Wireless Sensor Networks
Localization with Isomap

Chengqun Wang, Jiming Chen, Youxian Sun, Xuemin (Sherman) Shen
Date Submitted: 7 June 2009
Date Published: 11 June 2009

Updated information and services can be found at:

These include:

Subject Classification  Vehicular Technology > Mobile Communications

Keywords  localization; sensor network; ISOMAP; Location algorithms and techniques;

Comments  You can respond to this article at:
https://engine.lib.uwaterloo.ca/ojs-2.2/index.php/pptvt/comment/add/491/0

Copyright  Copyright © Date Submitted: 7 June 2009 Chengqun Wang et al. This is an open access article distributed under the Creative Commons Attribution 2.5 Canada License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.
Abstract—This paper studies the problem of determining the sensors’ locations in wireless sensor networks. To alleviate the influence of the noise and the inaccurate measurement in the complicated environment, rather than estimating the pair-wise Euclidean distance between sensors, we use the geodesic distance to measure the dissimilarity between sensors, and employ the isomap algorithm to determine the relative locations of sensors. Given sufficient anchors, the relative locations can be aligned to absolute locations by using coordinate transformation. The coordinate transformation matrix can be obtained by minimizing the sum of squares of the errors between the true locations of the anchors and their transformed locations. Since isomap is parameter-sensitive, we also present an adaptive parameter selection procedure based on the locations of anchors. Simulation results show that the isomap algorithm achieves smaller average location error with little quantity of anchors.

I. INTRODUCTION

Advances in micro electromechanical system technology have made it possible to deploy wireless sensor networks by using inexpensive nodes of low power processor. Many applications are emerging, e.g., wildlife habitat monitoring [1], remote patient monitoring [2], manufacturing [3], environmental monitoring [4]. In these applications, it is necessary to accurately orient the sensors in order to report data which is geographically meaningful. Furthermore, basic middle ware services are often relevant to location information, e.g., geographic routing. In most cases, sensors are not equipped with any GPS-like receiver, or when such a unit is not functional due to environmental difficulties. Therefore, an important problem is to design an accurate and fast converging technique to estimate the sensor locations.

Most current techniques assume that each sensor can measure the distance between itself and each of its neighbors, up to a determined communication range [5]. The localization algorithms estimate the locations of sensors with initially unknown location information by using knowledge of the absolute positions of a small percentage of the sensors and the blurred inter-sensor distance, or build a relative coordinate system when the absolute positions most unavailable [6]. Sensors with known location information are called anchors and their locations can be obtained by deploying anchors at points with known coordinates or by using a global localization system (e.g., GPS). But GPS devices cannot work indoors, they are bulky, expensive and are inefficient in power consumption, while wireless sensor nodes are required to be small, low priced and low powered. Adding a GPS device to all the nodes in a network is not practical. Therefore, most of the sensors are not aware their locations. Those sensors with unknown location information are called non-anchor nodes and their coordinates will be estimated by the localization algorithm.

Localization algorithms can be roughly divided into two categories: distributed algorithms and centralized algorithms. Distributed algorithms extrapolate the unknown sensor location from anchor locations in distributed manner, and assume that the most likely position of a sensor can be recovered by using its neighbors locations [7][8]. Centralized algorithms gather the information (e.g., connectivity, pair-wise distance) about the entire network into one place, where the collected information is processed centrally to estimate the sensors’ locations using mathematical algorithms, such as semidefinite programming (SDP) [9], sum of squares method [10], multidimensional scaling (MDS) [11] and monifold learning techniques [12][13]. Compared with the distributed algorithms, centralized algorithms always obtain excellent location accuracy.

In this paper, we propose a centralized algorithm for wireless sensor networks localization problem. Most of the centralized algorithms require the pair-wise distance between sensors, such as SDP related methods, MDS mapping algorithms. But in most real applications, the environment that most sensors deployed is the anisotropic network and complex terrain, which leads to the estimated pair-wise distance that is not accurately enough, and also leads to most of the localization algorithms fail to perform well [14]. Furthermore, the cumulative measurement error caused by the long distance between two sensors, is a common problem of existing localization algorithms [15][16]. Taking received signal strength (RSS) indication for example, we can easily get the accurate pair-wise distance between sensors in ideal case. Unfortunately, real measurement of RSS can be highly inaccurate due to variability caused by multipath effects and ambient noise interference. Moreover, with the increasing of the pair-wise absolute distance, the probability of RSS blurred will monotonously increase, and the accuracy of estimated pair-wise distance will be decreased [5]. In order to accurately position sensors in anisotropic network and complex terrain and alleviate the noise or measurement error. We employ the geodesic distances to measure the dissimilarity between sensors, and propose a centralized algorithm based on isomap technique [17]. In specific, we first construct the neighborhood graph by using the sensors and their pair-wise distance, and then compute the
geodesic distance of each pair of sensors, finally, construct the 2D embedding and obtain the relative coordinate system of the sensors. If given sufficient anchor nodes, we transform the relative coordinate system to the global coordinate system based on the anchors locations. Moreover, the transformation matrix can be estimated by solving a least-squares optimization problem. Since the performance of isomap is sensitive to the pregiven parameter, in order to alleviate the influence of the parameter, we also propose an adaptive parameter selection procedure based on the true locations of the anchors and their transformed locations.

The remainder of the paper is organized as follows. We first introduce the isomap for wireless sensor networks localization in section II, and then give the the transformation procedure from relative coordinate system to global coordinate system in section III. For isomap, the adaptive parameter selection procedure is given in section IV. Simulation results and the conclusion will be shown in section V and section VI, respectively.

II. LOCALIZATION USING ISOMAP

Isomap is a manifold learning algorithm, which is first proposed for nonlinear dimension reduction, and has been widely used for the analysis of dissimilarity of data set, and can discover the low-dimension spatial structure in data [17]. The key benefit of using isomap for location estimation is that it employs the geodesic distance to measure the dissimilarity between sensors and it can always generate relatively high accurate location estimation even based on the error-prone distance information.

We consider wireless sensor networks of \( m \) sensor nodes \( \{X_1, \ldots, X_m\} \) deployed in a 2D geographic area. Let \( x_i \in \mathbb{R}^2 \) denote the location of the sensor \( X_i \). Without loss of generality, we suppose the first \( n \) nodes are anchor nodes that the locations are known, i.e., the location of the anchor \( X_i \) is \( x_i \), for all \( i = 1, \ldots, n \), where \( n \ll m \). For every pair of sensors \( X_i \) and \( X_j \), \( d(x_i, x_j) \) denotes their Euclidean distance,

\[
d(x_i, x_j) = \left( \sum_{k=1}^{d} (x_{ik} - x_{jk})^2 \right)^{\frac{1}{2}},
\]

where \( d = 2 \) in 2D environment. The local distance problem is to recover the locations of \( X_{n+1}, \ldots, X_m \) based on \( \{d(x_i, x_j)\}_{i,j=1}^{m} \) and \( \{x_i\}_{i=1}^{n} \). If \( n = 0 \), we estimate a relative coordinate system to denote the relative locations of sensors. Note that when sensor \( X_i \) is out of the communication range of sensor \( X_j \), \( d_{ij} \) can’t be obtained directly, we denote \( d_{ij} = \infty \), where \( \infty \) means infinite.

As analyzed in many literatures, the pair-wise distance \( d_{ij} \) is error-prone, especially for faraway sensors [6][18]. Isomap algorithm seeks to preserve the intrinsic geometry of sensors’ location, as captured in the geodesic manifold distances between all pairs of sensors. For neighboring sensors, the Euclidean distance provides a good approximation to geodesic distance. For faraway sensors, geodesic distance can be approximated by adding up a sequence of “short hops” between neighboring sensors. These approximations can be computed efficiently by finding the shortest paths in the graph with edges connecting neighboring sensors. The complete isomap algorithm for localization is described as follows.

1) Collect Euclidean distance matrix \( D \), where \( D = [d_{ij}]_{m \times m} \).

2) Construct neighborhood graph \( G \), define the graph \( G \) over all sensors by connecting sensor \( X_i \) and sensor \( X_j \) if the distance (as measured by \( d_{ij} \)) between \( X_i \) and \( X_j \) is less than \( \epsilon \), or if \( X_i \) is one of the \( K \) nearest neighbors of \( X_j \). Set edge lengths equal to \( d_{ij} \).

3) Compute geodesic distance matrix \( D_G \) over neighborhood graph \( G \). Initialized \( d_G(x_i, x_j) = d(x_i, x_j) \) if sensors \( X_i \) and \( X_j \) are linked by an edge; \( d(x_i, x_j) = \infty \) otherwise. For each value of \( l = 1, \ldots, m \) in turn, replace all entries \( d_G(x_i, x_j) \) by \( \min \{d_G(x_i, x_j), d_G(x_i, x_l) + d_G(x_l, x_j)\} \). The final matrix \( D_G = [d_{Gij}]_{m \times m} \) is the geodesic distance matrix which contains the shortest path distances between all pairs of sensors in the neighborhood graph \( G \).

4) Apply double centering to the geodesic distance matrix \( D_G \). Let \( H = I - \epsilon e e^T / m \), where \( e \) is all ones \( m \)-dimensional column vector, \( I \) is \( m \)-dimensional identity matrix. Then \( J = -\frac{1}{2} HSH \), where \( S_{ij} = D_G^2 \).

5) Compute the eigen-decomposition \( J = UV^T \).

6) Construct \( d \)-dimensional embedding. \( d = 2 \) is 2D case. We denote the matrix of largest \( d \) eigenvalues by \( V_d \) and \( U_d \) the first \( d \) column of \( U \). The \( d \)-dimensional relative coordinate matrix \( Y \) is equal to \( U_d V_d^2 \). \( Y \) contains the relative coordinate of all sensors.

7) Aligning Relative Locations. Given sufficient anchor sensors (3 or more for 2D, 4 or more for 3D), transform the relative coordinate matrix \( Y \) to an absolute coordinate matrix \( X \) based on the locations of anchors.

Patwari et al briefly mentioned that the manifold learning techniques (including isomap) can be used for localization problem under the spatially correlated sensor model [12]. Generally, this algorithm is similar to the classic algorithm MDS-MAP [11], because isomap can be considered as a geodesic distance version of the MDS [17]. Instead of using the Euclidean distance for embedding, isomap considers the geodesic distance on a weighted neighboring graph, as show in step 2 and step 3. We assume that when the absolute distance between sensor \( X_i \) and sensor \( X_j \) is less than \( \epsilon \), then the estimated distance approximates the absolute distance well, or when sensor \( X_i \) is among the \( K \) nearest neighbors of sensor \( X_j \), then the estimated distance approximates the absolute distance well. So we can construct a neighborhood graph \( G \) to represent the low error-prone distance estimation (as shown in step 2), and then calculate the geodesic distance between sensors (as shown in step 3).

The procedure of the shortest path calculation in step 3, is also known as Floyd’s algorithm. It requires \( O(m^3) \) operations. More efficient algorithms exploiting the sparse
structure of the neighborhood graph $G$ can be found in [19].

Except the geodesic distance, the combination of step 4, step 5 and step 6 is the classic metric-MDS. The core of classical MDS is singular value decomposition (SVD) (as shown in step 5), which takes $O(m^3)$ operations. The result of SVD (as shown in step 6) is the relative coordinate matrix that contains locations for all sensors. Although these relative locations may be accurate relative to one or two anchors, the entire coordinate will be arbitrarily rotated and flipped relative to the true sensors locations.

The previous 6 steps give an efficient way to calculate the relative coordinate of all sensors. If given sufficient anchors, the relative coordinate can be transformed through linear transformations which include scaling, reflection, and rotation. The goal is to map the relative coordinates to physical coordinates based on anchors. We will discuss this procedure in the next section.

Note that our algorithms also can be extended when geographic area is more than 2D environment. The difference is the distance calculation (parameter $d$ in Eq. (1)) and the aligning relative locations (parameter $d$ in step 7). By simply set $d = 3$, this algorithm is available for 3D environment.

III. ALIGNING RELATIVE LOCATIONS

In most applications, it is desired to compute the absolute locations of the sensors eventually, so it’s necessary to align the relative location to absolute location with the aid of anchors. The alignment usually includes scaling, shift, rotation and reflection of the coordinate, i.e., system coordinate transformation. Given $m$ sensors, without loss of generality, let the first $n(n \ll m)$ nodes be anchors. For 2-dimensional environment, at least 3 anchors’ absolute locations are needed in order to identify the absolute locations of the non-anchors.

Let $Y = [y_1, \ldots, y_m]$ denote the relative locations of all sensors, $Y_a = [y_1, \ldots, y_n]$ denote the relative location of anchors. $X = [x_1, \ldots, x_m]$ denotes the true locations of sensors, and $X_a = [x_1, \ldots, x_n]$ denotes the true locations of the anchors. We assume the coordinate transformation is give by

$$y_i = Ax_i + b \quad \text{for all } i = 1, \ldots, m \quad (2)$$

The goal is to minimize the sum of squares of the errors between the true locations of the anchors and their transformed locations. The bias $b$ in Eq. (2) can be eliminated by considering the relative distance between sensors. For a given anchor $u$, we set $\Delta X_a = [\Delta x_1, \ldots, \Delta x_{u-1}, \Delta x_{u+1}, \ldots, \Delta x_n]$, and $\Delta x_i = x_i - x_u$, and $\Delta Y_a = [\Delta y_1, \ldots, \Delta y_{u-1}, \Delta y_{u+1}, \ldots, \Delta y_n]$, where $\Delta y_i = y_i - y_u$. The alignment problem can be formulated as a least-squares optimization problem as follows

$$\min_{A,b} \sum_{i=1,i\neq u}^n \|A\Delta y_i - \Delta x_i\|^2 \quad (3)$$

The solution of above least-squares problem can be reduced to solve a set of linear equations

$$A(\Delta Y_a \Delta Y_a^T) = \Delta X_a \Delta Y_a^T \quad (4)$$

so we have the analytical solution $A = \Delta X_a \Delta Y_a^T (\Delta Y_a \Delta Y_a^T)^{-1}$. The above problem can be solved with high accuracy, reliability and approximately $O(n^2)$ complexity [20]. Finally, the absolute locations of non-anchors can be estimated by

$$\tilde{x}_i = A(y_i - y_u) + x_u \quad \text{for all } i = n, \ldots, m \quad (5)$$

where $\tilde{x}_i$ is the estimation of $x_i$. The performance of the algorithm can be measured with the mean location estimation error, which is widely used in previous research works.

$$\text{error} = \frac{\sum_{i=n+1}^m \|\tilde{x}_i - x_i\|^2}{m-n} \quad (6)$$

where $\| \cdot \|^2$ denotes the 2-norm. A low error means good performance of the evaluated method.

IV. ADAPTIVE PARAMETER SELECTION

To construct the neighborhood graph $G$ for isomap algorithm, we must preclude a parameter $K$ or $c$, the performance of isomap is sensitive to these parameters. Let $K$ be nearest neighbors. In this section, we will present an adaptive parameter selection procedure based on the true locations of the anchors and their transformed locations. $Y_a$ is the relative locations of anchor, and $X_a$ is the true location of anchors. By using Eq. (5), we obtain the estimated absolute locations of anchors $X_a$. In most of the cases, this estimated locations of anchors can not match the true locations perfectly. There is an error. Let

$$\text{error}_K = \frac{\sum_{i=1}^n \|\tilde{x}_i - x_i\|^2}{n} \quad (7)$$

be the alignment error of anchors with respect to the given $K$.

The adaptive parameter selection procedure is described as follows.

1) Given $\hat{K} = [K_1, \ldots, K_v]$ be the dictionary of parameter $K$. For each $K_i$, we calculate its corresponding alignment error $\text{error}_{K_i}$, where $i = 1, \ldots, v$.

2) We choose the $K_i$ with respect to the minimum $\text{error}_{K_i}$ as the optimal parameter of isomap for localization.

V. EXPERIMENT RESULTS

We simulate our localization algorithms with Matlab. To evaluate the performance of our algorithm, we ran isomap on various topologies of networks. The sensors are uniformly, or randomly placed in a 2D square region. We also consider the errors of the neighboring sensor distance estimation. The measurement error is in the range 0% - 40% of the average radio range, randomly distributed. Moreover, we compare our algorithm with the classic MDS-MAP algorithm proposed in [11], the dictionary of parameter $K$ is chosen $\hat{K} = [5, 6, \ldots, 15]$. All of the reported results are the average ones over 100 trials.
A. Uniformly Placement

In this set of experiments, 256 sensors are uniformly placed in a 200-by-200 units square region, and the average radio range is 25 units. The anchors are uniformly placed too. Fig. 1 shows the result of isomap with the adaptive parameter selection procedure (ISOMAP(Adaptive)), where relative location is transformed based on 16 anchors. The squares are anchors and the circles denote the non-anchors. Each line connects a true sensor location and its estimation. The average estimation error in this experiment is about 4.1207 units.

We also present a quantitative analysis of the effects of ISOMAP(Adaptive) with respect to different number of anchors, and compare ISOMAP(Adaptive) with MDS-MAP and isomap with different parameters $K$ ($K = 6, 7, 8$) in Fig. 2. While Fig. 2(a) shows the average location error of different algorithms, Fig. 2(b) shows the standard deviation of the location error.

B. Randomly Placement

In this group of experiments, 256 sensors are randomly placed in a 200-by-200 units square region, and the average radio range is 25 unit. The anchors are uniformly placed. We present the final solution of ISOMAP(Adaptive) based on 16 anchors in Fig. 3. The squares are the anchor nodes, as well as the circles denote the non-anchors. Each line connects a true sensor location and its estimation. The average location error is about 7.7713 units. Fig. 4 shows statistics on location error under different number of anchors. The average location error and the standard deviation of the location error are shown in Fig. 4(a) and Fig. 4(b), respectively.

Several observations can be made from these simulation results. First, compared with MDS-MAP, isomap performs better when we carefully choose the parameter $K$. However, by considering an adaptive parameter selection procedure, the isomap algorithm–ISOMAP(Adaptive)–performs more accurate than the fixed parameter isomap. Second, the average location error is always decreasing with the increasing of anchors, as shown in Fig. 2(a), Fig. 4(a). Third, a small number

Fig. 1. Estimation for the uniform network with 16 anchors.

Fig. 2. Simulation Results of the uniform deployed sensor network.

Fig. 3. Estimation for the random network with 16 anchors.
of anchors should be enough to get a sufficient estimation error by using ISOMAP(Adaptive).

VI. CONCLUSION

In this paper, we have proposed the isomap algorithm for localization problem in wireless sensor networks. Isomap employs the geodesic distance to measure the dissimilarity between sensors. Isomap first uses the pair-wise distance to construct the neighborhood graph, then computes the geodesic distance of all pair-wise sensors, and finally constructs the 2D relative coordinates of the sensors by using the metric-MDS. While given sufficient anchors, the relative coordinates can be transformed to the absolute locations of sensors. The transformation matrix can be obtained by solving a least-squares optimization problem under the goal of minimizing the sum of squares of the errors between the true locations of the anchors and their transformed locations. Isomap is parameter-sensitive. To alleviate the parameter influence of isomap, we have also proposed an adaptive parameter selection procedure based on the locations of anchors.

Our future work includes evaluating our algorithm for other networks. In addition, We will implement this algorithm in a real prototype and investigate its applications.

ACKNOWLEDGMENT

This work is partially supported by NSFC-Guangdong joint Project under grants U0735003; NSFC under grants No.60604029, 60702081; NSF of Zhejiang Province under grants No.Y106384; and 863 high-tech Project 2007AA041201.

REFERENCES