

Robot-Assisted Localization Techniques for Wireless Image Sensor Networks

Huang Lee, Hattie Dong, Hamid Aghajan
Wireless Sensor Networks Lab
Department of Electrical Engineering
Stanford University, Stanford, CA 94305
{huanglee, dongh, aghajan}@stanford.edu

Abstract—We present a vision-based solution to the problem of topology discovery and localization of wireless sensor networks. In the proposed model, a robot controlled by the network is introduced to assist with localization of a network of image sensors, which are assumed to have image planes parallel to the agent’s motion plane. The localization algorithm for the scenario where the moving agent has knowledge of its global coordinates is first studied. This baseline scenario is then used to build more complex localization algorithms in which the robot has no knowledge of its global positions. Two cases where the sensors have overlapping and non-overlapping fields of view (FOVs) are investigated. In order to implement the discovery algorithms for these two different cases, a forest structure is introduced to represent the topology of the network. We consider the collection of sensors with overlapping FOVs as a tree in the forest. The robot searches for nodes in each tree through boundary patrolling, while it searches for other trees by a radial pattern motion. Numerical analyses are provided to verify the proposed algorithms. Finally, experiment results show that the sensor coordinates estimated by the proposed algorithms accurately reflect the results found by manual methods.

I. INTRODUCTION

Most applications in wireless sensor networks, including event detection and reporting, rely on the knowledge of sensor positions [1] [2]. High deployment cost and scalability issues make manual localization unrealistic in large networks, and render node localization a fundamental problem in many sensor networks deployments [3] [4] [5] [6] [7].

Recently, research on wireless image sensor networks has received much interest. In such networks, each node is usually only equipped with a low-resolution camera, because of complexity and cost limitations. Furthermore, calibration in multi-camera systems is impractical in large networks [8]. Hence, localization algorithms that

employ lightweight image processing and require minor camera calibration are desired for distributed implementation.

Solutions based on Global Positioning System (GPS) may not work for indoor applications and may be too costly for sensor nodes. Furthermore, in many applications, information about the orientation angle and coverage area of each image sensor is also necessary to perform event and target tracking. Such information cannot be provided by the GPS technology.

The use of signal strength of the RF signal has been used for estimating distances between the nodes for localization purposes [9] [10]. While the technique is attractive from a device cost perspective, experience has shown that such measurements yield poor distance estimates [11]. This is due to the dependence of the radio signal strength on the propagation environment characteristics, which are hard to model and render the measurements subject to fading and multipath effects.

Studies of the localization problem in distributed vision networks have been reported in [12] [13] [14] [15] [16]. These papers mainly focus on the case where the sensor image planes are perpendicular to the object motion plane.

A key motivation for solving the localization problem in sensor networks where the image planes are parallel to the object plane is to address indoor applications with ceiling-mounted cameras (Fig. 1), or outdoor applications in which image sensors are deployed face up in some field monitoring cases (Fig. 2).

The work in [17] employs a model for the image planes parallel to the robot’s motion plane, and uses a MAP approach for simultaneous camera calibration and object tracking. Nevertheless, because of its high computational complexity, the MAP approach presented in that work may not be suitable for wireless sensor networks.

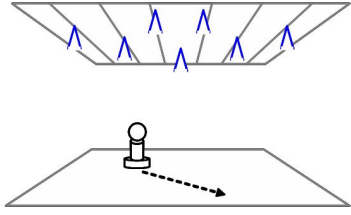


Fig. 1. Indoor application with image planes parallel to the robot's motion plane.

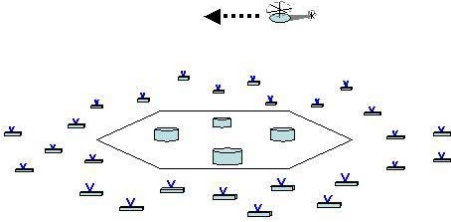


Fig. 2. Outdoor application with image planes parallel to the helicopter's motion plane.

The approach proposed in this paper is based on employing simple and distributed algorithms, in which in-node processing of visual data at the network nodes results in reduced amounts of data shared among the nodes for estimating their relative positions. The paper presents a robot-assisted localization algorithm which can be applied to both overlapping and non-overlapping fields of view (FOVs). The algorithms for both scenarios focus on the topology discovery of the sensor network as opposed to absolute node locations. This allows the proposed techniques to achieve low computational complexity.

We consider two scenarios based on whether the moving agent observed by the network knows and provides its own coordinates to the network. As opposed to the approach in [16], which is based on an autonomous robot, the proposed techniques introduce robot controlled by the network, and are based on discovery procedures by the commanding image sensor node to find its neighboring nodes. The agent is wirelessly controlled by the image sensor node observing it, which sends real-time control commands to the robot via an IEEE 802.15.4 radio link. The proposed methods are implemented and tested for a network of image sensors deployed on the ceiling with image planes parallel to the ground (Fig. 1). The objective of the procedure is to have a distributed and collaborative algorithm by which each node can find its FOV by estimating its coordinates, rotation angle, and height in a global coordinate system.

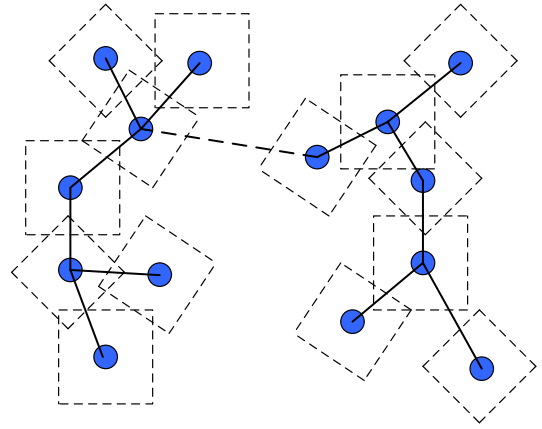


Fig. 3. Forest structure containing two trees.

The remainder of this paper is organized as follows. A model for the network and a description of the setup used to implement the proposed techniques are provided in Section II. Two scenarios where 1) the moving robot knows its global coordinates, and 2) the robot does not know its global positions are discussed in Sections III and IV, respectively. More specifically, in Section III, we review the algorithm in [16] and provide additional simulation results. In Section IV, we explain how to extend the localization algorithm to the second scenario and introduce the discovery algorithms for two cases: 1) sensors with overlapping FOVs, and 2) sensors with non-overlapping FOVs. The localization algorithms are then demonstrated by experiments in an indoor environment in Section V. Finally, Section VI summarizes our conclusions.

II. NETWORK MODEL

The overall algorithm for this scenario is explained in terms of a *forest structure* shown in Fig. 3. The forest is synonymous with the network under study. Each node in the forest represents a sensor. In a forest, each collection of the overlapping sensors is called a *tree*. In each tree, the first localized node is the root of the tree. Different discovery algorithms are considered for searching the neighboring nodes in the same tree, and for searching neighboring trees in a forest. Fig. 4 shows the tree concept in a forest. The arrows in the FOVs form one of the possible paths for a robot to travel in the tree. The picture showing two overlapping screens is a snapshot of a robot traveling between the FOVs of two image sensors.

In our system, the robot controlled by a radio link moves through the FOVs of several sensors. The robot and the control radio device are shown in Fig. 5 (a)

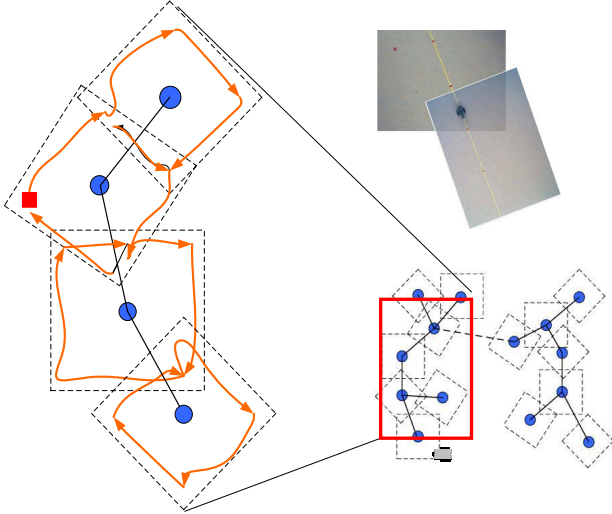


Fig. 4. A tree structure in a forest and a snapshot of a robot traveling in a tree.

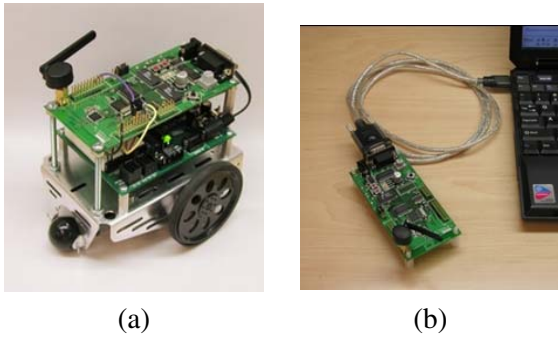


Fig. 5. (a) The robot and (b) The IEEE 802.15.4 command radio device.

and (b), respectively. The implementation employs the Horus platform developed at our lab [18], which provides interface to image sensors and radio boards, as well as a user interface for algorithm development and real-time observations. The FOVs of the image sensors are displayed on the computer screen for visualization purposes and the robot's information such as position coordinates can be extracted or logged. The information can then be used for issuing commands, such as straight-line movements and rotations, to the robot through the wireless link. Because the approach is purely vision-based, the effectiveness of the algorithm is not affected by the integrity loss of the radio signals as they travel through air. Fig. 6 shows the Horus interface on a computer screen with two sensor's FOVs displayed.

The speed of the robot is assumed to be constant, allowing straightforward calculation of the localization

data. Furthermore, the sensor FOVs are assumed to be parallel to the ground plane. To make this assumption plausible, the camera is adjusted such that the projection of the camera's position on the floor corresponds to the center of its FOV.

Two coordinate systems are used during localization: virtual and global. The virtual coordinates of a point measured in pixels are defined in the scope of the FOV, whereas its global coordinates in inches are defined with respect to the entire network. In Fig. 7, the x-y system defines the global coordinate system, where (p_x, p_y) is the sensor node's global location, and θ is the rotation angle. The i-j system is the virtual coordinate system in the child node's FOV. For a point s , we use (s_x, s_y) to denote its global coordinates, and (s_i, s_j) to denote its virtual coordinates.

III. FIRST SCENARIO: ROBOT GLOBAL POSITIONS KNOWN

A. The Algorithm

In this scenario, the robot broadcasts its coordinates at each stop as it travels through the network. The robot may be observed by a few image sensors at each stop, each of which then detects the robot's location on its image plane by simple frame subtraction using an initial background frame. At each stop k , the relationship of the observed virtual coordinates $(s_i(k), s_j(k))$ and the global coordinates $(s_x(k), s_y(k))$ can be expressed by

$$\begin{bmatrix} s_i(k) \\ s_j(k) \end{bmatrix} = \alpha \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} s_x(k) - p_x \\ s_y(k) - p_y \end{bmatrix} + \begin{bmatrix} n_i(k) \\ n_j(k) \end{bmatrix} \quad (1)$$

where (p_x, p_y) are the global coordinates of the sensor, α is the scaling factor in pixels per inch related to sensor height; θ is the rotation angle with respect to the global system, and $(n_i(k), n_j(k))$ is the added noise.

To solve for the unknown variables α , θ , and (p_x, p_y) , we use N observations. Rearranging (1) to cancel (p_x, p_y) , we get

$$\begin{bmatrix} \tilde{s}_i(1) \\ \tilde{s}_j(1) \\ \vdots \\ \tilde{s}_i(N-1) \\ \tilde{s}_j(N-1) \end{bmatrix} = \begin{bmatrix} \tilde{s}_x(1) & \tilde{s}_y(1) \\ \tilde{s}_y(1) & -\tilde{s}_x(1) \\ \vdots & \vdots \\ \tilde{s}_x(N-1) & \tilde{s}_y(N-1) \\ \tilde{s}_y(N-1) & -\tilde{s}_x(N-1) \end{bmatrix} \begin{bmatrix} \alpha \cos \theta \\ \alpha \sin \theta \end{bmatrix} + \begin{bmatrix} \tilde{n}_i(1) \\ \tilde{n}_j(1) \\ \vdots \\ \tilde{n}_i(N-1) \\ \tilde{n}_j(N-1) \end{bmatrix} \quad (2)$$

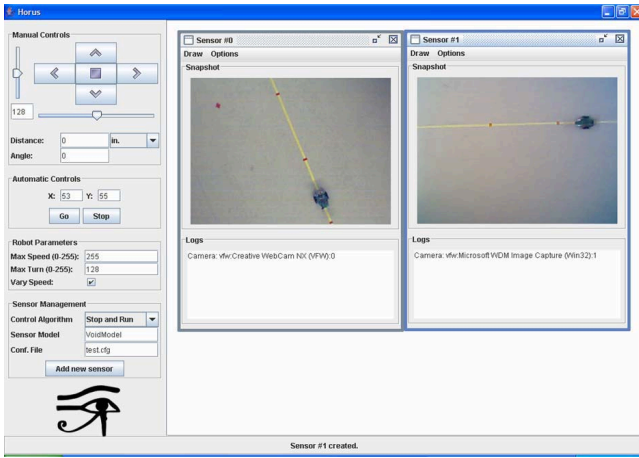


Fig. 6. The Horus interface showing the robot moving in the FOVs of 2 image sensors.

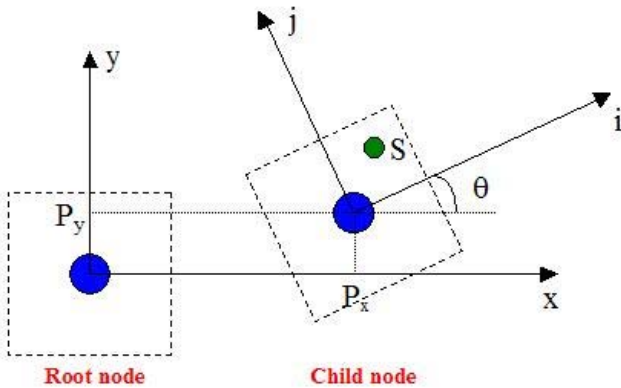


Fig. 7. Sensor network coordinate systems.

where $\tilde{s}_i(k) = s_i(k+1) - s_i(k)$, $\tilde{s}_j(k) = s_j(k+1) - s_j(k)$, $\tilde{s}_x(k) = s_x(k+1) - s_x(k)$, $\tilde{s}_y(k) = s_y(k+1) - s_y(k)$, $\tilde{n}_i(k) = n_i(k+1) - n_i(k)$, and $\tilde{n}_j(k) = n_j(k+1) - n_j(k)$.

Using a minimum of three observation points, we can solve for $\hat{\alpha}$ and $\hat{\theta}$ by a standard least-squares technique, which leads to a closed-form solution. The sensor coordinates (p_x, p_y) can then be derived by

$$\begin{bmatrix} \hat{p}_x \\ \hat{p}_y \end{bmatrix} = \frac{1}{N} \sum_k \begin{bmatrix} s_x(k) \\ s_y(k) \end{bmatrix} - \frac{1}{\hat{\alpha}} \begin{bmatrix} \cos \hat{\theta} & -\sin \hat{\theta} \\ \sin \hat{\theta} & \cos \hat{\theta} \end{bmatrix} \begin{bmatrix} s_i(k) \\ s_j(k) \end{bmatrix}. \quad (3)$$

B. Simulation Results

We use simulations to show that the proposed algorithm can provide highly accurate results, so that we are justified to extend it for use in the scenario where the robot does not know its global coordinates. The simulation parameters are shown in Table I. We first analyze the effect of noise on the image plane. We generate

TABLE I
SIMULATION PARAMETERS.

Parameter	Value
Size of the network (inch ²)	600 × 600
Height of the sensors (inch)	150
Number of sensors	9
Camera's FOV size (pixel)	350 × 350
Camera angle (degree)	45

integer random variables between the numbers a and $-a$ and add them to the observed virtual coordinates. The results are shown in Fig. 8 where each value a on the x-axis represents a noise range a to $-a$, and “observation points” in the legend means the average number of points observed by all sensors. The charts show that the performance improves with more observations. We show the estimated sensor coverage result ($a = 15$ and 4 observations) in Fig. 9, where the blue marks are the true positions while red marks are the estimated results.

The set of other simulations in Fig. 10 show the effect of the error in the received global coordinates. We add uniform random noise $[-a/2, a/2]$ to the coordinates received by the sensors. The performance also improves with more observations. The estimated sensor coverage results ($a = 12$ and 4 observations) are shown in Fig. 11.

IV. SECOND SCENARIO: ROBOT GLOBAL POSITIONS UNKNOWN

When the robot is not capable of broadcasting its global positions, a more sophisticated localization technique is necessary. We first choose a node in the network as the root node. Initially, the robot is controlled by the root node and is in its FOV. We also define the global coordinate system based on the root node's virtual coordinate system, such that the origin of the global system is at the center of the root node's FOV, and the rotation of the root node is zero (see Fig. 7).

Under the assumption that the robot moves with a constant speed, we can find the scaling factor α of the root node by instructing the robot to move a certain time period. Since robot's speed is assumed known, the physical distance traveled can be calculated. Taking the ratio of pixels over physical distance, the scaling factor is determined. Then, we proceed to discover and localize other nodes based on the defined global coordinate system.

To realize the localization process, we use the robot to search other nodes in the forest. Separate discovery

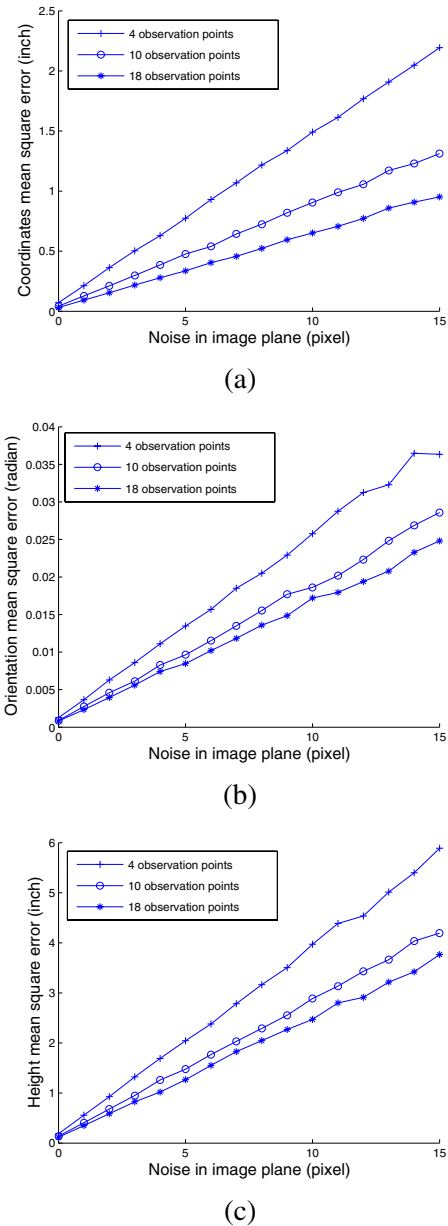


Fig. 8. The effect of noise on the image plane.

algorithms must be designed for overlapping and non-overlapping FOVs cases. We discuss the two cases in the following sections.

A. Overlapping FOV Case

Beginning at the root node, the sensor uses the robot to find its neighboring nodes. The unlocalized nodes found by a localized node are called the children of the localized node. Each node has one parent and zero or more children. If a node already has a parent, it cannot be claimed as the child of another parent. The robot searches for child nodes by patrolling the boundaries

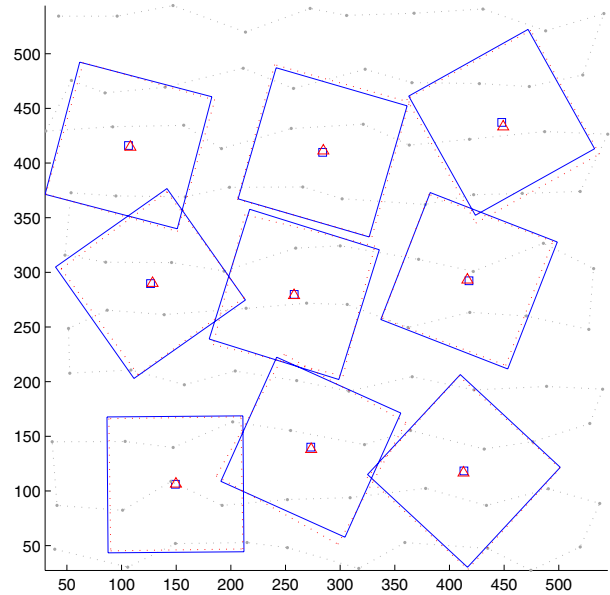


Fig. 9. The estimated sensor coverage result when the noise is on the image plane.

of the localized parent (sensor 0 in Fig. 12). If two sensors have overlapping FOVs, the overlapping region covers the boundaries of both sensors. Hence, it is most sufficient to only search along the boundaries.

Once a child sensor (sensor 1) sees the robot in its FOV, it broadcasts a message making its presence known. When the parent receives the broadcast, it ceases patrolling and remembers the global coordinates of the robot's current location (s_x, s_y) . Child node also notes robot's virtual coordinates with respect to its own FOV, (s_i, s_j) . Parent node then instructs the robot to move to two more points within the common region of the FOVs. Child node extracts the virtual coordinates of these two points as well. Since the parent node knows its global coordinates, it can calculate the robot's global coordinates based on the robot's positions on its image plane. The parent then sends the global coordinates $(s_x(1), s_y(1))$, $(s_x(2), s_y(2))$, and $(s_x(3), s_y(3))$ to its child node. Equipped with the global coordinates and their corresponding virtual coordinates, the child node localizes itself using the algorithm in Section III.

Depth as opposed to breadth traversal is used for discovery to minimize distance traveled and the number of control handover times. After the localization is done, the parent hands robot control to its new child, and the new child repeats the patrolling and localizing protocol. When a sensor finishes patrolling its boundary and all its child nodes are localized, it instructs the robot to move back to its parent and hands robot control back to its

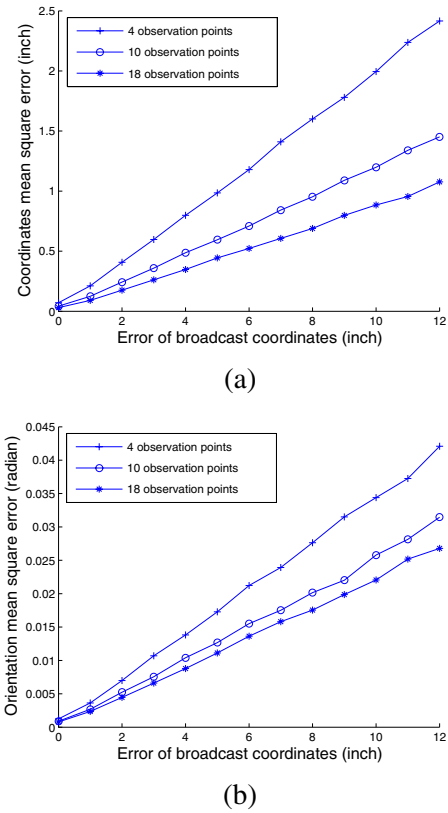


Fig. 10. The effect of error in the received global coordinates.

parent. When a node gets control back, it picks up where it left off and continues patrolling until its boundary search ends. Once the root node finishes patrolling its boundary, the overlapping FOVs portion of the program ends. Using this recursive algorithm, all the nodes in a tree can be localized.

B. Non-overlapping FOV Case

Once the root node finds and localizes all the nodes in the same tree, it runs the discovery algorithm to search for neighboring trees. From the center of the localized sensor (sensor 0 in Fig. 13), the robot is instructed to patrol radially outwards in eight directions. For each patrol direction, the robot moves straight forward for a certain time period specified in the software.

Since the search path orientation is critical in determining the rotation angle of any child found, a step is taken to tune the robot's heading while it is still in sensor 0's FOV. The robot is commanded to arrive at two points in the FOV that are on the search path, Q_1 and Q_2 , before being sent outside of the FOV. At each point, the robot's orientation is adjusted by turning an angle defined by two vectors: one indicating the robot's current orientation, and another indicating the robot's desired orientation. If

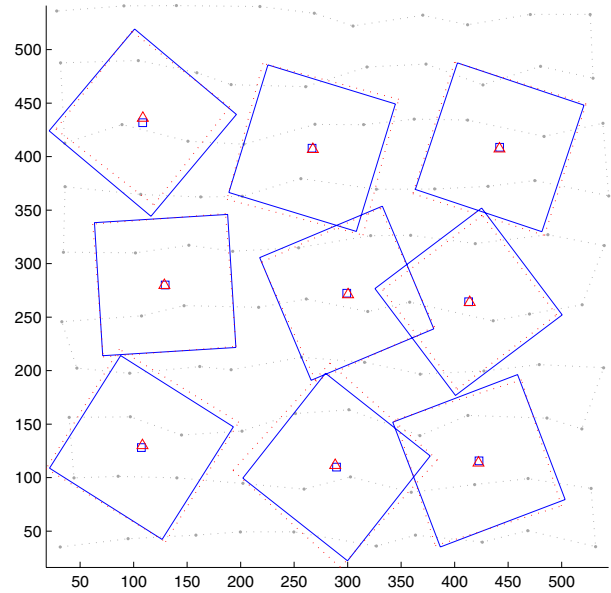


Fig. 11. The estimated sensor coverage result when there is error in the received global coordinates.

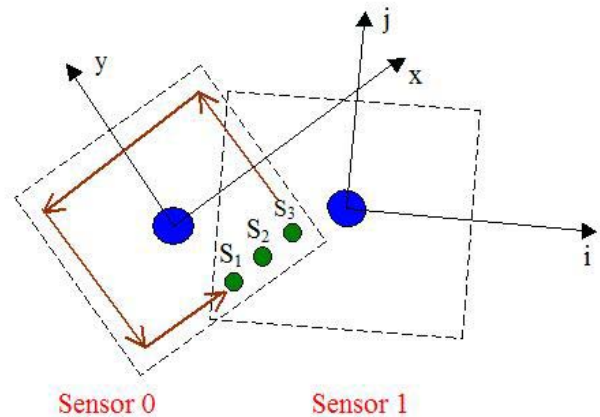


Fig. 12. Discovery algorithm for overlapping case.

no other sensor is encountered during the search path, the robot is then called to move straight backwards to the center of sensor 0. The robot proceeds to turn an angle for the next search direction.

If a child sensor (sensor 1) sees the robot in its FOV, a procedure similar to the one in the overlapping case is used to localize it, but with a key difference. The global coordinates of the three observation points ($s_x(1)$, $s_y(1)$), ($s_x(2)$, $s_y(2)$), and ($s_x(3)$, $s_y(3)$) cannot be simply extracted by sensor 0 because they are outside of its FOV. To get the global coordinates ($s_x(1)$, $s_y(1)$), the time between the moment at which robot moves outside of sensor 0's FOV and the robot's discovery by sensor 1 is stored. This time difference multiplied

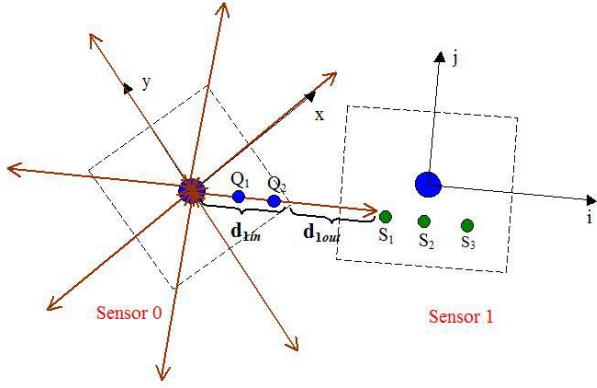


Fig. 13. Discovery algorithm for non-overlapping case.

by robot's speed gives the physical distance the robot has traveled since moving outside of sensor 0's FOV, d_{1out} . The sum of d_{1out} and d_{1in} gives the overall distance traveled from the origin of the physical system. Physical coordinates can then be calculated. The robot then travels to two more points $s(2)$ and $s(3)$, and their global coordinates are calculated. Sensor 1 computes the corresponding virtual coordinates of these points. Now, similar to the overlapping case, sensor 0 sends sensor 1 the global coordinates of the three points, and sensor 1 proceeds to localize itself using the algorithm presented in Section III. After sensor 1 is localized, the robot is commanded to move back to sensor 0's FOV. To do this, the robot's heading is calculated using the calculated localization information. This process can be repeated more than once to improve the performance. When the robot enters the FOV of sensor 0, sensor 0 picks up where it left off and continues radial patrolling until all eight directions are completed and all its neighbors are localized.

Overall, the algorithms for both scenarios focus on the topology discovery of the sensor network as opposed to absolute node locations. Hence, we are able to achieve low computational complexity at the expense of precise sensor locations.

V. EXPERIMENTAL RESULTS

A. Robot Global Positions Known

Refer to Fig. 14 for the setup of this experiment. The arrows define the global coordinate system. All coordinates are in inches. The robot is instructed to travel to the three global points $(0, 0)$, $(24, 12)$ and $(0, 36)$ represented by the three circles (blue in the electronic version). The sensor's manually measured localization data and its computed data are presented in Table II. The

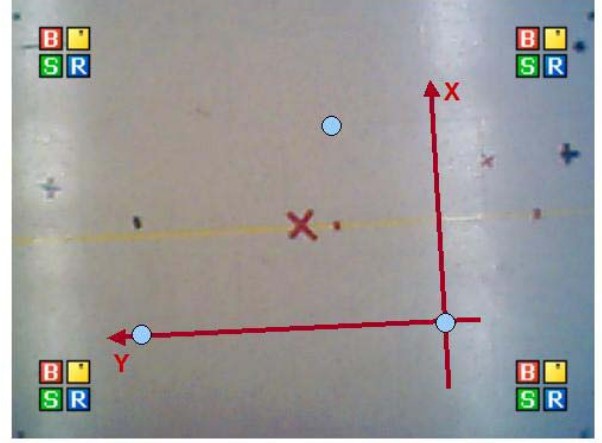


Fig. 14. Experiment run for the scenario where where robot global positions are known.

TABLE II
MEASURED VS. COMPUTED LOCALIZATION DATA FOR THE SCENARIO WHERE ROBOT GLOBAL POSITIONS ARE KNOWN

	Measured	Computed
Center coordinates (inches)	(12, 16)	(11.262, 15.916)
Scaling factor (pixels/inch)	4.5	4.698
Rotation angle (degrees)	-94.7	-90.229

two sets of data closely resemble each other. The small differences can be attributed to the following reasons.

As the robot travels to each of the desired observation points, it may not land at the exact coordinates targeted due to mechanical errors. Further, the robot is not a point object, but has finite dimensions. Hence, when the program extracts the virtual coordinates of the robot, the accuracy suffers due to the robot's finite size.

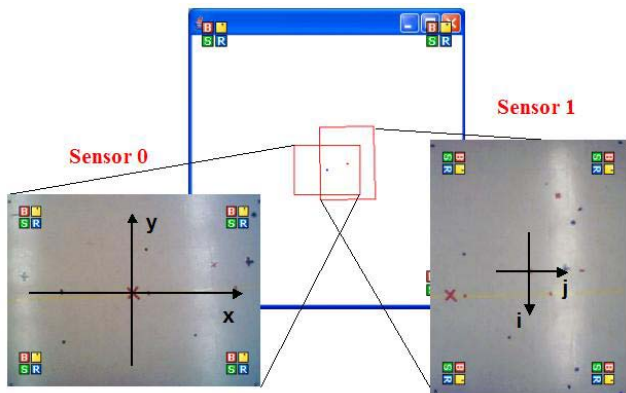
This algorithm is robust against poor mechanical controls and inconsistent robot speed. As long as the robot travels to the prescribed points, despite variations in time or route taken, fairly accurate localization data can be computed. Moreover, since the localization of each sensor node is independent of the localization results of the others, there is no error propagation.

B. Robot Global Positions Unknown

For both overlapping and non-overlapping FOVs, since the localization of each node depends on the localization data of its parent, error propagation exists. To minimize error propagation, the network can run the algorithms several times using root nodes as geometrically far apart as possible. For each node, results from the runs where it is relatively close to the root node can be averaged to provide localization data that have little



(a)



(b)

Fig. 15. Overlapping FOVs: (a) Experiment run (b) Illustration of localization results.

TABLE III

MEASURED VS. COMPUTED LOCALIZATION DATA FOR OVERLAPPING FOVS

	Measured	Computed
Center coordinates (inches)	(23, 7)	(22.27, 7.06)
Scaling factor (pixels/inch)	4.5	3.95
Rotation angle (degrees)	-90	-89.038

error propagation.

a) Overlapping FOVs: Refer to Fig. 15 for the setup of this experiment. In Fig. 15 (a), the left screen is the FOV of the parent node and the right screen is the FOV of the child node. The square close to $s(1)$ (red in the electronic version) is the point at which the robot is discovered by sensor 1. The three observation points are $s(1)$, $s(2)$ and $s(3)$ (green in the electronic version). Parent node's scaling factor is computed to be 4.38 pixels/inch, compared to the measured value of 4.5 pixels/inch. Child node's measured and calculated data are presented in Table III. It can be observed that the measured and computed data have only little discrepancy. Fig. 15 (b) presents the coverage area results.

Since three points are needed for localization, some points may lay outside of the overlapping region of

the FOVs, especially when this region is small. If this happens, localization will not be done. We can insert a routine that ensures three points are extracted by calling the robot back should it move outside of the overlapping region of the FOVs. Of course, localization results will be more accurate if more observation points are taken.

1) Non-overlapping FOVs: Refer to Fig. 16 for the setup. In Fig. 16 (a), the left screen is the FOV of sensor 0 and the right screen is the FOV of sensor 1. In this experiment, sensor 1 is discovered while the robot is performing radial search in the first of eight directions, which is indicated by the squares (blue in the electronic version). The square close to $s(1)$ (red in the electronic version) on sensor 1's FOV is where the robot is first discovered. The three observation points are $s(1)$, $s(2)$ and $s(3)$ (green in the electronic version). Sensor 1's scaling factor is calculated to be 4.35 pixels/inch versus the true value of 4.5 pixels/inch. Sensor 1's localization data are presented in Table IV, and the two sets of data are close to one another. The coverage area results are illustrated in Fig. 16 (b).

The main reason for the difference between computed and measured coordinates and rotation data is that the localization algorithm is sensitive to the robot's path exiting sensor 0's FOV. Because the global coordinates are calculated based on the angle of exit, it is crucial that the robot follows the correct path as much as possible. The localization data from several runs can be averaged to improve the performance, at the expense of longer experiment time.

In the non-overlapping case, the assumption of a fixed robot speed is not only critical in determining the root node's scaling factor, but it is also important for calculating the global coordinates of the observation points outside the root node's FOV, and hence in computing localization data. Measuring the speed before each run is therefore vital.

Similar to the overlapping FOV case, some observation points may lay outside of the child sensor's FOV. Shorter distance between test vectors will reduce this possibility. Furthermore, a procedure can be incorporated to call the robot back as soon it moves outside of the child's FOV to make sure that three test points are taken for localization.

The algorithm currently contains eight patrol directions. If a sensor's FOV does not overlap with any of the eight search paths, it does not get localized. One solution is to add more patrol directions as shown in Fig. 17. Depending on the goals for a specific network's localization, trade-off between accuracy and time can be

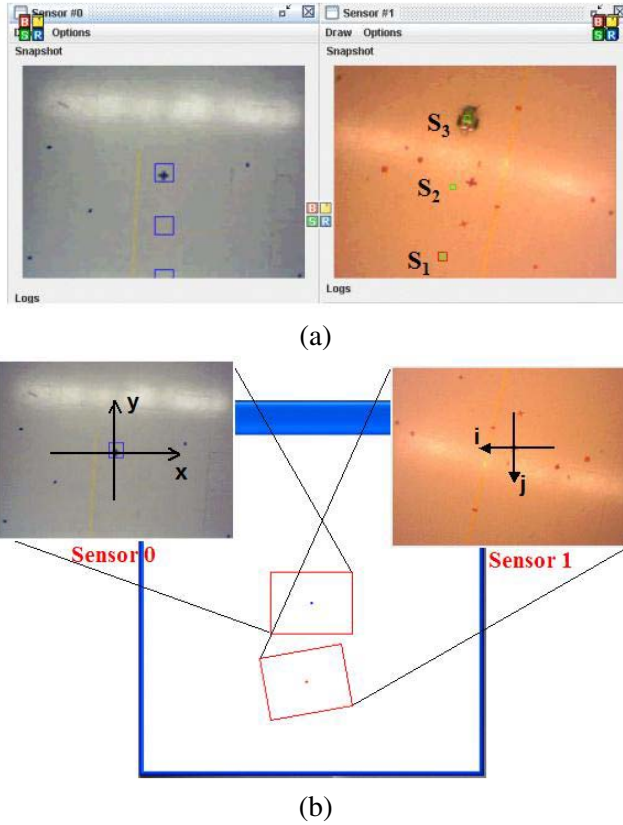


Fig. 16. Non-overlapping FOVs: a) Experiment run b) Illustration of localization results.

TABLE IV
MEASURED VS. COMPUTED LOCALIZATION DATA FOR
NON-OVERLAPPING FOVS

	Measured	Computed
Center coordinates (inches)	(-1, -64)	(-4.74, -70.19)
Scaling factor (pixels/inch)	4.5	4.31
Rotation angle (degrees)	-180	-169.882

appropriately adjusted.

VI. CONCLUSIONS

A localization solution focusing on network topology discovery for vision-enabled wireless sensor networks was presented. Because the algorithm is purely image-based, the deterioration of radio signals through air is not a major concern. A controllable robot was introduced to assist the localization process of image sensors deployed on the ceiling with image planes parallel to the ground. Two scenarios in which the robot knows its global coordinates or does not have this information were discussed. The localization algorithm proposed for the first scenario was verified by simulations. To extend this localization algorithm to the second scenario, two

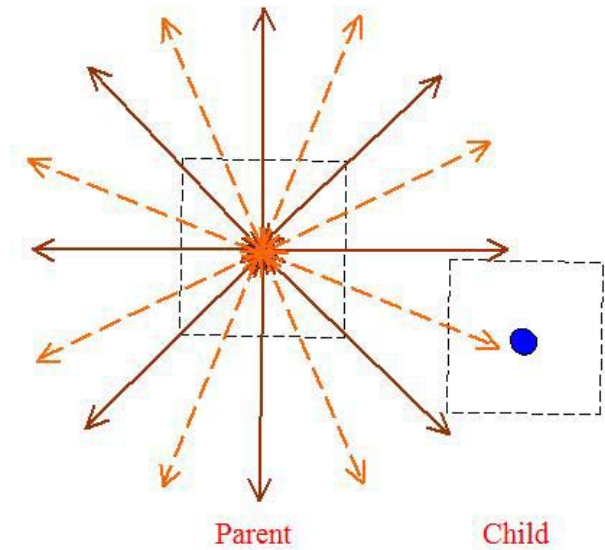


Fig. 17. Improving the accuracy of the non-overlapping discovery algorithm by increasing patrol directions.

different discovery algorithms were proposed for the cases of: 1) sensors with overlapping FOVs, and 2) sensors with non-overlapping FOVs. Experiments were performed for both cases on a platform of multiple image sensors in which visual observations of the moving robot were used to issue control commands by the observing network node. Different neighbor discovery schemes were proposed and used in discovering the topology of the network, facilitating localization of the network nodes via data sharing between the nodes as they make observations of the moving robot.

REFERENCES

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.
- [2] C.-Y. Chong and S. P. Kumar, "Sensor networks: evolution, opportunities, and challenges," *Proc. IEEE*, vol. 91, no. 8, pp. 1247–1256, Aug. 2003.
- [3] J. C. Chen, K. Yao, and R. E. Hudson, "Source localization and beamforming," *IEEE Signal Processing Mag.*, vol. 19, no. 2, pp. 30–39, Mar. 2002.
- [4] L. Hu and D. Evans, "Localization for mobile sensor networks," in *Proc. of the ACM MobiCom'04*, Sept. 2004.
- [5] A. Galstyan, B. Krishnamachari, K. Lerman, and S. Patten, "Distributed online localization in sensor networks using a moving target," in *Proc. of the Third International Symposium on Information Processing in Sensor Networks*, Apr. 2004, pp. 61–70.
- [6] C. Savarese, J. M. Rabaey, and J. Beutel, "Locationing in distributed ad-hoc wireless sensor networks," in *Proc. ICASSP*, May 2001, pp. 2037–2040.
- [7] P. Pathirana, N. Bulusu, A. Savkin, and S. Jha, "Node localization using mobile robots in delay-tolerant sensor networks,"

IEEE Transactions on Mobile Computing, vol. 4, pp. 285–296, 2005.

- [8] F. Pedersini, A. Sarti, and S. Tubaro, “Multi-camera parameter tracking,” in *in IEE Proceedings Vision, Image and Signal Processing*, vol. 148, Feb. 2001, pp. 70–77.
- [9] P. Bergamo and G. Mazzini, “Localization in sensor networks with fading and mobility,” in *Proc. of IEEE PIMRC*, 2002, pp. 750–754.
- [10] F. Mondinelli and Z. M. Kovacs-Vajna, “Self localizing sensor network architectures,” *IEEE Transactions on Instrumentation and Measurement*, pp. 277–283, April 2003.
- [11] J. Hightower, R. Want, and G. Borriello, “Spoton: An indoor 3d location sensing technology based on rf signal strength,” *University of Washington UW CSE 00-02-02*, Feb. 2000.
- [12] W. Mantzel, H. Choi, and R. G. Baraniuk, “Distributed alternating localization-estimation of camera networks,” in *Proceedings of the 38th Asilomar Conference on Signals, Systems, and Computers*, Nov. 2004.
- [13] D. Agathangelou, B. P. Lo, J. L. Wang, and G.-Z. Yang, “Self-configuring video-sensor networks,” in *Proceedings of the 3rd International Conference on Pervasive Computing (PERVASIVE 2005)*, May 2002.
- [14] M. Carreras, P. Ridao, R. Garcia, and T. Nicosevici, “Vision-based localization of an underwater robot in a structured environment,” in *IEEE International Conference on Robotics and Automation ICRA’03*, vol. 1, 2003, pp. 971–976.
- [15] H. Lee and H. Aghajan, “Collaborative self-localization techniques for wireless image sensor networks,” in *Asilomar Conference on Signals, Systems and Computers*, Asilomar, CA, Oct. 2005.
- [16] H. Lee, L. Savidge, and H. Aghajan, “Subspace techniques for vision-based node localization in wireless sensor networks,” in *International Conference on Acoustics, Speech and Signal Processing*, Toulouse, France, May 2006.
- [17] A. Rahimi, B. Dunagan, and T. Darrell, “Simultaneous calibration and tracking with a network of non-overlapping sensors,” in *Proceedings of the IEEE Computer Vision and Pattern Recognition*, vol. 1, June 2004, pp. I-187 – I-194.
- [18] C. McCormick, P. Lalgand, H. Lee, and H. Aghajan, “Distributed agent control with self-localizing wireless image sensor networks,” in *COGnitive systems with Interactive Sensors*, Paris, France, Mar. 2006.