

Localizing Sensors from Their Responses to Targets

Shigeo SHIODA^{†a)}, Member

SUMMARY The spatial relations between sensors placed for target detection can be inferred from the responses of individual sensors to the target objects. Motivated by this fact, this paper proposes a method for estimating the location of sensors by using their responses to target objects. The key idea of the proposal is that when two or more sensors simultaneously detect an object, the distances between these sensors are assumed to be equal to a constant called *the basic range*. Thus, new pieces of proximity information are obtained whenever an object passes over the area in which the sensors are deployed. This information is then aggregated and transformed into a two dimensional map of sensors by solving a nonlinear optimization problem, where the value of the basic range is estimated together. Simulation experiments verify that the proposed algorithm yields accurate estimates of the locations of sensors.

key words: sensor, localization, range free, target detection, optimization

1. Introduction

Recent progress in electronics and micromechanics has led to the creation of small sensors that have a built-in battery and the capability for wireless communication. Such sensors present us with great opportunities for a wide range of applications, including tracking, emergency response, or environmental monitoring. Even if the individual sensors do not provide much data, substantial information can be obtained if a large number of sensors are deployed.

In most cases, the data collected by a sensor is only meaningful in conjunction with knowledge of the location of that sensor. The location of a sensor is also important for the geographical routing or topology control of sensor networks. For these reasons, accurate and low-cost sensor *localization* has been a key technical challenge in wireless sensor networks [1].

This paper focuses on the localization of sensors placed for target detection. Spatial relations between these sensors can be inferred from the responses of the individual sensors to a target object. For example, if two sensors simultaneously detect an object, they are likely to be located in close proximity to each other. Inspired by this observation, this paper proposes a method for estimating sensor locations based on the detection of objects by individual sensors. The proposed method assumes that when two or more sensors simultaneously detect an object, the distances between these sensors are equal to a constant, called *the basic range*. Thus,

in the proposed method, new pieces of proximity information are obtained whenever an object passes over the area in which the sensors are deployed. This information can then be aggregated and transformed into a two-dimensional map of sensors. This transformation is achieved by solving a nonlinear optimization problem, where the value of the basic range is estimated together.

The proposed localization technique is range free; sensors do not need to be able to detect the distances to neighbor sensors, but only to detect a target object when it is in their proximity. Note that the location of the object does not need to be known. The proposed algorithms work even if the locations of the sensors are unknown at first. A map of the locations of the sensors is then gradually built up as a number of objects pass over the area in which the sensors are deployed. Simulation experiments verified that the proposed simple algorithm can yield surprisingly accurate estimates of the absolute locations of sensors.

This manuscript is organized as follows: Sect. 2 is an overview of the literature on sensor localization. Section 3 explains the key idea of the proposal, formulates it as a nonlinear optimization problem, and presents an efficient solution. Section 4 numerically evaluates the effectiveness of the proposal using a simple simulation experiment. Finally, we present our conclusions in Sect. 5.

2. Related Work

Existing studies on localization can be classified as either range based or range free. In range-based methods, sensors at unknown locations determine the distances to special sensor nodes, usually called *anchors*, whose exact locations are known, based on the signal strength or time of arrival; they then use trilateration, triangulation, or maximum-likelihood estimation to determine their locations themselves. Sensors in unknown locations can cooperatively determine their locations by sharing their estimated locations. This leads to a new paradigm for localization, called *cooperative localization*, and a number of studies have been published. The simplest implementation of this is a centralized algorithm, in which a central processor gathers the distance measurements from the sensors and determines the locations of all the sensors by solving some global optimization problem [2]–[5]. Centralized algorithms are usually not scalable, so most of the algorithms for cooperative localization are distributed [6]–[10] (see also [1] and the references therein).

Range-free methods rely only on the locations of the

Manuscript received March 31, 2014.

Manuscript revised August 7, 2014.

[†]The author is with the Graduate School of Engineering, Chiba University, Chiba-shi, 263-8522 Japan.

a) E-mail: shioda@faculty.chiba-u.jp

DOI: 10.1587/transcom.E98.B.145

anchor nodes; they do not use the distance to these nodes. In the centroid algorithm [11], anchors are placed in a way that their regions of coverage overlap, and they transmit periodic beacon signals to provide their positions to the neighboring sensors. Each sensor localizes itself in relation to the centroid of the anchor nodes. In the DV-HOP algorithm [12], each anchor floods the network by sending packets that contain its physical location. Each recipient of a packet knows how many hops are required to reach the corresponding anchor node and its physical location; this information is used to estimate the position of the sensor.

The proposal of this paper is range free in the sense that sensors do not need to have functions for ranging. However, it also has a flavor of range-base approaches because the pair-wise distances between sensors estimated from their responses to targets is used for localization. This paper is an extended version of our conference paper [13], which focuses on obtaining relative sensor-location map; the proposal of this paper allows us to have the absolute sensor-location map as well as the relative map.

3. Locating Sensors by Detecting Objects

3.1 Proximity Estimation Between Sensors

Consider a two-dimensional field in which N sensors are deployed. Let \mathcal{A}_i be the sensing area of the i th sensor. Sensing area \mathcal{A}_i is assumed to be time invariant. The i -th sensor detects a target object at time t if and only if $\mathcal{A}_i \cap \mathcal{O}_t \neq \emptyset$, where \mathcal{O}_t is the area occupied by the object at time t .

The distance between sensors i and j is denoted by d_{ij} . At the beginning, it is assumed that $d_{ij} = \infty$ for all $i-j$ pairs. At regular time intervals, each sensor investigates if it detects some target; the result is sent to the central server via a low-bit-rate but long-range wireless link[†]. If sensors i and j detect a target object at the same instant, the server assumes that $d_{ij} = d$, where d is a parameter called the *basic range*. Figure 1 shows an example of proximity estimation by the proposed algorithm. Each sensor has a sensing area shaped as the sector of a circle. An object with a rectangle shape moves from left to right. Sensors 1 and 2 detect the target when the object is at location 1; sensors 5 and 6 then detect the target when the object arrives at location 2. Lastly, sensors 8 and 9 detect the target when the object is located at location 3. Based on these results, the server recognizes that $d_{12} = d_{56} = d_{89} = d$.

The estimated distances should have errors. In fact, in Fig. 1, d_{12} is not equal to d_{56} or d_{89} . The errors in the estimated distances between sensors, however, would balance against each other; an estimate of the distance between some sensor pairs is smaller than the true one, while an estimate of the distance between other sensor pairs is larger than the true one. The errors in estimated distances between different sensor pairs would be more well-balanced as the number

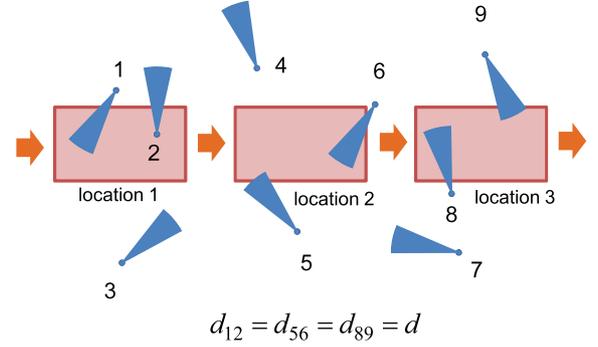


Fig. 1 Distance estimation based on the detection of a rectangle-shaped object by individual sensors.

of sensors increases. As a result, the proposed localization algorithm provides more accurate estimates of sensor locations as the number of sensors (sensor densities) increases, as shown in Sect. 4. In general, the cooperative localization is not so sensitive against errors in distance measurements [15].

Remark 1. This paper assumes that sensors do not have a function for identifying each of objects when multiple objects exist in the sensor-deployment field, and thus only one object is allowed to exist in the field. In a typical application, we let only one object move about in the field for sensor localization after sensor deployment. Note that if sensors have an ability to identify objects, the proposed algorithm can be applied to cases where multiple objects exist. Also note that objects can be identified when locations of sensors are roughly known even if each sensor does not have an ability of object identification; when multiple objects exist, sensors detecting objects are partitioned into several clusters, and each cluster is composed of sensors detecting each object [16]. Thus, once sensors are roughly localized, we can localize sensors more accurately using the proposed algorithm based on the responses of sensors to objects.

3.2 Formulation and Solution of Relative Sensor Localization

A central server first creates a map of the relative sensor locations based on the distance matrix $D = \{d_{ij}\}$. Let s_i and \hat{s}_i , respectively, denote the location of sensor i and its estimate. Without loss of generality, the location of the 1st sensor is assumed to be at the origin of the coordinate system; that is, $s_1 = \mathbf{0}$. The server estimates the locations of the sensors by solving the following nonlinear optimization problem:

$$(\hat{s}_2, \dots, \hat{s}_N) = \arg \min_{(x_2, \dots, x_N)} \varepsilon(x_2, \dots, x_N), \quad (1)$$

where

$$\varepsilon(x_2, \dots, x_N) \stackrel{\text{def}}{=} \sum_{i=1}^N \sum_{j \neq i; d_{ij} < \infty} (|x_i - x_j| - d_{ij})^2, \quad (2)$$

[†]Some proposals, such as a wide area ubiquitous network [14], aim at supporting such communication capability.

and $\mathbf{x}_1 = \mathbf{0}$. The right-hand side of (1) is a typical optimization problem, frequently appearing in the field of graph drawing, and this can be efficiently solved by a method called *stress majorization* [17]. We will briefly explain this, as follows. We define

$$g(\mathbf{x}_2, \dots, \mathbf{x}_N, \mathbf{y}_2, \dots, \mathbf{y}_N) \stackrel{\text{def}}{=} \sum_{i=1}^N \sum_{j \neq i; d_{ij} < \infty} \left\{ |\mathbf{x}_i - \mathbf{x}_j|^2 + (d_{ij})^2 - 2(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{y}_i - \mathbf{y}_j) \frac{d_{ij}}{|\mathbf{y}_i - \mathbf{y}_j|} \right\}.$$

The function $g(\mathbf{x}_2, \dots, \mathbf{x}_N, \mathbf{y}_2, \dots, \mathbf{y}_N)$ has the following property:

$$\begin{aligned} \varepsilon(\mathbf{x}_2, \dots, \mathbf{x}_N) &= g(\mathbf{x}_2, \dots, \mathbf{x}_N, \mathbf{x}_2, \dots, \mathbf{x}_N) \\ &\leq g(\mathbf{x}_2, \dots, \mathbf{x}_N, \mathbf{y}_2, \dots, \mathbf{y}_N). \end{aligned} \quad (3)$$

The solution obtained by stress majorization is recursive. Assume that we have obtained estimates at the n th recursive step $(\hat{\mathbf{s}}_2^{(n)}, \dots, \hat{\mathbf{s}}_N^{(n)})$. The estimates at the $(n+1)$ th recursive step $(\hat{\mathbf{s}}_2^{(n+1)}, \dots, \hat{\mathbf{s}}_N^{(n+1)})$ are then the $(\mathbf{x}_2, \dots, \mathbf{x}_N)$ that minimize $g(\mathbf{x}_2, \dots, \mathbf{x}_N, \hat{\mathbf{s}}_2^{(n)}, \dots, \hat{\mathbf{s}}_N^{(n)})$; thus, $(\hat{\mathbf{s}}_2^{(n+1)}, \dots, \hat{\mathbf{s}}_N^{(n+1)})$ should satisfy

$$\begin{aligned} &\left[\frac{\partial g(\mathbf{x}_2, \dots, \mathbf{x}_N, \hat{\mathbf{s}}_2^{(n)}, \dots, \hat{\mathbf{s}}_N^{(n)})}{\partial \mathbf{x}_i} \right]_{(\mathbf{x}_2, \dots, \mathbf{x}_N) = (\hat{\mathbf{s}}_2^{(n+1)}, \dots, \hat{\mathbf{s}}_N^{(n+1)})} \\ &= 2 \sum_{j \neq i; d_{ij} < \infty} \left\{ (\hat{\mathbf{s}}_i^{(n+1)} - \hat{\mathbf{s}}_j^{(n+1)}) - (\hat{\mathbf{s}}_i^{(n)} - \hat{\mathbf{s}}_j^{(n)}) \frac{d_{ij}}{|\hat{\mathbf{s}}_i^{(n)} - \hat{\mathbf{s}}_j^{(n)}|} \right\} \\ &= 0, \end{aligned} \quad (4)$$

where $\hat{\mathbf{s}}_1^{(n)} = \hat{\mathbf{s}}_1^{(n+1)} = \mathbf{0}$. It follows from (4) that

$$\begin{aligned} &c_i \hat{\mathbf{s}}_i^{(n+1)} - \sum_{j=1; j \neq i}^N c_{ij} \hat{\mathbf{s}}_j^{(n+1)} \\ &= \hat{\mathbf{s}}_i^{(n)} \sum_{j \neq i; d_{ij} < \infty} \frac{d_{ij}}{|\hat{\mathbf{s}}_i^{(n)} - \hat{\mathbf{s}}_j^{(n)}|} - \sum_{j \neq i; d_{ij} < \infty} \frac{d_{ij} \hat{\mathbf{s}}_j^{(n)}}{|\hat{\mathbf{s}}_i^{(n)} - \hat{\mathbf{s}}_j^{(n)}|}, \end{aligned}$$

where

$$c_i = \sum_{j=1; j \neq i}^N c_{ij}, \quad c_{ij} \stackrel{\text{def}}{=} \begin{cases} 1, & d_{ij} < \infty \\ 0, & \text{otherwise} \end{cases}$$

The above equation is transformed into the following recursive formula:

$$\begin{aligned} (\hat{\mathbf{s}}_{2x}^{(n+1)}, \dots, \hat{\mathbf{s}}_{Nx}^{(n+1)})^T &= A^{-1} (\beta_{2x}^{(n)}, \dots, \beta_{Nx}^{(n)})^T, \\ (\hat{\mathbf{s}}_{2y}^{(n+1)}, \dots, \hat{\mathbf{s}}_{Ny}^{(n+1)})^T &= A^{-1} (\beta_{2y}^{(n)}, \dots, \beta_{Ny}^{(n)})^T, \end{aligned} \quad (5)$$

where

$$\beta_i^{(n)} = \hat{\mathbf{s}}_i^{(n)} \sum_{j=1; d_{ij}=d}^N \frac{d_{ij}}{|\hat{\mathbf{s}}_i^{(n)} - \hat{\mathbf{s}}_j^{(n)}|} - \sum_{j=1; d_{ij}=d}^N \frac{d_{ij} \hat{\mathbf{s}}_j^{(n)}}{|\hat{\mathbf{s}}_i^{(n)} - \hat{\mathbf{s}}_j^{(n)}|},$$

$$A = \underbrace{\begin{pmatrix} c_2 & -c_{23} & \dots & -c_{2N} \\ -c_{32} & c_3 & \dots & -c_{3N} \\ \vdots & \vdots & \ddots & \vdots \\ -c_{N2} & -c_{N3} & \dots & c_N \end{pmatrix}}_{N-1}.$$

In (5), the x- and y-coordinates of $\hat{\mathbf{s}}_i^{(n+1)}$ are $\hat{\mathbf{s}}_{ix}^{(n+1)}$ and $\hat{\mathbf{s}}_{iy}^{(n+1)}$, and those of $\beta_i^{(n)}$ are $\beta_{ix}^{(n)}$ and $\beta_{iy}^{(n)}$. Appendix shows that A has its inverse. It follows from the definition of $\{\hat{\mathbf{s}}_i^{(n)}\}$ and (3) that

$$\begin{aligned} \varepsilon(\hat{\mathbf{s}}_2^{(n)}, \dots, \hat{\mathbf{s}}_N^{(n)}) &= g(\hat{\mathbf{s}}_2^{(n)}, \dots, \hat{\mathbf{s}}_N^{(n)}, \hat{\mathbf{s}}_2^{(n)}, \dots, \hat{\mathbf{s}}_N^{(n)}) \\ &\geq g(\hat{\mathbf{s}}_2^{(n+1)}, \dots, \hat{\mathbf{s}}_N^{(n+1)}, \hat{\mathbf{s}}_2^{(n)}, \dots, \hat{\mathbf{s}}_N^{(n)}) \\ &\geq g(\hat{\mathbf{s}}_2^{(n+1)}, \dots, \hat{\mathbf{s}}_N^{(n+1)}, \hat{\mathbf{s}}_2^{(n+1)}, \dots, \hat{\mathbf{s}}_N^{(n+1)}) \\ &= \varepsilon(\hat{\mathbf{s}}_2^{(n+1)}, \dots, \hat{\mathbf{s}}_N^{(n+1)}). \end{aligned}$$

Thus, repeated substitution via (5) constantly decreases the objective function, and it should converge to the point at which it is (locally) minimized.

3.3 Avoidance of Dependence on Initial Solutions

Stress majorization (and other descent methods) requires a set of initial estimates of the sensor locations, $(\hat{\mathbf{s}}_2^{(1)}, \dots, \hat{\mathbf{s}}_N^{(1)})$, and thus, if there are multiple local minimums, the final solution may be dependent on the initial estimates. It was found through experiments [15] that, in most cases, the solution described in Sect. 3.2 reaches the optimal solution regardless of the initial estimates, providing that all pair-wise distances between sensors are known; in other words, all elements of the distance matrix D are finite. In contrast, when D has non-finite-value elements, different initial estimations generally lead to objective functions that have different values.

Considering this result, this paper uses the following two-step localization (Fig. 2), as used in [15]. In the first step, the *modified* distance matrix with all finite-value elements is obtained by using the Dijkstra shortest-path algorithm; if $d_{ij} = \infty$, d_{ij} is assumed to be equal to the length of the shortest path between sensors i and j , which is composed of links of finite distance. Based on the obtained *modified* distance matrix, denoted by D_M , the relative location map is obtained by the method explained in Sect. 3.2. In the second step, the obtained relative location map is used to provide the initial estimates to obtain the final relative location map, using the original distance matrix D . The estimation result and the speed of convergence do not so largely depend on the parameter of convergence criterion, ϵ , in Fig. 2; ϵ can be chosen at very small value, say 10^{-8} †.

3.4 Absolute Location Map from the Relative Location Map

If there are at least three anchors, the relative location map

†The results shown in Sect. 4 are obtained when $\epsilon = 10^{-8}$.

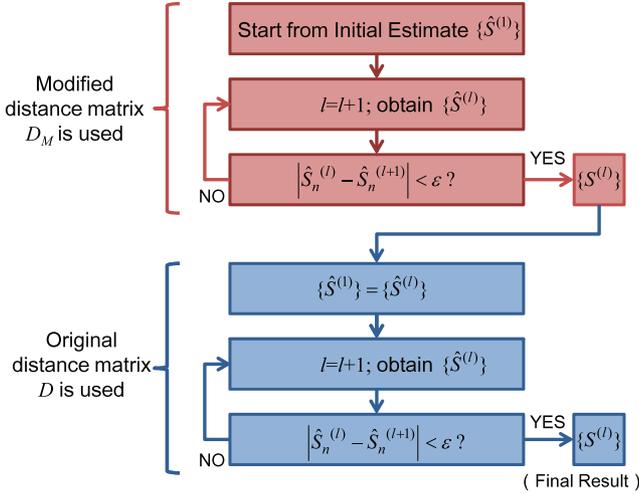


Fig. 2 Algorithm for obtaining the relative location map.

obtained in Sect. 3.3 can be transformed into an absolute location map, in which absolute coordinates are assigned to each of the sensors. This transformation is achieved by translation, rotation, reflection, and scaling, and the algorithm by Horn et al. [18] can be used. The algorithm for the transformation is briefly explained below. Assume that, first n nodes, labeled 1 through n , are anchors whose exact locations are known, and the rest $N - n$ sensors, labeled $n + 1$ through N , are ordinary sensors whose locations are unknown. The estimated location of sensor i when the basic range is equal to d , $\hat{s}_i(d)$, is written in the following form:

$$\hat{s}_i(d) = s_a + d(\hat{s}_i(1) - \hat{s}_a),$$

where s_a is the centroid of real locations of anchors, and \hat{s}_a is the centroid of estimated locations of anchors:

$$s_a = \frac{1}{n} \sum_{i=1}^n s_i, \quad \hat{s}_a = \frac{1}{n} \sum_{i=1}^n \hat{s}_i.$$

The basic range is estimated as

$$d = \sqrt{\frac{\sum_{i=1}^n |\hat{s}_i(1) - \hat{s}_a|^2}{\sum_{i=1}^n |s_i - s_a|^2}}.$$

We then obtain the final location estimates by rotating the location estimates $(\hat{s}_1(d), \dots, \hat{s}_N(d))$ around s_a (and reflecting it if necessary) so as to make $(\hat{s}_{n+1}(d), \dots, \hat{s}_N(d))$ close to (s_{n+1}, \dots, s_N) as much as possible.

Remark 2. We say that sensors i and j are connected by an undirected link if $d_{ij} < \infty$. If there is a set of sensors satisfying the following conditions, we say that the *localizability condition* is satisfied:

- (1) Sensors in the set and links between these sensors compose a connected graph;
- (2) The set includes at least three anchors;

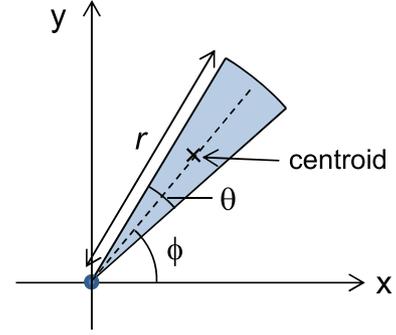


Fig. 3 Centroid of the sensing area shaped as the sector of a circle.

- (3) Each sensor in the set has at least three connected neighbors in the set.

If the localizability condition is satisfied, the locations of sensors in the set can be estimated by the proposed algorithm.

3.5 Bias Correction for Directional Sensing Area

The location estimate of a sensor obtained in the previous section does not give its real location, but the centroid of its sensing area. The location of a sensor is not usually equal to the centroid of its sensing area when the shape of the sensing area is directional (anisotropic), and thus the location estimate should have a bias. Fortunately, the bias can be removed if the shape and the direction of the sensing area are known. For example, if the sensing area of a sensor is shaped as the sector of a circle with radius r , central angle θ , and directional angle ϕ as shown in Fig. 3, then x and y coordinates of centroid of the sensing area, $x_{centroid}$ and $y_{centroid}$, are given below

$$x_{centroid} = \frac{2r}{3\theta}(\sin(\phi + \theta/2) - \sin(\phi - \theta/2)),$$

$$y_{centroid} = \frac{2r}{3\theta}(\cos(\phi - \theta/2) - \cos(\phi + \theta/2)).$$

Thus, if θ , ϕ^\dagger , and r are known, the bias can be removed.

4. Simulation Experiments

4.1 Experiment 1

4.1.1 Simulation Conditions

In the first experiment, sensors are randomly deployed with a density of 0.25, 0.5, or 1 [$1/m^2$] in a square region whose side is 20 m (Fig. 4). Each sensor has a sensing area shaped as the sector of a circle whose central angle θ is 2π , $\pi/2$, $\pi/10$, or $\pi/40$ (Fig. 5). The direction of the sensing area of each sensor is randomly given. The sensors have the common sensing range r ; the simulation is conducted when r

[†]Directional angle can be known if the sensor has an instrument indicating its direction, like compass.

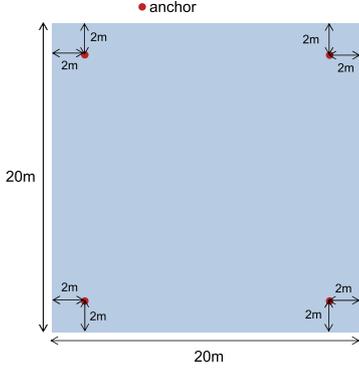


Fig. 4 Region where sensors are deployed.

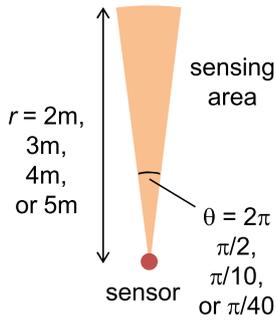


Fig. 5 Sensing area shape.

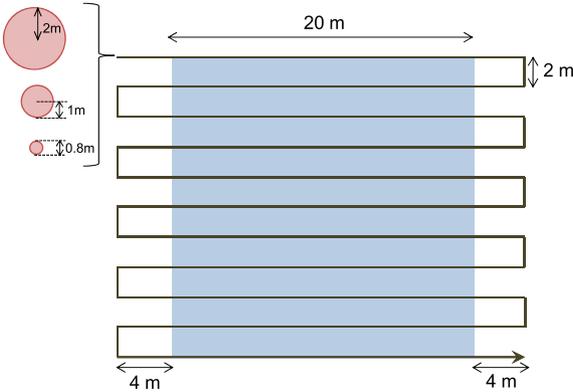


Fig. 6 Trajectory of the object.

is 2 m, 3 m, 4 m, or 5 m. An anchor node is placed at each corner of the region (that is, $n = 4$), as shown in Fig. 4.

A disk-shaped target object moves on the region with the velocity of 1 m/s along the trajectory shown in Fig. 6. Every 2 second, each sensor determines if it can detect the target object. Based on the responses of the sensors to the moving objects, the proposed algorithm is used to estimate the absolute locations of the sensors. Simulation experiments are conducted with ten different patterns of sensor placement.

4.1.2 Results

Figure 7 ($\theta = 2\pi$), Fig. 8 ($\theta = \pi/2$), Fig. 9 ($\theta = \pi/10$), and

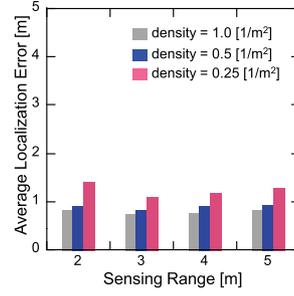


Fig. 7 Average localization error: object radius = 1 m, $\theta = 2\pi$.

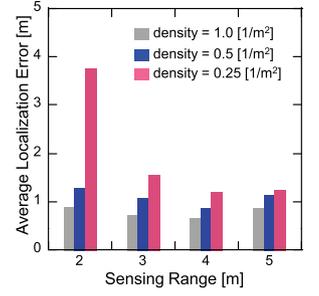


Fig. 8 Average localization error: object radius = 1 m, $\theta = \pi/2$.

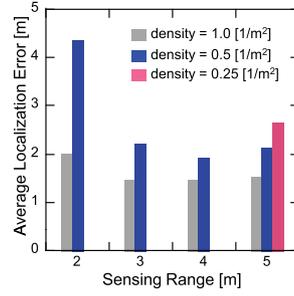


Fig. 9 Average localization error: object radius = 1 m, $\theta = \pi/10$.

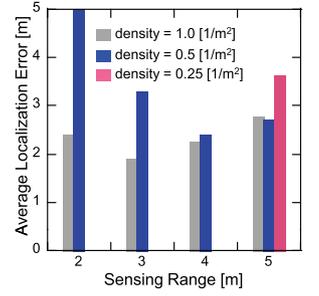


Fig. 10 Average localization error: object radius = 1 m, $\theta = \pi/40$.

Fig. 10 ($\theta = \pi/40$) show the average localization error for different sensor densities and sensing ranges when the object radius is 1 m. The average localization error is defined below.

$$\text{average localization error} = \frac{1}{N-4} \sum_{i=5}^N |\hat{s}_i - s_i|,$$

where nodes labeled 5 through N are ordinary sensors placed at unknown locations. (Nodes labeled 1 through 4 are anchor nodes respectively placed at each corner of the region.) Note that both figures show the results averaged over the ten different patterns of sensor placement. When sensor density is 0.25 [1/m²], sensing range is less than 5 m, and the central angle of a sensing area is $\pi/10$ or less, the localizability condition (see Remark 2 in Sect. 4) is not met and the localization cannot be carried out. The results of such cases are not shown in the figures (see Figs. 9 and 10).

The figures indicate that the estimation error becomes smaller as the sensor density increases. This finding is consistent with the characteristics of general range-based sensor localization [15]. The amount of information (distances between sensors) used for localization roughly increases with N^2 (N is the number of sensors), while the number of parameters (coordinates of sensors) to be known increases with N . Thus, in general, the localization error decreases the sensor density increases.

The figures also show that the estimation error is minimized when the sensing range is 3 m or 4 m. Small sensing range is suitable for locating sensors at a fine granularity, while large sensing range is suitable for collecting the prox-

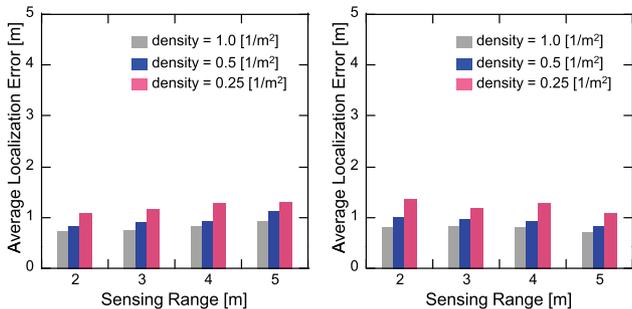


Fig. 11 Average localization error: object radius = 2 m, $\theta = 2\pi$.

Fig. 12 Average localization error: object radius = 2 m, $\theta = \pi/2$.

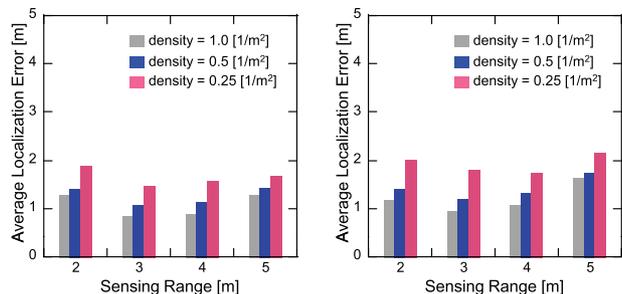


Fig. 13 Average localization error: object radius = 2 m, $\theta = \pi/10$.

Fig. 14 Average localization error: object radius = 2 m, $\theta = \pi/40$.

imity information between sensors. Thus, the best sensing range would exist, depending on the sensor density and the object size. The dependence of localization accuracy on the sensing range is not so significant when the sensor density is sufficiently large.

The central angle θ significantly affects the localization accuracy. When $\theta = 2\pi$, the sensing area has an omnidirectional shape (disk shape) and the localization is the most accurate. As θ decreases, the localization error increases. This is simply because the amount of collected proximity information between sensors decreases as θ decreases. The bias due to the anisotropy of the sensing area also increases as θ decreases, which would deteriorate the accuracy of localization even if the bias due to the anisotropy of the sensing area can be removed as explained in Sect. 3.5.

Figure 11 ($\theta = 2\pi$), Fig. 12 ($\theta = \pi/2$), Fig. 13 ($\theta = \pi/10$), and Fig. 14 ($\theta = \pi/40$) show the average localization error when the object radius is 2 m. The results of localization when the object radius is 2 m are much more accurate than those when the object radius is 1 m. This is because more proximity information between sensors can be obtained as the object size is larger. Using large object is not, however, suitable for locating sensors at a fine granularity, and thus increasing the size of object does not always increase the localization accuracy.

To see the applicability to more practical cases, the results when the object radius is 0.4 m are shown in Fig. 15 ($\theta = 2\pi$), Fig. 16 ($\theta = \pi/2$), and Fig. 17 ($\theta = \pi/10$). The results shown in these figures are somewhat worse than those

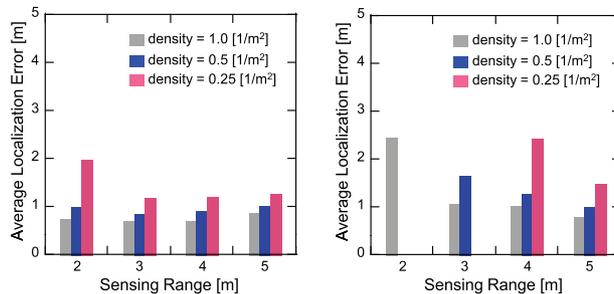


Fig. 15 Average localization error: object radius = 0.4 m, $\theta = 2\pi$.

Fig. 16 Average localization error: object radius = 0.4 m, $\theta = \pi/2$.

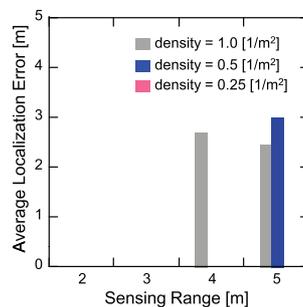


Fig. 17 Average localization error: object radius = 0.4 m, $\theta = \pi/10$.

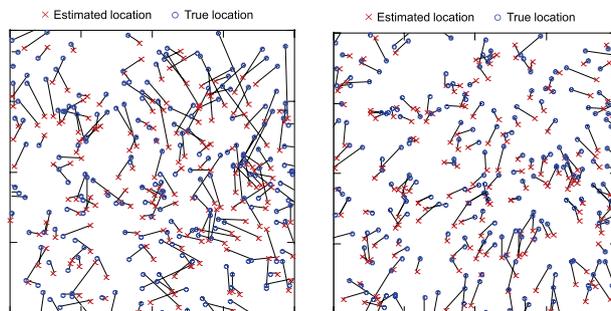


Fig. 18 Estimated and true sensor locations: object radius = 1 m.

Fig. 19 Estimated and true sensor locations: object radius = 2 m.

when the object radius is 1 m. One serious problem is that the localizability condition is hardly satisfied when the target object and the sensing area are both small. This is because, in such cases, the probability that two or more sensors detect the target at the same instant is very small. In fact, when $\theta = 1/40$ and the object radius is 0.4 m, the localizability condition is not met at all in the simulation experiments and thus the figure of the case when $\theta = 1/40$ is not shown in the paper. If the sensing area is disk-shaped, however, the proposed algorithm works well even when the target object is small.

For references, the estimated and true locations of sensors are compared in Fig. 18 (object radius = 1 m) and Fig. 19 (object radius = 2 m), where sensors are deployed with a density of 0.5 [1/m²] and the central angle of the sensing area is $\pi/10$. Both figures show that the estimated locations are consistent with the true locations.

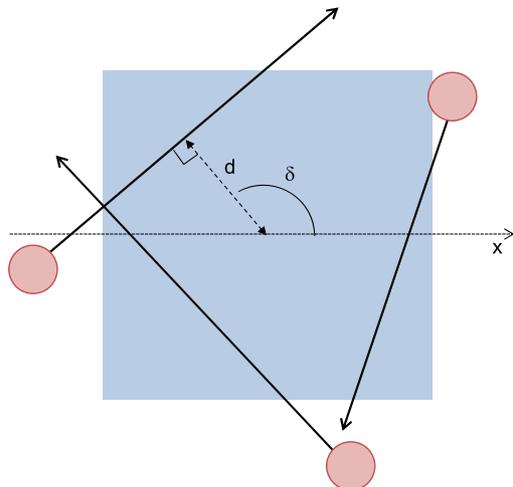


Fig. 20 Trajectory of the object.

4.2 Experiment 2

4.2.1 Simulation Conditions

To see the dependence of the localization accuracy on the trajectory of the object, another set of simulation experiments is carried out in the following condition: several disk-shaped objects traverse the region along straight lines one by one (Fig. 20). The straight-line trajectory of each object is randomly chosen: the distance from the center of region to the trajectory, d ($0 \leq d \leq 10\sqrt{2}$), and the angle with x axis, δ ($0 \leq \delta < 2\pi$), are randomly chosen to determine the trajectory of each object. The conditions of sensor deployments are the same as with those of experiments explained in Sect. 4.1.1[†]. The shape of sensing area is the sector of a circle whose central angle θ is 2π or $\pi/10$, and the sensing range is 4 m. The disk-shaped objects have the common radius, which is equal to 1 m or 2 m.

4.2.2 Results

Figure 21 shows how the average localization error decreases as the number of objects traversing the region increases when the central angle of the sensing area is $\pi/10$. When the number of objects is less than 12, the localizability condition is not met and thus the localization is not possible. After localizability condition is met, the average localization error gradually decreases as the number of objects increases, and it approaches the result when the object moves along the trajectory shown in Fig. 6. Figure 22 shows the results of the cases where the central angle of the sensing area is 2π . There is a rapid drop of the average localization error when the number of objects traversing the region is around 10. After the drop, the localization error still continues to decrease and approaches the result when the object moves

[†]Only one deployment pattern of sensors is used to get results in Experiment 2.

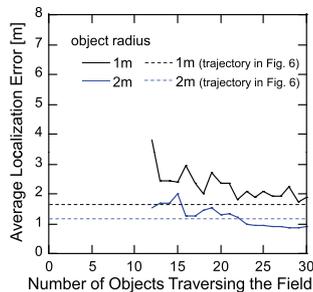


Fig. 21 Transition of localization error: $\theta = \pi/10$, $r = 4$ m.

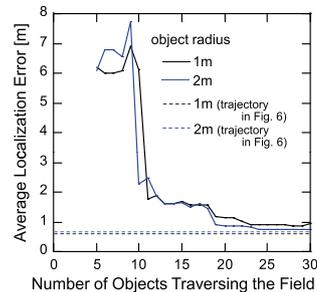


Fig. 22 Transition of localization error: $\theta = 2\pi$, $r = 4$ m.

along the trajectory shown in Fig. 6. In summary, it is the most essential whether the trajectories of objects cover the field to satisfy the localizability condition, and the trajectory itself does not have so large impact on the performance once the localizability condition is satisfied.

5. Conclusion

This paper proposes a simple method for estimating sensor locations based on their responses to moving objects. The proposed algorithm relies only on information about the proximity between the sensors, but the preliminary simulation results indicate that this simple algorithm would have a high potential for accurate sensor localization. The influence of the sensor density, sensing area size, and the target object size on the accuracy of the localization needs to be theoretically investigated, which remains as a future work.

Acknowledgement

We thank the anonymous reviewers for their careful reading of the manuscript and their many insightful comments and suggestions. This work has been supported by the Grant-in-Aid for Scientific Research No. 25330099 of the Ministry of Education, Science, Sports and Culture (MEXT). We are grateful for their support.

References

- [1] H. Wymeersch, J. Lien, and M. Win, "Cooperative localization in wireless networks," *Proc. IEEE*, vol.97, no.2, pp.427–450, 2009.
- [2] L. Doherty, K. Pister, and L. Ghaoui, "Convex position estimation in wireless sensor networks," *IEEE INFOCOM*, pp.1655–1663, 2001.
- [3] R. Moses, D. Krishnamurthy, and R. Patterson, "An auto-calibration method for unattended ground sensors," *IEEE ICASSP*, 2002.
- [4] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz, "Localization from mere connectivity," *Proc. ACM MobiHoc*, pp.201–212, 2003.
- [5] X. Ji and H. Zha, "Sensor positioning in wireless Ad-Hoc sensor networks using multidimensional scaling," *IEEE INFOCOM*, pp.2652–2661, 2004.
- [6] A. Savvides, H. Park, and M. Srivastava, "The bits and flops of the n-hop multilateration primitive for node localization problems," *ACM WSNA*, pp.112–121, 2002.
- [7] K. Langendoen and N. Reijers, "Distributed localization in wireless sensor networks: A quantitative comparison," *Comput. Netw.*, vol.43, no.4, pp.499–518, 2003.
- [8] N. Patwari, J. Ash, S. Kyperountas, A. Hero, R. Moses, and

N. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *IEEE Signal Process. Mag.*, vol.22, no.4, pp.54–69, 2005.

- [9] J. Costa, N. Patwari, and A. Hero, III, "Distributed weighted-multidimensional scaling for node localization in sensor networks," *ACM Trans. Sensor Networks*, vol.2, no.1, pp.39–64, 2006.
- [10] A. Alsindi, K. Pahlavan, B. Alavi, and L. Xinrong, "A novel cooperative localization algorithm for indoor sensor networks," *IEEE PIMRC*, pp.1–6, 2006.
- [11] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low cost outdoor localization for very small devices," *IEEE Pers. Commun. Mag.*, vol.7, no.5, pp.28–34, 2000.
- [12] D. Niculescu and B. Nath, "DV based positioning in Ad Hoc networks," *Telecommunication Systems*, vol.22, no.1–4, pp.267–280, 2003.
- [13] S. Shioda and K. Shimamura, "Relative localization of sensors based on their responses to moving objects," *IEEE MASS*, pp.419–420, 2013.
- [14] H. Saito, O. Kagami, and M. Umehira, "Wide area ubiquitous network: The network operator's view of a sensor network," *IEEE Commun. Mag.*, vol.46, pp.112–120, 2008.
- [15] S. Shioda and K. Shimamura, "Cooperative localization revisited: Error bound, scaling, and convergence," *ACM MSWiM*, pp.197–206, 2013.
- [16] S. Shioda, "Counting with binary proximity sensors based on sensor-cluster identification," *IEEE MASS*, pp.505–506, 2014.
- [17] E. Gansner, Y. Koren, and S. North, "Graph drawing by stress majorization," *Graph Drawing, Lect. Notes Comput. Sci.*, pp.239–250, Springer, 2005.
- [18] B. Horn, H. Hilden, and S. Negahdaripour, "Closed-form solution of absolute orientation using orthonormal matrices," *J. Opt. Soc. Am.*, vol.5, no.7, pp.1127–1135, 1988.

Appendix: Existence of A^{-1}

Define

$$B \stackrel{\text{def}}{=} I - \frac{1}{\rho}A, \quad \rho \stackrel{\text{def}}{=} \max\{c_2, \dots, c_m\},$$

where I is the identity matrix. Observe that for all i

$$\begin{aligned} \sum_j b_{ij} &= 1 + \frac{1}{\rho} \sum_{j=2, j \neq i}^N c_{ij} - \frac{c_i}{\rho} \\ &\leq \frac{1}{c_i} \sum_{j=2, j \neq i}^N c_{ij} - \frac{\rho - c_i}{\rho} \\ &\leq \frac{1}{c_i} \sum_{j=1, j \neq i}^N c_{ij} = 1, \end{aligned}$$

where b_{ij} is the ij -element of matrix B . In particular, if $c_{i1} = 1$ ($d_{i1} < \infty$), then

$$\sum_j b_{ij} < 1.$$

That is, each row sums of B is less than or equal to 1, and at least one of row sums is less than 1. Such a matrix is called strictly sub-stochastic, and it is known that its Perron-Frobenius eigenvalue is less than 1. If A does not have its inverse, there exists a non-zero vector denoted by \mathbf{x} satisfying

$$\mathbf{x}A = \mathbf{0}.$$

Thus, we obtain

$$\mathbf{x}B = \mathbf{x} \left(I - \frac{1}{\rho}A \right) = \mathbf{x}.$$

However, this contradicts the fact that Perron-Frobenius eigenvalue of B is less than 1. Thus, A should have its inverse.



Shigeo Shioda received the B.S. degree in physics from Waseda University in 1986, the M.S. degree in physics from University of Tokyo in 1988, and the Ph.D. degree in teletraffic engineering from University of Tokyo, Tokyo, Japan, in 1998. In 1988 he joined NTT, where he was engaged in research on traffic measurements and controls for ATM-based networks. He moved to Chiba University in 2001. Currently he is a Professor in the Department of Architecture and Urban Science, Graduate School

of Engineering, Chiba University. His current research interests include wireless sensor networks, wireless LANs, peer-to-peer systems, and online social networks. He received Network System Research Award, Information Network Research Award, and Communications Society Distinguished Contributions Award of IEICE respectively in 2003, 2004, 2007, and 2013, and IEEE MASS Best Poster Award in 2013. Prof. Shioda is a member of the ACM, the IEEE, and the Operation Research Society of Japan.