Research Article

A Graph Embedding Method Based on Sparse Representation for Wireless Sensor Network Localization

Xiaoyong Yan,1,2,3 Aiguo Song,1,2 and Hao Yan4

1 School of Instrument Science and Engineering, Southeast University, Nanjing 210096, China
2 Remote Measurement and Control Key Lab of Jiangsu Province, Nanjing 210096, China
3 School of Intelligence Science and Control, Jinling Institute of Technology, Nanjing 211169, China
4 School of Software Engineering, Jinling Institute of Technology, Nanjing 211169, China

Correspondence should be addressed to Aiguo Song; a.g.song@seu.edu.cn

Received 18 December 2013; Revised 1 June 2014; Accepted 2 June 2014; Published 23 July 2014

Academic Editor: Ki-Il Kim

Copyright © 2014 Xiaoyong Yan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In accordance with the problem that the traditional trilateral or multilateral estimation localization method is highly dependent on the proportion of beacon nodes and the measurement accuracy, an algorithm based on kernel sparse preserve projection (KSPP) is proposed in this dissertation. The Gaussian kernel function is used to evaluate the similarity between nodes, and the location of the unknown nodes will be commonly decided by all the nodes within communication radius through selection of sparse preserve projection self-adaptation and maintaining of the topological structure between adjacent nodes. Therefore, the algorithm can effectively solve the nonlinear problem while ranging, and it becomes less affected by the measuring error and beacon nodes quantity.

1. Introduction

For most wireless sensor network (WSN) applications, it is essential to know the location of sensor nodes. Related researches show that, in the information related to the context of monitoring deployment area provided to the users, more than 80% is related to the information of location [1].

The self-localization algorithm of nodes can be divided into the two major categories: range-based and range-free [2, 3]. Range-based approaches have a relatively higher accuracy, but this final location estimation accuracy depends on the measuring accuracy. The most common measuring technologies include Radio Signal Strength (RSSI) [4], Time of Arrival (ToA) [5], Time Difference of Arrival (TDoA) [6], and Angle of Arrival (AoA) [7]. The measurement data of RSSI can be achieved during each data communication between nodes, and it does not occupy additional bandwidth and energy, so the hardware expense is relatively simple and cheap. These turn it into a hotspot research direction to use the RSSI measurement data to conduct localization. However, in a complicated monitoring environment, the measurement of RSSI signal is also affected by multiple factors; for example [8–10], the communication between nodes generally uses free common channel, which is inevitably interfered by other devices in the monitoring area; RSSI signal itself has multipath feature; the hardware of sensor node is relatively cheap, simple, and with poor operational capability, and in addition, the unstable production technique has also caused unstable quality of certain node products, blockage by static or moving obstacles in the monitoring area, and so forth. All these will provoke a large amount of uncertain factors in the collected signals and make the signal present nonlinearity.

Figure 1 shows the cause of nonlinearity in a complex environment. When the part between node A and node C is interfered by other devices of the same frequency band, or if it is blocked by a static or moving object, the measured value is generally bigger than the actual value. If only linear method is used without considering the actual environment, and if the empirical model is directly used to obtain the measuring distance, the measuring precision is generally very low, which obviously cannot satisfy the actual application. Therefore, there are multiple difficulties to face in order to obtain accurate localization, which requires exploring new
technologies and methods or combining other solutions to handle it.

The rest of the paper is structured as follows. In Section 2, we address the related work in the area of range-based localization schemes. In Section 3, we give a brief review of related concepts. And Section 4 presents our localization method. Section 5 shows simulated results, and Section 6 gives the conclusions.

2. Related Work

Related literature [11] generally divides the localization algorithm based on signal strength ranging into two types: one type uses empirical formula transformation to obtain the distance after obtaining the signal strength between nodes and then uses the trilateration method or multilateral method to obtain the location of unknown node; another type directly uses the signal strength information between nodes to obtain the similarity between nodes through the machine learning method, and then, the relation between nodes is mined in accordance with the similarity and the location of beacon node. The former depends on the empirical model, and classic methods include the RADAR system [12] and SpotON system [13]. They only can be applied to some scenarios with relatively unchanging environment, and in order to obtain the localization result with high accuracy, training calibration should be conducted to different environments, which might cost a large amount of manpower and material resources. If the fitting model is not accurate or the deployment scenario is modified, then the original model cannot reflect the relation between the Euclidean distances and RSSI values, the distance measurement accuracy is low, and it will result in undesirable localization effects. The latter regards the nodes in network as independently distributed devices, uses the RSSI measurement values between adjacent nodes, refers to the machine learning algorithm, trains and learns a prediction model in the deployment area, and further estimates the location information of unknown nodes in the area, such as the LANDMARC localization system based on k nearest neighboring (k-nn) [14], kernel principal component analysis (KPCA) localization algorithm [15], kernel canonical correlation analysis (KCCA) location estimation method [16, 17], and localization algorithm based on kernel locality preserving projection (KLPP) [18]. The range-based localization technology based on machine learning method is not sensitive to RSSI measurement error and it has the characteristics of low requirement for measurement technology and high self-adoption degree; therefore, it has attracted considerable attention.

The k-nn method [18] relies on the “nearest adjacent distance criteria” weighted idea to obtain the information of unknown nodes, its algorithm is a kind of linear algorithm, the k value is generally artificially set, and it has high randomness. In addition, when the nodes are assigned to a complicated environment, the RSSI measured values have a high nonlinearity, which would lead to poor localization result. For nonlinear data, the researchers find that it is a practical and effective solution to use the kernel method to build the mapping model. The kernel method [19, 20] maps the original data into an appropriate high-dimensional feature space through a certain suitable kernel function, which transfers the nonlinear problem difficult to solve in the original space into a linear problem in the feature space. By referring to the characteristics of the kernel method, this paper uses it to solve the nonlinear problem in the RSSI measured data.

A large number of related researches show that after kernel function and training set are given, the kernel matrix (or Gram matrix) can be built, and the real internal structure of data can be disclosed in accordance with the similarity provided by the kernel matrix. The KPCA and KCCA localization algorithms taking into account this theory have been proposed successively.

KPCA is an extension of principal component analysis (PCA) [21] using techniques of the kernel method. Using kernel method, the nonlinear data is mapped into a high-dimensional feature space and then PCA is employed in feature space. The linear PCA in the high-dimensional feature space can best represent the projection of original data in order to realize dimensionality reduction and denoising of data. KCCA is also a kind of dimensionality reduction method similar to KPCA, and different from KPCA; by building the mapping of signal space and physical space in the feature space, the localization method based on KCCA makes it possible for the nodes to use RSSI values to deduce back the relative topological structure of their physical space and then calculate the location of these nodes. However, both of the KPCA and KCCA computations have adopted a global nonlinear mapping algorithm, which is simple and highly efficient, and although it can obtain good results under some situations, it has ignored the data distribution and failed to consider that the data have the local distribution feature. In addition, the KCCA algorithm is only applicable to fingerprint-based database for indoor localization, and it requires artificially collecting the training datum and building the distribution diagram of the relation between RSSI and Euclidean distances beforehand. During localization, in accordance with the distribution diagram obtained during previous training, k nearest access points (APs) will be found, and their center of mass will be used as the estimated location, and if the APs density is not high enough, it requires further iterative operation. Therefore, the KCCA estimation algorithm does not apply to the random
deployment environment or any scenarios that cannot be reached by human.

Based on that, Wang et al. [18] used the KLPP algorithm and proposed the KLPP-based localization algorithm. The KLPP localization method uses the kernel function to measure the similarity between nodes, and after the RSSI values are mapped into the feature space, it uses the LPP algorithm [22] to construct the adjacency graph between nodes, which formulates the localization problem as a graph embedding problem and allows us to consider the topological structure of the networks. The KLPP-based localization method uses the kernel method to solve the nonlinear problem of RSSI values; by using the adjacency graph, the location of unknown nodes is commonly decided by their adjacent nodes, so the measurement error caused by remote beacons during the process of traditional multilateral or trilateral method is avoided, and the impact by the number of beacon nodes is small.

However, the KLPP algorithm is overly dependent on the adjacency graph, while the construction of the adjacency graph mainly use \( k \)-nn and \( \varepsilon \)-ball approaches [23]. The \( k \)-nn and \( \varepsilon \)-ball methods are the two most popular ones for graph construction in the literature. The \( k \)-nn graph method chooses \( k \) nearest adjacent samples, while the \( \varepsilon \)-ball method chooses the samples which falling into its \( \varepsilon \)-ball neighborhoods are linked to it. In addition, the weight of side in the graph is generally obtained through methods like binary, Gaussian kernel, or \( \ell_p \)-reconstruction [24–26]. All of them determine the neighboring data based on the pairwise Euclidean distance, which is, however, very sensitive to data noises, especially in high-dimensional spaces; that is, the graph structure may dramatically change the presence of noise. The construction of the adjacency graph and the choice and setting of weight all restrict the final localization result of KLPP-based algorithm. For the localization algorithm, once the adjacency graph between nodes is given, in the subsequent operation, the adjacency graph parameters might have no change at all. However, for the WSN, its deployment area is generally affected by various actual factors such as hardware error, network attacks, and lack of energy and severe weather, which causes that the network topology can change at any time. Therefore, it is difficult for the KLPP-based localization method with preset parameters to adapt to a complicated environment, and as showed in Figure 1, when it is affected by the interference source, the structure in the figure will be affected in a certain degree. How to self-adaptively determine the relation between nodes is the foundation to obtain satisfying location estimation result. Researchers [27] found that sparse representation (SP) has natural discrimination ability, and through construction of \( \ell_1 \)-graph, each sample can automatically choose the adjacent samples through SP method and automatically make each sample be represented by the combination of surrounding training samples. In addition, \( \ell_1 \)-graph is robust owing to the overall contextual \( \ell_1 \)-norm formulation and the explicit consideration of data noises. Qiao et al. [28] smartly applied SP to the structure of the adjacency graph and proposed the sparsity preserving projections (SPP) algorithm. This algorithm uses the sparsity of coefficient in SP as natural differentiation information and introduces it into the reconstructed adjacency graph, through “sparse” constraint, the local structure feature of data will be self-adaptively captured, and the sample will be automatically linearly represented by the samples in its adjacent area. Then, Yin and Yang [29] use the kernel method to expand SPP to the nonlinear area, and through mapping of the kernel method, the coefficient of SP is used to construct the adjacency graph in the feature space. Because the kernel method has stronger discrimination information than the original SP method, therefore, KSPP has more efficient discrimination ability than SPP. Inspired by the KSPP algorithm mentioned above, during a study on the localization problem, this paper proposes the KSPP-based localization algorithm, called location estimation-KSPP (LE-KSPP). The algorithm will obtain the signal strength between nodes and measure the similarity between nodes through Gaussian kernel function, then, the network adjacency graph will be self-adaptively constructed through SP, the localization problem will be transformed into a graph embedding problem, the coordinate of unknown code will be determined by all the nodes within the communication radius, and in this way, the impact of measurement error and the beacon node number will be reduced. The experiment and simulation results show that the LE-KSPP method has high localization accuracy, the impact of beacon node number is small, and it has strong environment adaptability, which apply to different deployment environments.

Figure 2(a) shows traditional multilateration or trilateration localization, whose measurements are made only between an unknown location sensor and known location sensors. When the location of beacon nodes is relatively far, the measurement error will definitely be bigger; in addition, the algorithm does not consider the impact of other nodes surrounding this node, so the location obtained through traditional method has a low location estimation accuracy. Figure 2(b) shows the LE-KLPP method; if its \( k \)-nn graph only chooses three pairs of sensors, then this algorithm has obviously ignored the impact of some other nodes in the adjacent area, so the location estimation result is not ideal. Figure 2(c) shows the LE-KSPP method, because each node can automatically obtain the neighbors with information through SP. The information of neighbors can be utilized to aid in the location estimation and enhances the accuracy and robustness of the localization system.

3. Related Concepts

3.1. Kernel Method. The kernel method is one of the current hot topics in the field of artificial intelligence and machine learning, and it is based on statistical learning theories and kernel technology. As early as 1964, during research of potential function, Aizerman et al. [30] introduced the idea of kernel function as the inner product of feature space into learning field. However, it was until 1992 when this idea attracted the attention of Boser et al. [31]; they combined it with large margin hyperplane to generate support vector machine (SVM), and since then, the concept of kernel method has become one of the mainstream directions of machine learning literatures. Kernel method using the feature
function $\phi$ embeds data into the feature space, so that the nonlinear data are presented linear model in such space, as showed in Figure 3.

In accordance with Figure 3, we can see that the basic idea of kernel method is to map the random vector datum in the input space into a higher feature space by using nonlinear function, and then, the linear learning algorithms are designed in this higher-dimensional feature space. The kernel method generally uses a kernel function to pack the nonlinear relation between input space and output space, and generally speaking, the solutions of any kernel methods consist of two parts, that is, a module and a learning algorithm. The module executes the process of mapping into the embedding or feature space, while the learning algorithm is used to find out the linear patterns in that space. The modularity of kernel methods proves its reusability as the learning algorithm. The matching algorithm can be combined with any kernel function, so it can be used in any data domain. The kernel component is for specific data, but it can be set with different algorithms to solve the task within the full range that we consider. Figure 4 shows the stages involved in the application of the kernel method.

The kernel function is a potential nonlinear and parameter equation of input variable. The kernel function relies on the input and output variables to realize control of parameters, and for the localization algorithm, the input variable is the input RSSI values matrix, and the output variable is relative coordinate of nodes. Therefore, the key of machine learning is estimation of parameter based on known input and output data. Consider there exists such a mapping, and assume a training sample set $S = \{(x_1, y_1), \ldots, (x_l, y_l)\}$, in which $x_i \in \mathbb{R}^n$, and the corresponding label is $y_i \in \mathbb{R}^m$.

**Definition 1.** $\phi$ is a mapping from $x$ to an (inner product) feature space $F$:

$$
\phi : x \in \mathbb{R}^n \mapsto \phi(x) \in F \subseteq \mathbb{R}^N.
$$

The purpose of the map $\phi$ aims to transform the nonlinear relation into a linear relation. This kind of change to the input space can generally increase the learning efficiency, but in addition to enriching the function expressive ability, a higher-dimensional feature space can also increase the amount of computation, which correspondingly reduces the generalization ability of learning algorithm. Therefore, we need an implicit way to complete the data change process, and
in the kernel method, this kind of direct computation process is called kernel function.

**Definition 2.** A kernel is a function $\kappa$ that for all $x, z \in X$ satisfies

$$\kappa(x, z) = \langle \phi(x), \phi(z) \rangle,$$

where $\langle \cdot, \cdot \rangle$ refers to the inner product. The definition of kernel function $\kappa(x, z)$ is to calculate the inner product of two data vectors under the mapping of nonlinear transformation $\phi(\cdot)$, while $\phi$ is a mapping of $X$ to an (inner product) feature space $F$:

$$\phi: x \mapsto \phi(x),$$

where $x$ is the input space and $\phi(x)$ is the feature space.

Kernel function can use the inner product as the direct function of the input space, it can calculate the inner product more efficiently, which makes the feature space with index dimension or even infinite dimension possible, and it does not need to explicitly calculate the mapping $\phi$. In other words, the kernel method uses a predefined kernel function to express the inner product of two sample vectors in the feature space, and it does not need to directly realize nonlinear mapping of the sample, so during use, it does not need to know the specific form of nonlinear mapping.

During actual application, there are three kinds of common kernel functions: polynomial kernel function, sigmoid kernel function, and Gaussian kernel function (also called radial basis kernel function) [19]. Of these, the Gaussian kernel function has the characteristic of maintaining the distance similarity of input function; this paper chooses the Gaussian kernel function to calculate the similarity between nodes, and its definition is as follows:

$$\kappa(x_i, x_j) = \exp \left( -\frac{\|x_i - x_j\|^2}{2\sigma^2} \right),$$

3.2. Kernel Sparsity Preserving Projections. First of all, KSPP uses the kernel method to map the data into a higher feature space, which makes it linearly separable in the higher feature space; then, it uses the SPP method to construct the adjacent matrix of sample data in the higher-dimensional feature space; finally, the adjacent matrix is used to conduct feature extraction through the graph embedding method.

Assume there is a set of training sample $X = \{x_1, x_2, \ldots, x_n\} \in \mathbb{R}^{M \times n}$; firstly, use the nonlinear mapping function $\phi$ to map the training samples into a higher-d feature space to obtain $\Phi = [\phi(x_1), \phi(x_2), \ldots, \phi(x_n)]$, and then, similar to the SPP approach, use the improved $\ell_1$ optimization to reconstruct the feature space samples of each sample $\phi(x_i)$ and obtain its corresponding weight coefficient $s_i$. Therefore, the optimization problem can be expressed as

$$\begin{align*}
\min_{s_i} & \quad \|s_i\|_1 \\
\text{s.t.} & \quad \phi(x_i) = \Phi s_i \\
& \quad 1 = 1^T s_i,
\end{align*}$$

where $s_i$ refers to the coefficient vector of SP in the feature space; $s_{ij}$ ($j \neq i$) refers to the contribution amount of the training sample $\phi(x_i)$ to the reconstructed $\phi(x_j)$ in the feature space. Similarly, higher-dimensional feature space also has noise, and in order to obtain the solution to the optimization, we need to relax restrictions on the optimization and obtain

$$\begin{align*}
\min_{\tilde{s}_i} & \quad \|\tilde{s}_i\|_1 \\
\text{s.t.} & \quad \|\phi(x_i) - \Phi \tilde{s}_i\| \leq \varepsilon \\
& \quad 1 = 1^T \tilde{s}_i,
\end{align*}$$

Because the mapping relation $\phi$ is unknown, $\Phi$ and the sample $\phi(x_i)$ in the higher feature space are also unknown, and therefore, the formula mentioned above cannot be directly solved. In accordance with (6), the optimization problem can be transformed to

$$\tilde{s}_i = \arg \min_{s_i} \|s_i\|_1$$

$$\begin{align*}
\text{s.t.} & \quad \|\Phi^T \Phi \tilde{s}_i - \Phi^T \phi(x_i)\| \leq \varepsilon \\
& \quad 1 = 1^T \tilde{s}_i.
\end{align*}$$

At this moment, the formula can be solved to obtain the estimated vector of SP coefficient, $\tilde{s}_i$ ($i = 1, 2, \ldots, M$), and its combination is used to obtain the sparse reconstructed matrix $S = [\tilde{s}_1, \tilde{s}_2, \ldots, \tilde{s}_M]$. At this moment, the kernel SPP objective function is transformed to

$$\min_{w} \sum_{i=1}^{M} (w^T \Phi (x_i) - w^T B \tilde{s}_i)^2.$$  

Through deduction similar to SPP, we obtain the optimization criteria for KSPP; that is,

$$\min_{w} \frac{w^T \Phi (1 - S - S^T + SS^T) \Phi^T w}{w^T \Phi \Phi^T w}.$$  

Then, the criterion is transformed to a solution of the generalized characteristic equation:

$$\Phi (S + S^T - SS^T) \Phi^T w = \lambda \Phi \Phi^T w.$$

Left-multiply $\Phi^T$ to the two sides of equation, and we obtain

$$\Phi^T \Phi (S + S^T - SS^T) \Phi^T w = \lambda \Phi^T \Phi \Phi^T w.$$  

$w$ can be expressed as

$$w = \Phi p.$$  

The generalized characteristic equation can be simplified to

$$K (S + S^T - SS^T) K p = \lambda K^2 p.$$  

Solve the eigenvectors of the first $d$ corresponding maximum eigenvalues $p_i$ ($i = 1, 2, \ldots, d$).
$K$ is the inner product of data in the feature space calculated by kernel function; that is,

$$\Phi^T \Phi = \left[ \phi(x_1), \phi(x_2), \ldots, \phi(x_M) \right]^T \cdot \left[ \phi(x_1), \phi(x_2), \ldots, \phi(x_M) \right]$$

$$= \begin{bmatrix} K(x_1, x_1) & K(x_1, x_2) & \cdots & K(x_1, x_M) \\ K(x_2, x_1) & K(x_2, x_2) & \cdots & K(x_2, x_M) \\ \vdots & \vdots & \ddots & \vdots \\ K(x_M, x_1) & K(x_M, x_2) & \cdots & K(x_M, x_M) \end{bmatrix} = K,$$

$$\Phi^T \phi(x_i) = \left[ \phi(x_1), \phi(x_2), \ldots, \phi(x_M) \right]^T \phi(x_i)$$

$$= \begin{bmatrix} K(x_1, x_i) \\ K(x_2, x_i) \\ \vdots \\ K(x_M, x_i) \end{bmatrix}. \tag{14}$$

After KSPP projection, the $i$th new eigenvector is

$$y_i = w_i^T \phi(x) = \left( \frac{Bp_i}{\sqrt{p_i^T K p_i}} \right)^T \phi(x) = \frac{p_i^T B^T \phi(x)}{\sqrt{p_i^T K p_i}}$$

$$= \frac{p_i^T}{\sqrt{p_i^T K p_i}} \begin{bmatrix} K(x_1, x) \\ K(x_2, x) \\ \vdots \\ K(x_M, x) \end{bmatrix}. \tag{15}$$

The total KSPP eigenvector is

$$y = p^T \begin{bmatrix} K(x_1, x) \\ K(x_2, x) \\ \vdots \\ K(x_M, x) \end{bmatrix}. \tag{16}$$

### 4. Localization Algorithm with KSPP

#### 4.1. The Connection between Kernel Function and Localization.

The localization algorithm based on kernel method uses a kernel function to map the RSSI vector between nodes into corresponding feature space. In this space, after a linear algorithm is used to calculate the relation between nodes, it is projected into the coordinate space; that is, through the kernel function, the RSSI vector $S (S = [s_1, s_2, \ldots, s_n])$ between $n$ nodes in the monitoring area is related to the distance $D (D = [d_1, d_2, \ldots, d_n])$ between physical spaces, and Figure 3 shows this kind of relation.

In accordance with Figure 5, we can see that when the communication radiuses of two nodes have intersection, then direct communication can be conducted between these two nodes; the weaker the formed RSSI vector is when two nodes have received the RSSI signal of other nodes in the network, the farther the actual distance between them is, which means the less similar they are; the closer the distance between two nodes, the stronger the RSSI vector between them formed by other nodes in the area.

In addition, in the literature [16], Pan proved that when the nodes are deployed in an idyllic environment, the RSSI matrix is positive semidefinite, so we can believe that the RSSI matrix itself is a kernel function matrix. Therefore, we can use the kernel method to measure the similarity between samples, which means the kernel function implicitly embeds the RSSI data into the feature space, and the linear algorithm is used in the feature space to solve the nonlinear relation of RSSI in the Euclidean space.

Assume in the monitoring area, $n$ nodes generate $n$ samples $[x_1, x_2, \ldots, x_n]$, which are mapped into the feature space $H$ through the kernel function $\kappa(\cdot)$; then, $H$ consists of set:

$$\{ \phi(x_1), \phi(x_2), \ldots, \phi(x_n) \}. \tag{17}$$

Furthermore, the closer the nodes in the area are, the more similar the signal strength that they have received from other nodes is; therefore, we can believe the signal strength and feature space $H$ are related, and

$$\phi(\cdot) = [s(x_1, \cdot), s(x_2, \cdot), \ldots, s(x_n, \cdot)]. \tag{18}$$

where $s(x_i, x_j)$ refers to the value of RSSI received by node $x_i$ from node $x_j$, and make $s(x_i, x_i) = 0$; the value of RSSI received by node $x_i$ from other nodes in the area is $s_i = [s(x_1, x_i), s(x_2, x_i), \ldots, s(x_n, x_i)]^T$. Because Gaussian kernel function has the characteristic of maintaining the distance similarity of input space, this paper uses Gaussian kernel function for the calculation of similarity between nodes:

$$\kappa(x_i, x_j) = \exp \left( -\frac{\| \phi(x_i) - \phi(x_j) \|^2}{2\sigma^2} \right)$$

$$= \exp \left( -\frac{\sum_{i=1}^{n} (s_i - s_j)^2}{2\sigma^2} \right). \tag{19}$$

[Figure 5: Correlation between the signal and physical location spaces.]
The localization process of LE-KSPP proposed in this paper includes three steps: first of all, by using the collected RSSI values between nodes, the similarity between nodes is calculated through the Gaussian kernel function; secondly, the adjacency graph is automatically constructed through SP, which transforms the localization problem to dimensionality reduction problem on the graph; finally, the relative coordinate of all nodes in the monitoring area are estimated. The collected RSSI matrix between nodes and the Gaussian kernel function are utilized to construct the kernel function matrix; by solving (13), we can obtain the corresponding feature vector of maximum eigenvalue, and because the experiment was conducted in a two-dimensional space, only the first two biggest eigenvalues are used. Assume \( \lambda_1, \lambda_2 \) are the first two biggest eigenvalues of (13), and \( \lambda_1 \geq \lambda_2 \); their corresponding feature vectors are \( \mathbf{p}_1, \mathbf{p}_2 \). Through \( \mathbf{p}_1, \mathbf{p}_2 \), the base of the relative coordinate space between nodes can be indirectly determined, and make \( \mathbf{c}_i \in \mathbb{R}^2 \) the estimated value of the relative coordinate of node \( \mathbf{x}_i \). \( \mathbf{c}_i \) can be estimated through the following formula:

\[
\mathbf{c}_i = \left[ \frac{\sum_{j=1}^{N} p_{1j} K(x_i, x_j)}{\sum_{j=1}^{N} p_{2j} K(x_i, x_j)} \right],
\]

where

\[
\mathbf{p}_1 = \Delta \mathbf{X}_b = \begin{bmatrix} \Delta x_1, \ldots, \Delta x_{b-1}, \Delta x_{b+1}, \ldots, \Delta x_m \end{bmatrix},
\]

\[
\mathbf{p}_2 = \Delta \mathbf{C}_b = \begin{bmatrix} \Delta c_1, \ldots, \Delta c_{b-1}, \Delta c_{b+1}, \ldots, \Delta c_m \end{bmatrix}.
\]

While \( \Delta x_i = x_i - x_b \), \( \Delta c_i = \mathbf{c}_i - \mathbf{c}_b \). In order to avoid collinear or near collinear relation between beacon nodes, which might cause (23) to be unsolvable, we can use PCA to transform \( \Delta \mathbf{C}_b \) and reduce the dimensionality of data, so that \( \Delta \mathbf{C}_b \Delta \mathbf{C}_b^T \) is not singular any more. Therefore, before obtaining the transformation matrix \( \mathbf{T} \), first of all, conduct PCA transformation to \( \Delta \mathbf{C}_b \); the projection matrix of PCA is recorded as \( \mathbf{P} \); then \( \Delta \mathbf{C}^{PCA} = \mathbf{P}^T \Delta \mathbf{C}_b \), so the expression of the transformation matrix \( \mathbf{T}^{PCA} \) is

\[
\mathbf{T}^{PCA} = \Delta \mathbf{X}_b \mathbf{P} \left( \Delta \mathbf{C}_b^{PCA} \right)^T \left( \Delta \mathbf{C}_b^{PCA} \Delta \mathbf{C}_b^{PCA^T} \right)^{-1}.
\]

At this moment, the absolute coordinate of unknown nodes can be obtained through the following formula:

\[
\tilde{x}_i = \mathbf{T}^{PCA} (\mathbf{c}_i - \mathbf{c}_b) + \mathbf{x}_b, \quad i = m + 1, \ldots, n.
\]

Consider there are \( n \) sensor nodes \( X_1, X_2, \ldots, X_n \) deployed in area \( S \); assume the first \( m \) nodes are beacon nodes \( \{X_1, X_2, \ldots, X_m; m \ll n \} \), the coordinates of beacon nodes are known, which are \( x_i, i = 1, \ldots, m \), the rest \( n - m \) nodes are unknown nodes, and their location information should be determined through certain localization algorithm.

About the LE-KSPP localization algorithm, see Algorithm 1, in which Steps (1) to (3) use the RSSI values between nodes to conduct training and learning; in Step (4), each node uses its adjacency relation with other nodes and estimates the relative coordinate of node through the training and learning model; by referring to the absolute location of beacon node, Step (5) transforms the node location of relative coordinate obtained in the area to absolute coordinate.

5. Simulation and Experiment

Consider a WSN which is comprised of \( n \) nodes \( \{X_1, X_2, \ldots, X_n\} \) deployed in a \( d (d = 2, 3) \)-dimensional monitoring area. The IDs of nodes are \( 1, 2, \ldots, n \), respectively; the actual coordinate of node \( X_i \) is \( \mathbf{x}_i \), and \( \mathbf{X} = [x_1, x_2, \ldots, x_n]^T \) represents the coordinate matrix of node. Without loss of generality, let the first \( m \) \( (m \ll n) \) nodes in the \( n \) nodes be as beacon nodes, and make \( \mathbf{X}_b = [x_1, x_2, \ldots, x_m]^T \) represent the coordinate matrix of beacon nodes. The purpose of localization method is to estimate the values of unknown nodes coordinate \( \bar{x}_i (i = m + 1, m + 2, \ldots, n) \), which makes the estimated coordinate \( \bar{x}_i \) as close to the actual coordinate \( x_i \) of unknown nodes as possible.

Related literature [20] points out that the received signal strength \( s \) between nodes has a certain proportional relation with their distance \( d \). In an ideal environment, node \( i \) is
Input:
- Beacon node coordinate: \( \{x_1, x_2, \ldots, x_m\}, m \geq 3 \)
- RSSI vector between nodes \( \{s_{ij}\}_{i,j=1}^n \)

Output:
- Estimated coordinate of unknown nodes \( \{\hat{x}_{m+1}, \hat{x}_{m+2}, \ldots, \hat{x}_n\} \):

1. By using the collected RSSI vector between nodes \( \{s_{ij}\}_{i,j=1}^n \), the similarity between nodes is calculated through Gaussian kernel function to form the kernel matrix \( K \);

2. By solving the constrained optimization problem (See (6)), we obtain the kernel sparse representation coefficient \( \hat{s}_i (i = 1, 2, \ldots, M) \), and use the \( \hat{s}_i \) combination to obtain the kernel sparse reconstructed adjacent matrix;

3. Solve the optimal projection vector \( p_1, p_2 \), the two maximum eigenvalues \( \lambda_1, \lambda_2 \) are obtained through the generalized characteristic equation

\[
K(S + SS^T - SS^T)Kp = \lambda K^2 p,
\]

and their corresponding eigenvectors \( p_1, p_2 \) are used as the optimal vectors;

4. Through (20), obtain the relative coordinate matrix;

5. Use the beacon nodes in the monitoring area, and transfer the relative coordinate into absolute coordinate; if the beacon nodes have a collinear or near collinear relation, adopt (26); if not, obtain the absolute coordinate \( \{\hat{x}_{m+1}, \hat{x}_{m+2}, \ldots, \hat{x}_n\} \) of unknown node through (22).

Algorithm 1: (LE-KSPP).

within the communication radius of node \( j \), and the signal strength and distance have the following relation:

\[
s(X_i, X_j) = kPd(X_i, X_j)^{-\mu},
\]

where \( P \) is the sending signal voltage, \( k \) is the proportionality coefficient, \( \mu \) is the signal attenuation coefficient, usually \( \mu > 2 \), and \( d(X_i, X_j) \) is the actual distance between node \( i \) and node \( j \). If node \( i \) is not within the communication radius of node \( j \), we denote \( s(X_i, X_j) = 0 \).

This section analyzes and assesses the performance of LE-KSPP localization method through experiment and simulation. In the experiment, the nodes were assigned in a two-dimensional space, and measurement of the distance between nodes adopted the range model-based simulation experiment and actual measurement-based data set, respectively. The parameters involved in the range model are fit with the values collected by Patwari [32], and the actual measurement data set is the RSSI data collected by Patwari experiment team [33] in a 12 m×14 m rectangular area. For the experiment of range model-based, the nodes are randomly or regularly deployed in the monitoring area. In order to investigate whether the algorithm is affected by the obstacle, a blocking experiment was added to the two deployment strategies mentioned above; that is, assume a big blocking object was put in the deployment area to make communication impossible between nodes, and this kind of area has a C sharp. For different network topology structures, by redeploying nodes in the same area for multiple times, all of the reported results are the average over 100 trials. The data set was actually measured through 44 nodes (including 4 beacon nodes) in the monitoring area, the center frequency of node was 2.4 Hz, the broadband direct sequence was used for spread spectrum communication, each RSSI value was measured for 10 times, and each node received and sent for 5 times.

Because it is difficult to evaluate performance by using relative coordinate, we adopted absolute coordinate to express location of nodes in this experiment. We also compare our LE-KSPP algorithm, which is built on one kind of graph embedding localization, with the same kind of algorithms such as MDS-MAP [34], Isomap [35], and LE-KLPP [18]. In addition, the location accuracy of ours and KLPP are
both related to kernel function, which is called Gauss kernel function in our selection. We also found there is a certain relationship between kernel parameter $\sigma$ and the distance between training samples, so in the experiment we assume the average distance between sample nodes of $\sigma$ (4) as 50 times, and the cumulative variance contribution rate of PCA as 90%.

The paper utilizes Average Localization Error (ALE) performance index to evaluate the performance of the algorithm. The formula is shown as follows:

$$\text{ALE} = \frac{\sum_{i=1}^{n} \sqrt{(x_i - \bar{x}_i)^2 + (y_i - \bar{y}_i)^2}}{n \times R},$$

where $n$ is the number of unknown nodes, $(x_i, y_i)$ is the unknown node actual location, $(\bar{x}_i, \bar{y}_i)$ is unknown node evaluated location, and $R$ is the radio range of sensor nodes.

5.1. Localization Results with Range Model-Based. In order to compare the impartiality of the experimental results, when the measurement information is based on signal strength model, in this section, the signal model in literature [32, 33] is used to simulate the signal strength between the nodes:

$$P_{ij} \sim N(\overline{P}_{ij}, \sigma_{\text{dB}}^2),$$

$$\overline{P}_{ij} = P_0 - 10n_p \log \left( \frac{d_{ij}}{d_0} \right),$$

among which $P_{ij}$ represents the transmitted signal power which is received by node $i$ from node $j$, and the unit is dBm; $P_0$ represents the received signal power corresponding to the point of the reference range $d_0$ (dBm); $\sigma_{\text{dB}}^2$ represents the shadow variance. Among these, $n_p$ uses the data in the literature, but $\sigma_{\text{dB}}^2/n_p = 1.7$.

5.1.1. Regular Deployment. In this group of experiments, the nodes were regularly deployed in a 200 m $\times$ 200 m area, in which the grid has a side length of 10 m. Under the unblocked situation, there were 441 nodes in total; under the blocked situation, the number of nodes was changed to 381; 5 to 15 nodes were chosen as the beacon nodes, which were assured that their location information was known.

Before analyzing performance of LE-KSPP algorithm, let us first investigate the two final localization results under different deployment. In Figure 6, the circles denote the unknown node and the squares are the beacon, the line connects the actual coordinate and estimated coordinate of unknown nodes, and the longer the line is, the more the estimated value deviates from the actual location. Figure 6 indicates the localization results for each node under regular deployment with 10 beacons. The LE-KSPP algorithm of ALE for this uniformly deployed network is about 14.9% (Figure 6(a)) and 19.9% (Figure 6(b)).

Figure 7 describes the impact of the beacon nodes number (from 5 to 15) on the ALE of the two localization algorithms of four localization algorithms in a regularly deployed network. They are regularly deployed in two blocked and unblocked environments, respectively. We can note that our algorithm always obtains the best results. Different with the MDS-MAP, Isomap and LE-KLPP, the localization accuracy of our algorithm raises with the increase of beacon nodes. This is because LE-KSPP method can reconstruct the relation between nodes through SP, and opposed to other methods, it
can more self-adaptively choose and decide the node number and weight around its location.

5.1.2. Random Deployment. In this group of experiments, 200 nodes were randomly deployed in a $200 \times 200$ m two-dimensional square area, and 5 to 15 nodes were chosen from the 200 as the beacon nodes. Like the regular deployment, in order to investigate the influence of non-line-of-sight on the localization algorithm, the experiment scenario with obstacle was added to the random deployment experiment, and in addition, the signal strength between nodes was still simulated by using (29).

Similarly, the two definitive localization results were analyzed first. As showed in Figure 8, in these two experiments, the number of beacon nodes is still 10. Under the unblocked situation, LE-KSPP is about 17.5%; the corresponding ALE is 24.6% under the blocked situation.

In Figure 9, for the randomly placed sensor network, we also compare it with the other algorithms (i.e., MDS-MAP, Isomap, and LE-KLPP) under different number of beacons. We can see that our algorithm always achieves the minimal average location error. When the number of beacons is 5 and 7, the ALE of value of MDS-MAP and LE-KLPP is even bigger than 40%, respectively.

5.2. Localization Results with the Actually Measured Data Sets. The literature and experiment of Section 5.1 show that the LE-KLPP has a better performance than the MDS-MAP and
Isomap algorithm; therefore, only LE-KSPP algorithm and LE-KLPP algorithm are compared in this group experiment. The experimental data come from the SPAN lab, and the experimental scenario is as shown in Figure 10.

By using the data set mentioned above, this paper has compared the localization performance of the LE-KLPP and LE-KSPP algorithms, and see Table 1 for the experiment results. In accordance with Table 1, we can see that under separate communication radius (CR), the localization accuracies of LE-KSPP are all higher than the LE-KLPP algorithm, and the ALE has an increase of more than 10%.

Figure 11 shows the localization result under the communication radius of 7.5 m. In the figure, we can see that the closer these two algorithms are to the beacon node point, the smaller the estimated error is, which means, because the beacon node has an accurate location, it is more accurate to use it to determine the location of unknown node; in Figure 11, the LE-KLPP algorithm is used, because it artificially sets parameter $k$ and it causes that an optimal solution (short line) can be obtained in certain area (far from the beacon node), while in some area, it does not apply (long line); in Figure 11, the LE-KSPP algorithm is chosen, because SP is used to self-adaptively obtain the number of adjacent points and the obtained estimation value is relatively stable (the length of line does not have big change).

6. Conclusion

This paper studies the localization problem of sensor network nodes built on signal strength, and the LE-KSPP algorithm is proposed. This LE-KSPP utilizes the kernel method to map the signal strength to a higher-dimensional feature space and conducts SP, and then, the SP coefficient of the obtained signal set is obtained through SP. Because the data in a higher-dimensional feature space have better linear separability, the SP can self-adaptively capture the “local” structure of data, and different sample points are automatically endowed with different “adjacency” number and parameter selection is avoided, which makes the algorithm more applicable to different environments.
The experiments in this paper show that the proposed LE-KSPP algorithm can obtain relatively satisfying localization results on both range model and actually measured data, the impact of the beacon nodes number and blockage on this algorithm is small, and it can self-adapt to various network topology environments, which has a high robustness.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The paper is sponsored by Natural Science Foundation of China (61272379), Prospective and Innovative Project of Jiangsu Province (BY2012201); Provincial University Natural Science Research Foundation of Jiangsu Education Department (12KJD510006, 13KJD520004); Doctoral Scientific Research Startup Foundation of Jinling Institute of Technology (JIT-B-201411).

References

[1] G. Cabri, L. Leonardi, M. Mamei, and F. Zambonelli, “Location-dependent services for mobile users,” IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans., vol. 33, no. 6, pp. 667–681, 2003.

[2] Y. Liu, Z. Yang, X. Wang, and L. Jian, “Location, localization, and localizability,” Journal of Computer Science and Technology, vol. 25, no. 2, pp. 274–297, 2010.

[3] Z. Chen, F. Xia, T. Huang, F. Bu, and H. Wang, “A localization method for the internet of things,” The Journal of Supercomputing, vol. 63, no. 3, pp. 657–674, 2013.

[4] C. Feng, W. S. A. Au, S. Valaee, and Z. Tan, “Received-signal-strength-based indoor positioning using compressive sensing,” IEEE Transactions on Mobile Computing, vol. 11, no. 12, pp. 1983–1993, 2012.

[5] C. Falsi, D. Dardari, L. Mucchi, and M. Z. Win, “Time of arrival estimation for UWB localizers in realistic environments,” Eurasip Journal on Applied Signal Processing, vol. 2006, Article ID 32082, 2006.

[6] L. Yang and K. C. Ho, “An approximately efficient TDOA localization algorithm in closed-form for locating multiple disjoint
sources with erroneous sensor positions,” *IEEE Transactions on Signal Processing*, vol. 57, no. 12, pp. 4598–4615, 2009.

[7] C.-D. Wann and H.-Y. Lin, “Hybrid TOA/AOA estimation error test and non-line of sight identification in wireless location,” *Wireless Communications and Mobile Computing*, vol. 9, no. 6, pp. 859–873, 2009.

[8] G. Mao and B. Fidan, *Localization Algorithms and Strategies for Wireless Sensor Networks: Monitoring and Surveillance Techniques for Target Tracking*, Information Science Reference, New York, NY, USA, 2009.

[9] C. Wang and H. Ma, “Data collection in wireless sensor networks by utilizing multiple mobile nodes,” in 2011 7th International Conference on Mobile Ad-hoc and Sensor Networks, MSN 2011, pp. 83–90, chn, December 2011.

[10] F. Chen, A. P. Chandrakasan, and V. M. Stojanović, “Design and analysis of a hardware-efficient compressed sensing architecture for data compression in wireless sensors,” *IEEE Journal of Solid-State Circuits*, vol. 47, no. 3, pp. 744–756, 2012.

[11] W. Chengqun, *Learning-Based Localization in Wireless Sensor Networks*, Faculty of Information Technology, Zhejiang University, Hangzhou, China, 2009.

[12] P. P. V. Bahl, “RADAR: an in-building RF-based user location and tracking system,” in *Proceedings of the 9th Annual Joint Conference of the IEEE Computer and Communications Societies*, pp. 775–784, 2000.

[13] J. Hightower and G. Borriello, “Location systems for ubiquitous computing,” *Computer*, vol. 34, no. 8, pp. 57–66, 2001.

[14] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, “LANDMARC: indoor location sensing using active RFID,” *Wireless Networks*, vol. 10, no. 6, pp. 701–710, 2004.

[15] M. Essoloh, C. Richard, H. Snoussi, and P. Honeine, “Distributed localization in wireless sensor networks as a pre-image problem in a reproducing kernel Hilbert space,” in *Proceedings of the 16th European Signal Processing Conference (EUSIPCO ’08)*, Lausanne, Switzerland, August 2008.

[16] J. Pan, *Learning-Based Localization in Wireless Sensor Networks*, The Hong Kong University of Science and Technology, Hong Kong, China, 2007.

[17] J. J. Pan, J. T. Kwok, Q. Yang, and Y. Chen, “Multidimensional vector regression for accurate and low-cost location estimation in pervasive computing,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 9, pp. 1181–1193, 2006.

[18] C. Wang, J. Chen, Y. Sun, and X. Shen, “A graph embedding method for wireless sensor networks localization,” in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM ’09)*, Honolulu, Hawaii, USA, December 2009.

[19] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge University Press, 2004.

[20] X. Nguyen, I. M. Jordan, and B. Sinopoli, “A kernel-based learning approach to ad hoc sensor network localization,” *ACM Transactions on Sensor Networks*, vol. 1, no. 1, pp. 134–152, 2005.

[21] I. T. Jolliffe, *Principal Component Analysis*, Springer Series in Statistics, Springer, New York, NY, USA, 2nd edition, 2002.

[22] X. He and P. Niyogi, *Locality Preserving Projections*, MIT Press, Cambridge, Mass, USA, 2003.

[23] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer, New York, NY, USA, 2008.

[24] C. Cortes and M. Mohri, “On transductive regression,” in *Proceedings of the 20th Annual Conference on Neural Information Processing Systems (NIPS ’06)*, December 2006.