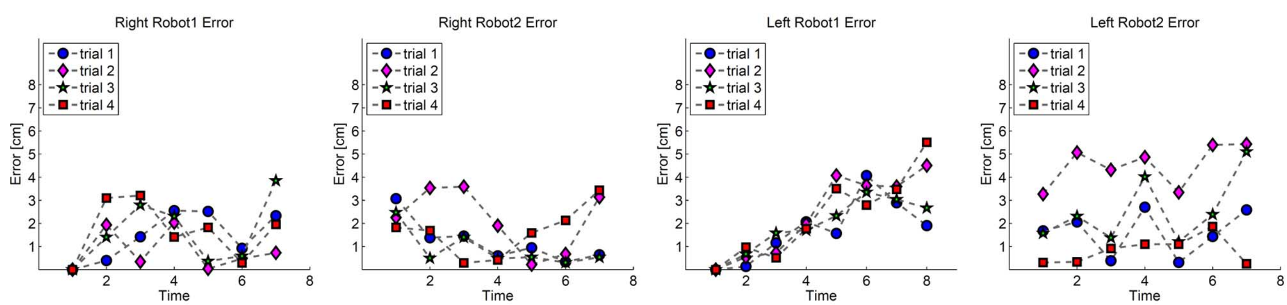
Fig. 6. Snapshots from the multirobot turning experiment. A team of robots are making left and right turns while maintaining its configuration. Black dotted arrows are the planned paths of the multirobot system, satisfying the condition (5) in Section IV-B. (a) Right turn. (b) Left turn.



(a)



(b)



(c)

Fig. 7. Estimated locations from turning experiments. Blue and magenta circles represent estimated positions of robot A and B, respectively. Green and yellow diamonds represent ground truth positions of robot A and B, respectively. (a) Left turn. (b) Right turn. (c) Localization errors as a function of time.

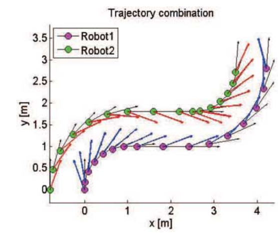
VI. Comparison to Leap-Frog [26]

We have also implemented the leap-frog localization method proposed in [26] for comparison. In its original implementation, the authors used the Learning Applied to Ground Vehicles (LAGR) platform equipped with three on-board computers, wheel encoders, and a set of four stereo cameras. However, since it is unclear if the approach is suitable for an inexpensive off-the-shelf robotic platform considered in this paper, we implemented the leap-frog algorithm using the robotic platform used in this paper, which includes PS3 Eye cameras, a white LED, and an iRobot Create platform, as shown in Fig. 10(a) and (b). We placed six cameras on the robot, as shown in Fig. 10(c), to emulate an omnidirectional camera system. Note that a PS3 Eye camera has 75° FOV. An omnidirectional camera is required for the leap-flog system since measurements for its extended Kalman filter (EKF) algorithm is relative bearing angles.

In [26], a red ball is placed on each vehicle and a circle Hough transform is used to detect the position of other vehicle. In our implementation, we used an LED as a marker for this purpose. In order to estimate the bearing angle using our omnidirectional camera system, we used the following second-order polynomial equation:

$$\theta_{\mathrm{LED}} = aw^2 + bw + c \tag{6}$$

where θ_{LED} is the relative angle of an LED with respect to the observing robot, w is the position of a detected LED in the horizontal coordinate, and a, b, and c are parameters of the polynomial. We collected 30 data pairs for each omnidirectional camera system and estimated values of a, b, and c based on the ground truth obtained from Vicon. Using estimated parameters, we found that the mean and standard deviation of the bearing angle error are 0.380° and 0.286°, respectively. The error is small enough and (6) is used to estimate the bearing angle.



(a)

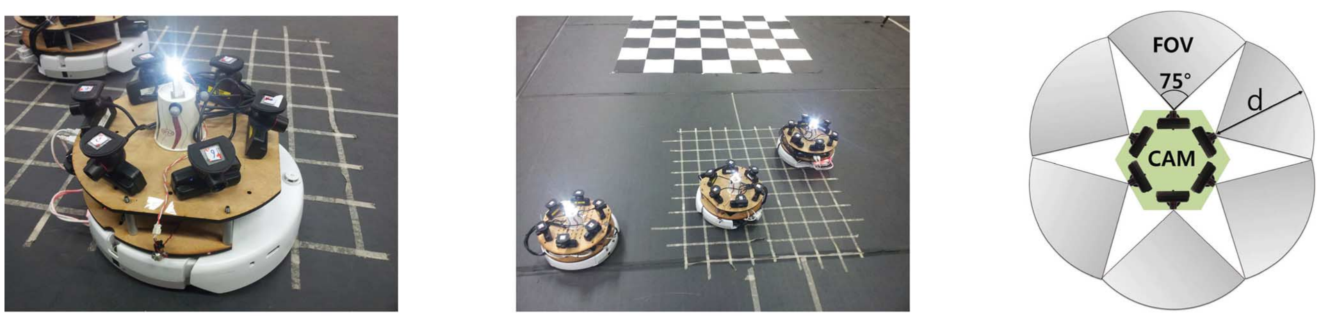
(b)

Fig. 8. (a) A trajectory found by the proposed multirobot navigation algorithm. Purple circles and blue arrows represent the planned positions of Robot A and corresponding headings, respectively. Green circles and red arrows represent the planned positions of Robot B and corresponding headings, respectively. (b) Photos from the experiment following the trajectory. Black lines show the actual trajectories of robots.

(a)

(b)

Fig. 9. Photos from the experiment when the obstacle exists. Two sequences of photos are taken by two overhead cameras (Cameras 1 and 2). In order to avoid an obstacle, a team of robots is making necessary coordinated turns. (a) Photos from Camera 1. (b) Photos from Camera 2.



(a) (b) (c)

Fig. 10. (a) A robot platform developed for leap-frog localization [26]. (b) A photo from the leap-frog localization experiment. (c) The configuration of six directional cameras for emulating an omnidirectional camera. The gray region represents the FOV of each camera placed on iRobot Create.

The key feature of the leap-frog localization is the extended Kalman filter (EKF) formulation using bearing-only measurements and multirobot formation which maximizes the information gain [26]. We acquired the bearing angle by extracting the position of the LED using the MSER detector. Before applying the MSER detector, we made a differential image between an image with an LED on and an image with an LED off, to get rid of noise. We also implemented a leap-frog formation controller by alternating go-straight and turn motions. After each movement, we performed the EKF localization using bearing-only measurements acquired from robots. Since there are blind spots, as shown in Fig. 10(c), an additional routine for handling blind spots is required. This is implemented by adding additional go-straight motions inside the leap-frog algorithm.

We first conducted a simulation to verify the localization performance of the leap-frog algorithm based on the dynamical model of robots given in Section IV-A. The moving robot follows the leap-frog path by alternating go-straight and turn motions. The initial distance between robots is 160 cm and the number of pairs of movements required to change the role of each robot is five (a movement pair consists of go-straight and turn motions). We added a Gaussian noise with variance of one to each bearing measurement. We performed a total of 100 trials in simulation. The average localization error of each robot is given in Table II. We also checked the results when we changed the number of movement pairs. As shown in Table II, the number of movement pairs does not have an effect on the results, so we applied five movement pairs in the physical experiment.

TABLE II
AVERAGE LOCALIZATION ERRORS FOR DIFFERENT STEP SIZES (METHOD: LEAP-FROG)

No. Movement Pairs	5	10	15
Robot1	$18.8\,cm$	$17.2\,cm$	$19\,cm$
Robot2	$22.9\,cm$	$21.4\,cm$	$23\,cm$
Robot3	$14.8\,cm$	$14.1\,cm$	$15.4\,cm$

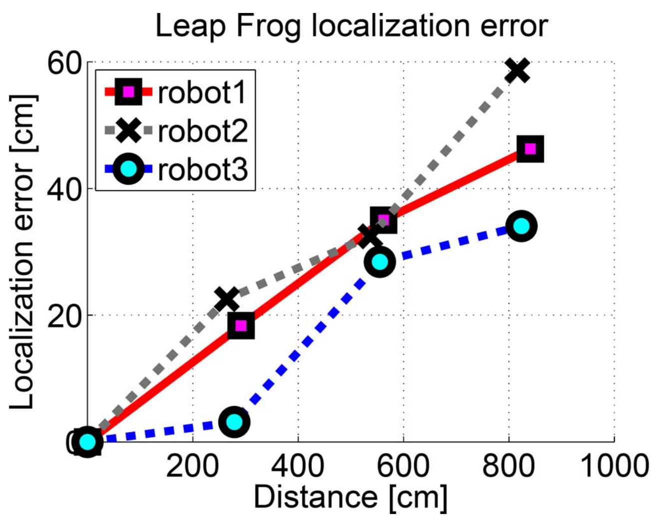


Fig. 11. Localization errors from the leap-frog localization experiment.

TABLE III
A COMPARISON BETWEEN THE PROPOSED METHOD AND THE "LEAP-FROG" METHOD

Algorithm	Mean Distance	Error	Error Rate
Leap-frog	$826.9\,cm$	$46.3\,cm$	5.6%
Proposed	$715 \sim 890\,cm$	$1.5 \sim 3.3\,cm$	$0.20\% \sim 0.37\%$

The localization results of physical experiments based on the leap-frog method using three robots are shown in Fig. 11. The localization error of robot 3 for the trajectory with length of 840 cm was 34 cm and the localization error of robot 1 and robot 2 are 46.3 cm and 58.6 cm, respectively. We can see that the localization error gets larger as each robot moves a longer distance. Note that the localization error of the actual experiment is relatively larger than that of simulation, showing the difficulty of controlling and localizing inexpensive robots. A comparison of the leap-frog method with the proposed algorithm is summarized in Table III. The proposed algorithm shows an error rate which is 15 times smaller than the leap-frog method. The result indicates that the proposed algorithm is more suitable for inexpensive robotic platforms.

VII. CONCLUSION

In this paper, we have presented a coordinated localization algorithm for mobile sensor networks. The algorithm is designed to solve the challenging localization problem under the GPS denied or unstructured indoor environment by taking the advantage of the multiagent system and mobility in mobile sensor networks. The proposed algorithm can solve the multirobot navigation problem by considering the configuration constraint of a group of robots. Our experiment shows that there exists a configuration of robots for good localization and this configuration is applied to find a trajectory which makes a group of robots move from one location to another location. We also compared the performance of the proposed algorithm against the leap-frog method using an inexpensive off-the-shelf robotic platform. In experiments, the proposed method achieves a localization error of 0.37% or less for trajectories of length between 715 cm and 890 cm. The experimental results show that the localization error increases as a robot travels a longer distance. This propagation of error can be reduced by detecting landmarks in the environment and this is our future research topic.

REFERENCES

[1] J. Suh, S. You, and S. Oh, "A cooperative localization algorithm for mobile sensor networks," in *Proc. IEEE Int. Conf. Autom. Sci. Eng. (CASE)*, Seoul, Korea, Aug. 2012, pp. 1126–1131.
[2] S. Oh, L. Schenato, P. Chen, and S. Sastry, "Tracking and coordination of multiple agents using sensor networks: System design, algorithms and experiments," *Proc. IEEE*, vol. 95, no. 1, pp. 234–254, Jan. 2007.
[3] A. Singh, M. Batalin, M. Stealey, V. Chen, Y. Lam, M. Hansen, T. Harmon, G. S. Sukhatme, and W. Kaiser, "Mobile robot sensing for environmental applications," *Field Service Robot.*, vol. 24, pp. 125–135, 2008.
[4] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. Roboti. Autom.*, vol. 20, no. 2, pp. 243–255, Apr. 2004.
[5] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, Stability of flocking motion Univ. Pennsylvania, Philadelphia, PA, USA, Tech. Rep. MS-CIS-03-03, 2003.
[6] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.
[7] W. Ren and R. W. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Trans. on Automat. Control*, vol. 50, no. 5, pp. 655–661, May 2005.
[8] M. T. Pham and K. T. Seow, "Discrete-event coordination design for distributed agents," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 1, pp. 70–82, Jan. 2012.
[9] K. Whitehouse, C. Karlof, A. Woo, F. Jiang, and D. Culler, "The effects of ranging noise on multihop localization: An empirical study," in *Proc. Int. Symp. Inform. Process. Sens. Netw.*, Los Angeles, CA, USA, Apr. 2005, pp. 73–80.
[10] M. Maróti, B. Kusý, G. Balogh, P. Völgyesi, A. Nádas, K. Molnár, S. Dóra, and A. Lédeczi, "Radio interferometric geolocation," in *Proc. Int. Conf. Embedded Networked Sens. Syst.*, San Diego, CA, USA, Nov. 2005, pp. 1–12.
[11] A. Ledeczi, P. Volgyesi, J. Sallai, B. Kusy, X. Koutsoukos, and M. Maroti, "Towards precise indoor RF localization," in *Proc. Int. Conf. Embedded Networked Sens. Syst.*, San Diego, CA, USA, Nov. 2005, pp. 1–12.
[12] J. Cho, Y. Ding, and J. Tang, "Robust calibration for localization in clustered wireless sensor networks," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 1, pp. 81–95, Jan. 2010.
[13] S. Gecizi, Z. Tian, G. B. Giannakis, Z. Sahinoglu, H. Kobayashi, A. F. Molisch, and H. V. Poor, "Localization via ultra-wideband radios: A look at positioning aspects for future sensor networks," *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 70–84, 2005.
[14] A. Prorok and A. Martinoli, "Accurate indoor localization with ultra-wideband using spatial models and collaboration," *Int. J. Robot. Res.*, vol. 33, no. 4, pp. 547–568, 2014.
[15] C. Taylor and B. Shirmohammadi, "Self localizing smart camera networks and their applications to 3D modeling," in *Proc. Int. Workshop Distrib. Smart Cameras*, Boulder, CO, USA, Oct 2006, pp. 46–50.
[16] S. Funiak, C. Guestrin, M. Paskin, and R. Sukthankar, "Distributed localization of networked cameras," in *Proc. Int. Conf. Inform. Process. Sens. Netw.*, Nashville, TN, USA, Apr. 2006, pp. 34–42.
[17] O. Tekdas and V. Isler, "Sensor placement for triagulation-based localization," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 3, pp. 681–685, Jul. 2010.
[18] M. Meingast, S. Oh, and S. Sastry, "Automatic camera network localization using object image tracks," in *Proc. IEEE Int. Conf. Comput. Vision*, Rio de Janeiro, Brazil, Oct. 2007.
[19] S. Oh, S. Russell, and S. Sastry, "Markov chain Monte Carlo data association for multitarget tracking," *IEEE Trans. Autom. Control*, vol. 54, no. 3, pp. 481–497, Mar. 2009.
[20] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[21] F. Zhang, B. Grocholsky, R. Kumar, and M. Mintz, *Cooperative control for localization of mobile sensor networks*. Philadelphia, PA, USA: GRASP Lab., Univ. Pennsylvania, Internal Paper, 2003.

[22] A. Prorok and A. Martinoli, "A reciprocal sampling algorithm for lightweight distributed multirobot localization," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, San Francisco, CA, USA, Sep. 2011, pp. 3241–3247.

[23] P. Giguere, I. Rekleitis, and M. Latulippe, "I see you, you see me: Cooperative localization through bearing-only mutually observing robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Vilamoura, Portugal, Oct. 2012, pp. 863–869.

[24] R. Sharma, S. Quebe, R. W. Beard, and C. N. Taylor, "Bearing-only cooperative localization," *J. Intell. Robot. Syst.*, vol. 72, no. 3–4, pp. 429–440, 2013.

[25] A. Ahmad, G. D. Tipaldi, P. Lima, and W. Burgard, "Cooperative robot localization and target tracking based on least squares minimization," in *Proc. IEEE Int. Conf. Robot. Autom.*, Karlsruhe, Germany, May 2013, pp. 5696–5701.

[26] S. Tully, G. Kantor, and H. Choset, "Leap-frog path design for multirobot cooperative localization," in *Proc. Int. Conf. Field Service Robots*, Cambridge, MA, USA, Jul. 2009.

[27] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *Int. J. Robot. Res.*, vol. 20, no. 3, pp. 378–400, May 2001.

[28] IRobot [Online]. Available: http://www.irobot.com/

[29] *An Invitation to 3-D Vision*, Y. Ma, J. Kosecka, S. Soatto, and S. Sastry, Eds. Berlin, Germany: Springer-Verlag, 2004.

[30] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *Proc. Brit. Mach. Vision Conf.*, Cardiff, U.K., Sep. 2002, pp. 1–36.

[31] M. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[32] *Springer Handbook of Robotics*, F. Chaumette and S. Hutchinson, Eds. Berlin, Germany: Springer-Verlag, 2008.

[33] J. Suh, S. You, S. Choi, and S. Oh, Vision-based coordinated localization for mobile sensor networks (longer version) ECE, Seoul National Univ., Seoul, Korea, Tech. Rep., 2014. [Online]. Available: http://cpslab.snu.ac.kr/publications/papers/2014_LocMSN_TR.pdf

[34] *Computational Principles of Mobile Robotics*, G. Dudek and M. Jenkin, Eds. Cambridge, U.K.: Cambridge Univ. Press, 2000.

Junghun Suh (S'12) received the B.S. degree in electrical engineering from Kwangwoon University, Seoul, Korea, and the M.S. degree in electrical and computer engineering from Seoul National University, Seoul, in 2009 and 2011, respectively, where he is currently pursuing the Ph.D. degree at the Department of Electrical and Computer Engineering.

His current research interests include cyber-physical systems, robotics, machine learning, optimization and their applications.

Seungil You (S'12) received the B.S. degree in electrical engineering from Seoul National University, Seoul, Korea, in 2011. He is currently working towards the Ph.D. degree at the Department of Computing and Mathematical Science, California Institute of Technology, Pasadena, CA, USA.

His research interests include information theory, control theory, statistical signal processing, smart grids, and optimization.

Sungjoon Choi (S'12) received the B.S. degree in electrical and computer engineering from Seoul National University, Seoul, Korea, in 2012. Currently, he is working towards the Ph.D. degree at the Department of Electrical and Computer Engineering, Seoul National University.

His current research interests include cyber-physical systems, nonparametric Bayesian methods, and machine learning algorithms with applications to robotics.

Songhwai Oh (S'04–M'07) received the B.S. (Hons.), M.S., and Ph.D. degrees in electrical engineering and computer sciences from the University of California, Berkeley, CA, USA, in 1995, 2003, and 2006, respectively.

He is currently an Associate Professor with the Department of Electrical and Computer Engineering, Seoul National University, Seoul, Korea. Before his Ph.D. studies, he was a Senior Software Engineer at Synopsys, Inc., Mountain View, CA, USA, and a Microprocessor Design Engineer at Intel Corporation, Santa Clara, CA, USA. In 2007, he was a Post-Doctoral Researcher with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA, USA. From 2007 to 2009, he was an Assistant Professor of Electrical Engineering and Computer Science in the School of Engineering, University of California, Merced, CA, USA. His current research interests include cyber-physical systems, robotics, computer vision, and machine learning.