Transactions Papers

Localization with Incompletely Paired Data in Complex Wireless Sensor Network

Jingjing Gu, Songcan Chen, and Tingkai Sun

***Abstract*—Localizing sensors based on Received Signal Strength Indicator (RSSI) localization technique in wireless sensor network can be treated as building a mapping between signal and physical spaces, and the mapping is established from a set of given paired signal strengths and physical location data of known sensors. However, in some realistic scenarios, such a set of completely-paired sensor data is not always accessible, which brings a big challenge for localization of sensors. The localization research in such a scenario is currently almost ignored. In this paper, we develop a novel algorithm to tackle this problem in localization with paired as well as many unpaired data by adapting our previously-proposed Locality Correlation Analysis model; the new algorithm is named as Partially Paired Locality Correlation Analysis (PPLCA). Experimental results in both outdoor and indoor environments do show the feasibility and effectiveness of the proposed algorithm.**

***Index Terms*—Wireless sensor network; sensor localization; partially or incompletely paired data.**

I. Introduction

WIRELESS Sensor Network (WSN) has attracted more and more attentions in many fields such as military affairs, aviation and navigation, urban traffic, and medical treatment. The research of sensor localization technique, as one of the fundamental technologies of WSN, is very important to the network activities. We can acquire sensor locations through GPS devices or localization techniques; however, deploying one GPS for each sensor is too expensive and infeasible for a large-scale of wireless network [1]. Instead, an economic and feasible approach is to deploy GPS just for a small number of sensors (also called known sensors or beacons) in the network, thus their locations can be relatively easily gathered while the locations of the remaining sensors are left unknown. Consequently, we have to invent some localization techniques to estimate physical locations of these unknown sensors, for performing some specific tasks through the information gathered from the whole network.

Manuscript received February 22, 2010; revised December 20, 2010 and April 18, 2011; accepted May 16, 2011. The associate editor coordinating the review of this paper and approving it for publication was T. Hou.

J.-J. Gu and S.-C. Chen are with the Department of Computer Science and Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, PR China (e-mail: s.chen@nuaa.edu.cn).

T.-K. Sun is with Nanjing University of Science and Technology.

Digital Object Identifier 10.1109/TWC.2011.070511.100270

Localization techniques can be roughly divided into the range-free and range-based ones in terms of whether the distance information is needed in localization process [2]. Typical examples of range-free techniques include DV-Hop [3], MDS-MAP [4] and Spotlight [5], while typical implementations of range-based techniques include Received Signal Strength Indication (RSSI) [6], Time Difference of Arrival (TDOA) [7], and Angle of Arrival (AOA) [8]. Among them, RSSI attracts much more attentions due to the fact that it does not need any additional actuator infrastructure but just needs existing communication parameters and downloadable wireless maps for the position determination [9]. In this paper we follow this line of research in developing a new localization algorithm.

However, the localization technique just based on RSSI easily produces large predictive bias [10]. To narrow such a deviation, a number of RSSI-based localization algorithms have been invented, where utilizing machine learning technique to localization has attracted more and more attention [11], [12], [13], [14], [15], [16]. One of the main insights of these algorithms is that the relationship between the Received Signal Strengths (RSSs) of sensors and their corresponding physical positions can be learned in advance to construct a mapping between the signal and physical spaces, and this mapping is used as a prediction model to localize the unknown sensors.

In practice, to establish such a model, most of the algorithms mentioned above require the *completely-paired* data in the form of (RSS, Position)s, where RSS denotes a signal strength vector received by the sensor at Position. However such a completely paired data can not always be satisfied in a complex network, which oppositely is easy to incur unpaired data, i.e., signal or position alone provided. At present, there have been very few works [12], [17] published aiming to solve the problem. LeMan in [12] utilizes the Semi-supervised Manifold Regularization learning method [18] to deal with the localization problem with absent physical locations. While [17] adopts mainly the signal attenuation model to impute or infer these missing parts based on MSE model; however, they both need a precise measurement model, more parameters selected and network constraints, which could probably yield a big deviation in location prediction if the measurement model is inappropriately defined.

In this paper, our goal is to solve this problem by developing a novel, more stable, robust and simpler algorithm called Partially Paired Locality Correlation Analysis (PPLCA). It uses a given set of paired and unpaired training data to establish a mapping between the signal and physical spaces. Our idea is inspired by our previous Locality Correlation Analysis (LCA) [19]] and Semi-supervised Laplacian Regularization of Kernel Canonical Correlation Analysis (Semi-LRKCCA) [20]. However, different from LCA, which can be only applied with completely paired data, PPLCA is a powerful extension which is able to cater for the scenario with both completely- and incompletely- paired data. On the other hand, compared with Semi-LRKCCA, which transplants the idea of the Laplacian Regularization (LR) [18] into the Kernel Canonical Correlation Analysis (KCCA) [21] and can deal with such incompletely paired data, PPLCA embodies its differences in the following aspects: (1) In way of utilizing the local structure: almost all localization problems will be formulated by mathematical linguistics finally, and how to reflect the local topology in mappings/models is a key point. Semi-LRKCCA simply embeds the local structures as an additional regularization terms into KCCA for a global mapping, these structures appear just in the constraints but not the main objective problem; by contrast, PPLCA embeds directly the local structures into the CCA [22] and these structures appear simultaneously in both the objective and constraints (refer to Sections III and IV for description in details). (2) In mapping (model) type: WSN environments always present nonlinear and noisy patterns due to path loss, shadowing, etc. Thus, a key point in modeling is how to take advantage of these hidden nonlinearities to improve localization accuracy. Semi-LRKCCA builds a global nonlinear mapping by borrowing the kernel method [23]; while PPLCA builds a linear mapping, and at the same time can implicitly reflect the nonlinear correlations between the spaces by aggregating all the local correlations from individual data points. (3) In the number of parameters involved: Semi-LRKCCA involves about 13 parameters to be tuned totally during establishing the mapping which costs much time and energy resources; while PPLCA just involves the number of neighbors and thus greatly reduces the time of building the mapping. Consequently, these differences lead to the differences in performances between Semi-LRKCCA and PPLCA. Furthermore, as far as the effectiveness in utilizing both local structure and computing resources is considered, Semi-LRKCCA in WSN does not seem so applicable, however, our PPLCA does not have such a trouble and is more stable, more accurate and more practical.

The rest of this paper is organized as follows. Section II describes application environment and localization problem. In Section III, we give a brief review of related works. In Section IV, we formulate our PPLCA and the localization algorithm. Experimental results are shown in Section V. In Section VI, we conclude the paper with a summary and an outlook to future work.

II. Application Environment and Localization Problem

Generally, each pair data (RSS_i, $Position_i$) comes from the ith known sensor and we can utilize the paired data collected from all known sensors to build a localization model, if one of RSS_i and $Position_i$ is unavailable or missing, this set of data is called unpaired data. Such unpaired data possibly occurs in some cases such as (1) Many known sensors fail to or uneasily collect their position information, e.g. 1) a small number of GPSs; 2) some environments in which GPSs cannot receive signal; 3) difficulties in collecting some location data by equipments or manual measurements. And in this scenario, these known sensors are hardly identified from unknown sensors, and as a result, their available data are only signal data. (2) In a bad weather or harsh network, for example, in a raining day, the signal can only be transmitted in a short distance (usually less than 1m according to our measurement) and will become very weak beyond this range; another example is in a harsh network of battlefield environment, some special distortion/isolation tricks or equipments set by enemies, e.g. absorbing barrier and signal-shielding device between the transmitter and the target, can isolate or change the signal propagation and may weaken the signal strengths so as to be unavailable, even to almost zero. In such a system, if sensors are designed to send packets in a time-specific period, the packets inevitably contain incomplete information as well.

Obviously, in these scenarios, *just* using those less paired data is difficult to build an accurate localization model. Though there are some other regular methods to solve these problems respectively, such as checking up and redeploying or replacing the sensors, our goal is just using a learning algorithm to work out these problems and localization simultaneously irrespective of those extra manual or electromechanical operations. Therefore, in order to establish an effective model, we jointly utilize paired and those additionally unpaired data to achieve this goal, our PPLCA is specially proposed, and it sufficiently uses inherent information contained in paired and unpaired data, such as local network structure and neighboring relationship.

The core procedure of our PPLCA employs the concept of machine learning. The main insight of machine learning-based methods is to take advantage of the inherent information from data (e.g. implicit local network structure, neighboring relationship, nonlinear and noisy pattern, etc.), which can (1) improve the localization performance and stability; (2) make up the impact of environment conditions (temperature, humidity, etc.) because the mapping can dig up and reflect both topology structures and characteristics of WSN; (3) avoid updating the mapping based on our method when the parameters and other settings of hardware devices are unchanged in the same environment, even if some sensors' positions vary, because the mapping reflects the characteristics of the whole network region. As long as sensors are deployed in this region, where the mapping is applicable, the collected data will imply these network characteristics.

In this paper, our PPLCA is inspired by our previous LCA, which based on Canonical Correlation Analysis (CCA) [22]. In fact, given two sets of paired data, CCA is a classical algorithm to reveal the hidden correlations between the two datasets. This correlation between signal and physical spaces in WSN is that: if two sensors are neighbors in physical space, their corresponding RSSs are similar. And thus CCA can be used to reveal this hidden correlation, which may be contaminated by nonlinear and noisy patterns due to some environments, equipments, or/and human factors. However, CCA only find the linear correlation, which makes it difficult to satisfy the need of real environment. Thus, some researchers resort to the kernel trick [23] in CCA (Kernel CCA/KCCA) [21] to solve those nonlinear correlation problems in the application to WSN localization [24]. But KCCA hardly considers local structures of WSN, which is important for accurate localization. In order to utilize the local topologies, we have proposed LCA which is also based on CCA but can characterize the nonlinear correlation between both spaces by aggregating all the local information of the data. However, CCA, LCA and KCCA can only handle the paired data and thus cannot deal with the incompletely-paired data. To solve this problem, Semi-LRKCCA in [20] was proposed and applied it to image/text processing but was not quite suitable to WSN, thus developing a novel localization algorithm for the above-mentioned scenario is our desired goal in this paper.

In the system which we mentioned in our paper, it consists of a base station and a set of sensors. The set includes Access Points (APs) which are able to send signals and the non-APs receiving signals. Generally, the propagation-based localization algorithms need to know all APs' locations for calculating the physical distances from the APs to any other sensors by RSS measurement model. In the contrast, learning-based algorithms (including ours) need neither the locations nor calculation of these distances, which could avoid more errors incurred by imprecise signal models and inappropriate selection of model parameters. However, due to the characteristic of learning-based algorithms themselves, we need to have a set of training data to build a predictive model. Therefore, the non-APs in this system can be both (a few) sensors with known positions and (many) position-unknown sensors. Here, the number of APs, i.e. p (normally 5-8 is enough), could be much less than the number of known sensor. Thus, n known sensors send n paired packets totally to the base station, and each one includes signal strength from p APs (forming a p-dimension signal vector) and physical position of the corresponding senor. No communication among the sensors occurs. It is worth pointing out that, our PPLCA can be applied not only in the above system, but also in other cases, e.g., the system consisting of APs and unknown sensors, in which APs are treated as known sensors, and the paired data is formed as (RSS, Position)s, where RSS denotes signal strengths from the AP at Position.

III. Brief Reviews of CCA, LCA, KCCA and Semi-LRKCCA

Before the introduction of our algorithm, for completeness, we will briefly review CCA, LCA and Semi-LRKCCA in the next sub-sections.

A. CCA

Given n pairs of data, $X = [x_1, \cdots, x_n] \in R^{p\times n}$ and $Y = [y_1, \cdots, y_n] \in R^{q\times n}$. For simplification, we usually suppose that X and Y have been centralized, which means $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} y_i = 0$ and $\overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i = 0$. CCA aims to find two basis or projection vectors, $w_x \in R^p$ and $w_y \in R^q$,respectively for two data sets such that their correlation between $w_x^T x$ and $w_y^T y$ is maximized, where x and y is any pair from given data set $\{x_i, y_i\}_{i=1}^{n}$. Thus we can write the correlation expression [21], [22] as follows:

$$\begin{aligned}\rho &= \frac{E\left[\left(w_x^T x\right)\left(w_y^T y\right)\right]}{E\left[\left(w_x^T x\right)^2\right]E\left[\left(w_y^T y\right)^2\right]}\\ &= \frac{E\left[w_x^T x y^T w_y\right]}{E\left[w_x^T x x^T w_x\right]E\left[w_y^T y y^T w_y\right]}\\ &= \frac{w_x^T C_{xy} w_y}{\sqrt{\left(w_x^T C_{xx} w_x\right)}\sqrt{\left(w_y^T C_{yy} w_y\right)}}\end{aligned} \quad (1)$$

Equivalently, (w_x,w_y)can also be obtained through solving the following optimization problem:

$$\begin{aligned}&\max_{w_x, w_y} w_x^T C_{xy} w_y\\ &st. w_x^T C_{xx} w_x = 1\\ &\quad w_y^T C_{yy} w_y = 1\end{aligned} \quad (2)$$

Where $C_{xx} = XX^T \in R^{p\times p}$ and $C_{yy} = YY^T \in R^{q\times q}$ denote the within-set covariance matrices of X and Y respectively, and $C_{xy} = XY^T = C_{yx}^T \in R^{p\times q}$ denotes the between-set covariance matrices. w_x and w_y are called the canonical vectors and $\lambda\,(\geq 0)$ is exactly the correlation between $w_x^T x$ and $w_y^T y$. It can be proved $w_y = \frac{1}{\lambda} C_{yy}^{-1} C_{yx} w_x$. In general, we can obtain a set of first d smallest (nonzero) correlation vectors $(w_{x_i}, w_{y_i})\,(i = 1, \cdots, d)$ from (2), thus we can have two projection matrices $W_x = [w_{x1}, \cdots, w_{xd}]$ and $W_y = [w_{y1}, \cdots, w_{yd}]$. For stability of the operating, generally we use Singular Value Decomposition (SVD) [21] to solve the above procedure.

B. LCA

First we will introduce definition of similarity: local (geometrical) structure can be characterized by Laplacian graph [18], [25] whose nodes and edges respectively correspond to the data points and the similarities between these points, we defined the similarity matrices $S_X = \left\{S_{ij}^X\right\}_{i,j=1}^{n}$ and $S_Y = \left\{S_{ij}^Y\right\}_{i,j=1}^{n}$ respectively for the two datasets to characterize their local structures. The similarity S_{ij}^X and S_{ij}^Y are defined as:

$$\begin{aligned}S_{ij}^X &= \begin{cases}\exp\frac{-\|x_i - x_j\|^2}{t_x} & \text{if i and j are neighbors in X}\\ 0 & \text{otherwise}\end{cases}\\ S_{ij}^Y &= \begin{cases}\exp\frac{-\|y_i - y_j\|^2}{t_y} & \text{if i and j are neighbors in Y}\\ 0 & \text{otherwise}\end{cases}\end{aligned} \quad (3)$$

Where $t_x = \sum_{i=1}^{n}\sum_{j=1}^{n}\|x_i - x_j\|^2 / n(n-1)$.We can see if the closer between x_i and x_j, the larger S_{ij}^X is, which makes locality of data be preserved better. The definition of t_y can be obtained as the same procedure. Now we review the definition of CCA (2), the correlation expression can be recast without data-centralization [21], [22] :

$$\max_{w_x,w_y} \sum_{i=1}^{n} w_x^T(x_i-\overline{x})(y_i-\overline{y})^T w_y$$
$$s.t. \sum_{i=1}^{n} w_x^T(x_i-\overline{x})(x_i-\overline{x})^T w_x = 1$$
$$\sum_{i=1}^{n} w_y^T(y_i-\overline{y})(y_i-\overline{y})^T w_y = 1 \quad (4)$$

Take a close look at the objective (4), for any given sample pair $(x_i,\ y_i)$, their partial canonical correlation is $w_x^T(x_i-\overline{x})(y_i-\overline{y})^T w_y$. And for the neighborhood of $(x_i,\ y_i)$, we can replace the sample means $(\overline{x},\ \overline{y})$ respectively with the local means $\Big(\sum_{j=1}^n(S_{ij}^X/\sum_{j=1}^n S_{ij}^X)x_j,$ $\sum_{j=1}^n(S_{ij}^Y/\sum_{j=1}^n S_{ij}^Y)y_j\Big)$. In fact, $S_{ij}^X/(\sum_{j=1}^n S_{ij}^X)$ is just the normalized term from S_{ij}^X, thus for simplification, we suppose that S_X and S_Y have been normalized, i.e. $S_{ij}^X = S_{ij}^X/(\sum_{j=1}^n S_{ij}^X)$, $S_{ij}^Y = S_{ij}^Y/(\sum_{j=1}^n S_{ij}^Y)$. Thus we can construct so-called local canonical correlation: $w_x^T(x_i - \sum_{j=1}^n S_{ij}^X x_j)(y_i-\sum_{j=1}^n S_{ij}^Y y_j)^T w_y$. Consequently, we form the following optimization problem with corresponding constraints:

$$\max_{w_x,w_y} w_x^T \sum_{i=1}^{n}(x_i-\sum_{j=1}^{n} S_{ij}^X x_j)(y_i-\sum_{j=1}^{n} S_{ij}^Y y_j)^T w_y$$
$$s.t. w_x^T \sum_{i=1}^{n}(x_i-\sum_{j=1}^{n} S_{ij}^X x_j)(x_i-\sum_{j=1}^{n} S_{ij}^X x_j)^T w_x = 1$$
$$w_y^T \sum_{i=1}^{n}(y_i-\sum_{j=1}^{n} S_{ij}^Y y_j)(y_i-\sum_{j=1}^{n} S_{ij}^Y y_j)^T w_y = 1 \quad (5)$$

The objective (5) is just the formulation of LCA [19]. And with matrix notation, (5) can be rewritten as:

$$\max_{w_x,w_y} w_x^T X M_{XY} Y^T w_y$$
$$s.t. w_x^T X M_{XX} X^T w_x = 1$$
$$w_y^T Y M_{YY} Y^T w_y = 1 \quad (6)$$

Where $M_{XY} = I - S_X - S_Y + S_X \circ S_Y$, $M_{XX} = I - 2S_X + S_X \circ S_X$, $M_{YY} = I - 2S_Y + S_Y \circ S_Y$, The operator "'$\circ$" is element-by-element product in two matrices, i.e., $(A \circ B)_{ij} = A_{ij}B_{ij}$. Obviously, LCA aims to describe the local structure of the two datasets. Using Lagrange multiplier method to solve this optimization problem (6), we can get canonical correlation vector pair $(w_x,\ w_y)$.

C. Semi-LRKCCA

The kernel trick solves a nonlinear problem by transferring the original nonlinear data into a higher (even infinite) dimensional feature space, where the linear algorithm can be subsequently used [21]. In order to kernelize CCA, we fist utilize (implicit) mappings $x \mapsto \phi(x)$ and $y \mapsto \varphi(y)$, and then perform traditional CCA in the two high-dimensional feature spaces. Let $\phi(X) = [\phi(x_1), \cdots, \phi(x_n)]$ and $\varphi(Y) = [\varphi(y_1), \cdots, \varphi(y_n)]$, $w_\phi = \phi(X)\alpha$ and $w_\varphi = \varphi(Y)\beta$ be two canonical vectors derived from the new mapped datasets [21], [23], where α and β are n-dimensional vectors. Accordingly, from (2), optimization problem of KCCA can be formulated as follows:

$$\max_{\alpha,\beta} \alpha^T K_X K_Y \beta$$
$$s.t. \alpha^T K_X^2 \alpha = 1$$
$$\beta^T K_Y^2 \beta = 1 \quad (7)$$

Here, kernel functions $K_X = \phi(X)^T\phi(X)$ and $K_Y = \varphi(Y)^T\varphi(Y)$.

With the incompletely paired data, authors of [20] provided the following implementation: given two sets of data $\widehat{X} = [x_1, \cdots, x_n, x_{n+1}, \cdots, x_{n+ps}] \in R^{p\times(n+ps)}$ and $\widehat{Y} = [y_1, \cdots, y_n, y_{n+1}, \cdots, y_{n+pl}] \in R^{q\times(n+pl)}$, in which $X = [x_1, \cdots, x_n] \in R^{p\times n}$ and $Y = [y_1, \cdots, y_n] \in R^{q\times n}$ $(n \le \min\{(n+ps), (n+pl)\})$ are n pairs data. Defining $K_{XX} = \phi(X)^T\phi(X)$, $K_{\widehat{X}X} = \phi(\widehat{X})^T\phi(X)$, and $K_{\widehat{X}\widehat{X}} = \phi(\widehat{X})^T\phi(\widehat{X})$ as kernel matrices between the various sets of the X data and analogously defining kernel matrices for Y data, thus Semi-LRKCCA can be formulated as the following optimization problem:

$$\max_{\alpha,\beta} \quad \alpha^T K_{\widehat{X}X} K_{Y\widehat{Y}} \beta$$
$$s.t. \quad \alpha^T (K_{\widehat{X}X} K_{X\widehat{X}} + R_{\widehat{X}})\alpha = 1$$
$$\beta^T (K_{\widehat{Y}Y} K_{Y\widehat{Y}} + R_{\widehat{Y}})\beta = 1 \quad (8)$$

Here $R_{\widehat{X}} = \epsilon_X K_{\widehat{X}\widehat{X}} + \frac{\gamma_X}{ps^2} K_{\widehat{X}\widehat{X}} L_{\widehat{X}} K_{\widehat{X}\widehat{X}}$ and $L_{\widehat{X}} = D_{\widehat{X}} - W_{\widehat{X}}$ [18] in $\widehat{X}$ data, where $W_{\widehat{X}_{ij}}$ is the edge weight between x_i and x_j, and diagonal matrix $D_{\widehat{X}}$ is given by $D_{\widehat{X}_{ii}} = \sum_{i=1}^{ps} W_{\widehat{X}_{ij}}$. $R_{\widehat{Y}}$ can be formulated analogously.

It is worth noting that in Semi-LRKCCA, 1) the local structure is just embedded in $R_{\widehat{X}}(R_{\widehat{Y}})$ in the constraints, but not in its objective function; 2) there are more than ten parameters to be tuned in learning (produced in $K_{\widehat{X}X}$, $K_{X\widehat{X}}$, $K_{\widehat{X}\widehat{X}}$, $R_{\widehat{X}}$, $K_{\widehat{Y}Y}$, $K_{Y\widehat{Y}}$, $K_{\widehat{Y}\widehat{Y}}$, $R_{\widehat{Y}}$), which is too complicated for real WSN application.

IV. Localization with Incompletely Paired Data

A. PPLCA

Given two sets of incompletely paired un-centralized data $\widehat{X} = [x_1, \cdots, x_n, x_{n+1}, \cdots, x_{n+ps}] \in R^{p\times(n+ps)}$ and $\widehat{Y} = [y_1, \cdots, y_n, y_{n+1}, \cdots, y_{n+pl}] \in R^{q\times(n+pl)}$, where n pairs of data $X = [x_1, \cdots, x_n] \in R^{p\times n}$ and $Y = [y_1, \cdots, y_n] \in R^{q\times n}$ $(n \le \min\{(n+ps), (n+pl)\})$ are one-to-one corresponded or paired data and the remaining data $[x_{n+1}, \cdots, x_{n+ps}] \in R^{p\times ps}$ and $[y_{n+1}, \cdots, y_{n+pl}] \in R^{q\times pl}$ are unpaired. Furthermore, we define the similarity matrices $\widehat{S}_X = \left\{\widehat{S}_{ij}^X\right\}_{i,j=1}^{n,n+ps}$ and $\widehat{S}_Y = \left\{\widehat{S}_{ij}^Y\right\}_{i,j=1}^{n,n+pl}$ to characterize the local structures in $\widehat{X}$ and $\widehat{Y}$ data sets according to Eq.(3), but $\widehat{S}_X(\widehat{S}_Y)$ reflects similarity between the whole data $\widehat{X}$ and the paired data X ($\widehat{Y}$ and Y). Thus the sizes of $\widehat{S}_X$ and $\widehat{S}_Y$ are respectively $n \times (n+ps)$ and $n \times (n+pl)$. Now let us take a further look at the objective of LCA (5) again, for any given one sample pair (x_i, y_i), their (local) canonical correlation embedded by its local structure is defined as $w_x^T(x_i - \sum_{j=1}^n S_{ij}^X x_j)(y_i - \sum_{j=1}^n S_{ij}^Y y_j)^T w_y$, where S_{ij}^X and S_{ij}^Y are the similarity only between paired data X and Y, and thus the whole correlation of all data pairs is aggregated to form as $w_x^T \sum_{i=1}^n (x_i - \sum_{j=1}^n S_{ij}^X x_j)(y_i - \sum_{j=1}^n S_{ij}^Y y_j)^T w_y$. Next, for the incompletely paired data, we synthesize the (whole) canonical correlation by aggregating all local correlations into a whole one as in LCA, to characterize the incompletely paired scenario in terms of $\widehat{w}_x^T \sum_{i=1}^n (x_i - \sum_{j=1}^{n+ps} \widehat{S}_{ij}^X x_j)(y_i - \sum_{j=1}^{n+pl} \widehat{S}_{ij}^Y y_j)^T \widehat{w}_y$.

Consequently, we form the following optimization problem with the corresponding constraints:

$$\max_{\widehat{w}_x, \widehat{w}_y} \widehat{w}_x^T \sum_{i=1}^{n} (x_i - \sum_{j=1}^{n+ps} \widehat{S}_{ij}^X x_j)(y_i - \sum_{j=1}^{n+pl} \widehat{S}_{ij}^Y y_j)^T \widehat{w}_y$$
$$s.t. \widehat{w}_x^T \sum_{i=1}^{n} (x_i - \sum_{j=1}^{n+ps} \widehat{S}_{ij}^X x_j)(x_i - \sum_{j=1}^{n+ps} \widehat{S}_{ij}^X x_j)^T \widehat{w}_x = 1$$
$$\widehat{w}_y^T \sum_{i=1}^{n} (y_i - \sum_{j=1}^{n+pl} \widehat{S}_{ij}^Y y_j)(y_i - \sum_{j=1}^{n+pl} \widehat{S}_{ij}^Y y_j)^T \widehat{w}_y = 1 \quad (9)$$

Optimizing the objective (9) results in PPLCA, with notation of matrix and from some algebraic manipulations, (9) can be recast as:

$$\max_{\widehat{w}_x, \widehat{w}_y} \widehat{w}_x^T (XY - X\widehat{S}_Y^T \widehat{Y}^T - \widehat{X}\widehat{S}_X Y^T + \widehat{S}_X \widehat{S}_Y^T \widehat{Y}^T) \widehat{w}_y$$
$$s.t. \widehat{w}_x^T (XX^T - X\widehat{S}_X^T \widehat{X}^T - \widehat{X}\widehat{S}_X X^T + \widehat{S}_X \widehat{S}_X^T \widehat{X}^T) \widehat{w}_x = 1$$
$$\widehat{w}_y^T (YY^T - Y\widehat{S}_Y^T \widehat{Y}^T - \widehat{Y}\widehat{S}_Y Y^T + \widehat{S}_Y \widehat{S}_Y^T \widehat{Y}^T) \widehat{w}_y = 1 \quad (10)$$

Denote $A = [I_n, 0]_{(n+ps)\times n}^T$ and $B = [I_n, 0]_{(n+pl)\times n}^T$, thus $X = \widehat{X}A$ and $Y = \widehat{Y}B$. Then we can rewrite (10) as:

$$\max_{\widehat{w}_x, \widehat{w}_y} \widehat{w}_x^T \widehat{X} \widehat{G}_{XY} \widehat{Y}^T \widehat{w}_y$$
$$s.t. \widehat{w}_x^T \widehat{X} \widehat{G}_{XX} \widehat{X}^T \widehat{w}_x = 1$$
$$\widehat{w}_y^T \widehat{Y} \widehat{G}_{YY} \widehat{Y}^T \widehat{w}_y = 1 \quad (11)$$

Where $\widehat{G}_{XY} = AB^T - A\widehat{S}_Y^T - \widehat{S}_X^T B^T + \widehat{S}_X \widehat{S}_Y^T$, $\widehat{G}_{XY} = AA^T - A\widehat{S}_X^T - \widehat{S}_X^T A^T + \widehat{S}_X \widehat{S}_X^T$, $\widehat{G}_{XY} = BB^T - B\widehat{S}_Y^T - \widehat{S}_Y^T B^T + \widehat{S}_Y \widehat{S}_Y^T$. Finally, we can get the canonical correlation vector pair ($\widehat{w}_x^T$,$\widehat{w}_y^T$) by solving the following eigen-equation (12):

$$\begin{bmatrix} & \widehat{X}\widehat{G}_{XY}\widehat{Y}^T \\ \widehat{Y}\widehat{G}_{YX}\widehat{X}^T & \end{bmatrix} \begin{bmatrix} \widehat{w}_x \\ \widehat{w}_y \end{bmatrix} = \lambda \begin{bmatrix} \widehat{X}\widehat{G}_{XX}\widehat{X}^T & \\ & \widehat{Y}\widehat{G}_{YY}\widehat{Y}^T \end{bmatrix} \begin{bmatrix} \widehat{w}_x \\ \widehat{w}_y \end{bmatrix} \quad (12)$$

Like the solution of CCA, the canonical vectors ($\widehat{w}_x$ and $\widehat{w}_y$) of (12) can be obtained by the SVD [21]. As a result, we have the following solutions: $\widehat{w}_{xi} = (\widehat{X}\widehat{G}_{XX}\widehat{X}^T)^{-\frac{1}{2}} \widehat{u}_i$ and $\widehat{w}_{yi} = (\widehat{Y}\widehat{G}_{YY}\widehat{Y}^T)^{-\frac{1}{2}} \widehat{v}_i$, where $\widehat{u}_i$ and $\widehat{v}_i$ are respectively left singular and right singular eign-vectors of the $\widehat{H} = (\widehat{X}\widehat{G}_{XX}\widehat{X}^T)^{-\frac{1}{2}} \widehat{X}\widehat{G}_{XY}\widehat{Y}^T (\widehat{Y}\widehat{G}_{YY}\widehat{Y}^T)^{-\frac{1}{2}}$. When $\widehat{G}_{XX}$ is singular, we generally replace $\widehat{G}_{XX}$ with $\widehat{G}_{XX} + \epsilon I$ (with ϵ being small positive number) to overcome its singularity as well as ensure the computational stability, $\widehat{G}_{XY}$ and $\widehat{G}_{YY}$ are done in the similar way. Actually, LCA is just a special case of PPLCA when the two datasets are paired. Compared with Eq.(5) and Eq.(9), it is easy to prove when $ps = pl = 0$, the expressions of LCA and PPLCA are same.

From the objective of PPLCA (9), it is worth noting that, 1) local structures are embedded directly into the LCA and appear simultaneously in both the objective and constraints; 2) a global mapping with implicit nonlinear correlation can be sought by aggregating all the local linear correlations as done in LCA; 3) only the number of neighbors needs to be tuned during establishing the model (in $\widehat{S}_X$ and $\widehat{S}_Y$), which greatly reduces time and energy resources for real WSN application.

B. Location estimation in WSN

Suppose that a set of sensors are deployed in some geographical area, including p APs which send signals to non-APs. The input to our algorithm only needs two sets of data from the non-AP sensors: a signal strength matrix from the p APs $\widehat{X} \in R^{p\times(n+ps)}$ and a physical coordinate matrix of sensors $\widehat{Y} \in R^{q\times(n+pl)}$. In $\widehat{X}$, x_{ij} denotes the RSS of sensor $j(j = 1, \cdots, n)$ receiving from AP $i(i = 1, \cdots, p)$ and x_{ij} takes zero value if sensor j lies beyond the transmission range of AP i. Our goal is to build a mapping between signal space and physical space from such a set ($\widehat{X}$,$\widehat{Y}$). Upon learning the mapping, we can estimate the physical location or coordinate y_g for an unknown sensor g from its signal vector $x_g = (x_{g1}, x_{g2}, \cdots, x_{gp})^T$.

Generally, RSSI-localization includes two phases: 1) model-building phase in which the mapping between signal and physical spaces is established; 2) an online localization phase in which the locations of unknown sensors are estimated using the learnt mapping. A summary of our algorithm is provided as follows, where Steps 1 & 2 are finished in a model-building phase while Steps 3 & 4 are done in an online localization phase:

Input: ($\widehat{X}$, $\widehat{Y}$).

Output: Estimated location coordinate y_g of the unknown sensor g.

Step 1: Compute the similarity matrices $\widehat{S}_X = \left\{\widehat{S}_{ij}^X\right\}_{i,j=1}^{n,n+ps}$ and $\widehat{S}_Y = \left\{\widehat{S}_{ij}^Y\right\}_{i,j=1}^{n,n+pl}$ between the whole data and the paired data according to (3).

Step 2: Solve (12) to obtain the projection matrices $\widehat{W}_X = [\widehat{w}_{x1}, \cdots, \widehat{w}_{xd}]$ and $\widehat{W}_Y = [\widehat{w}_{y1}, \cdots, \widehat{w}_{yd}]$ which maximize the correlation between $\widehat{W}_X^T x_i$ and $\widehat{W}_Y^T y_i$ $(i = 1, \cdots, n)$. For the whole data, their projection are $\widehat{P}_X = \widehat{W}_X^T X$ and $\widehat{P}_Y = \widehat{W}_Y^T Y$. Thus we can estimate the location of an unknown sensor g through its neighbors in the corresponding signal space.

Step 3: Project $x_g = (x_{g1}, x_{g2}, \cdots, x_{gp})^T$ to the new space using $\widehat{W}_x$: $\widehat{P}_{Xg} = \widehat{W}_X^T x_g$.

Step 4: Find K neighbors closest to $\widehat{P}_{Xg}$ in $\widehat{P}_X$ and then use the centroid method to estimate the physical location of the g from these neighbors' physical coordinates : $\{y_1, y_2, \cdots, y_K\} \in R^{q\times n}$: $y_g = \frac{y_1+y_2+\cdots+y_K}{K}$.

C. Time Complexity of Localization Algorithms

As described in Introduction, rare localization research focuses on such scenario of incompletely paired data, and we only find Semi-LRKCCA is one closest in methodology to our work so far, but it is just applied in image/character processing.

In other words, applying it to the localization in WSN is our try and mainly for comparison with our PPLCA as well.

Now let us carry out a quick review of sizes of given datasets or (data) matrices: $\widehat{X} - p \times ps$; $\widehat{Y} - q \times pl$; $X - p \times n$; $Y - q \times n$, where $n \leq \min\{ps, pl\}$ and $\{p, q = 2 \text{ or } 3 \text{ in real world usually}\} \ll \min\{ps, pl\}$. Thus the main computing terms and their corresponding time complexities of Semi-LRKCCA are: $K_{\widehat{X}\widehat{X}} - O(p \times ps^2)$; $K_{X\widehat{X}} - O(p \times ps \times n)$; $K_{\widehat{X}X} - O(p \times ps \times n)$; $L_{\widehat{X}} - O(p \times ps^2)$; $K_{\widehat{Y}\widehat{Y}} - O(q \times pl^2)$; $K_{\widehat{Y}Y} - O(q \times pl \times n)$; $K_{Y\widehat{Y}} - O(q \times pl \times n)$; $L_{\widehat{Y}} - O(q \times pl^2)$; solving (8) by recasting a eigen-equation costs $O((ps + pl)^3)$. As a result, the total complexity in the model-building phase of Semi-LRKCCA is $O(p \times ps \times (ps + 2n + ps) + q \times pl \times (pl + 2n + pl) + (ps + pl)^3)$. Moreover, due to $n \leq \min\{ps, pl\}$, this complexity can be reduced to $O(p \times ps^2 + q \times pl^2 + (ps + pl)^3)$. All of the above mentioned kernel functions are chosen as RBF kernel [23].

Next, we turn to the main computing terms and their corresponding time complexities of our PPLCA: $\widehat{S}_X - O(p \times ps \times n)$; $\widehat{S}_Y - O(q \times pl \times n)$; solving eigen-equation (12) costs $O((p + q)^3)$. As a result, the total complexity in the model-building phase is $O(n \times (p \times ps + q \times pl) + (p + q)^3)$.

From the above analysis and in comparison with the cubic terms of Semi-LRKCCA and PPLCA, obviously, $(ps + pl)^3$ is much larger than $(p + q)^3$. And for their remaining terms, Semi-LRKCCA has the quadratic terms while PPLCA only has linear terms. In addition, Semi-LRKCCA has about 13 parameters needing to be tuned, thus clearly, our PPLCA has greater time-saving.

Besides, in the online localization phase, due to no need of the kernel computation, the time of PPLCA costs just $1/ps$ times of the Semi-LRKCCA's, which the former is $O(q \times ps)$ and the latter is $O(q \times ps^2)$.

V. Experimental Results

This section presents a series of comparative studies of localization in the outdoor and indoor environment, respectively. The first experimental data are collected from the fire alarming project of ancient city wall, whose partial network environment is showed in Fig. 1(sub-section C), and the second one from Department of Computer Science of the Hong Kong University of Science and Technology (HKUST). Besides PPLCA, we also carry out Semi-LRKCCA, LCA which lacks a mechanism of using the unpaired data, and LeMan which only uses in such a case that the number of RSSs is greater than or equal to the number of physical positions.

A. Experimental Setting

As mentioned in section IV with the description of PPLCA, the mathematical expression of unpaired data here is the unmatch between the numbers of the RSSs and physical positions. For simulation, we split randomly the whole collected (paired) data set into two halves: the one is used in training for building model and the other is for test. For the (paired) training dataset, we artificially produce these cases: randomly miss some parts from the RSSs (here from 0% to 50%) and from physical positions (0%, 10%, 30%, 50%) respectively to yield incompletely paired data. Here we do not consider the case of more than 50% unpaired data because this occurs rarely in real world (if it happened, we would think the network should be re-deployed).

In this paper, all the WSN localization comparative experiments are run in Matlab 7.0 on a Pentium IV processor of 1.60GHz with 1G RAM. Each signal value in db from an AP is negative and then is converted to corresponding positive one by adding the minimal signal value. To reduce the statistical variability, the reported results here are averaged over 10 repetitions.

B. Parameter Selection

In our experiments, we select all the parameters by Cross-validation (CV) [26]: the training data is generally split into training and validation subsets, and then through an exhaustive search in the given ranges of the parameters to be tuned, the values of the parameters corresponding to the best validation result would be selected. Here, to determine the parameters in our PPLCA and other comparative algorithms, we use 70% of given training data for building the model and the rest for validating, and restrict the given ranges of all the real-valued parameters to $[10^{-4}, 10^4]$ with different multiplying increments and those of the integer-valued parameters k and K to $[1,\ 0.10 \times pl]$ with the additive increment 1. Finally, the averaged results are reported after replicating the CV 10 times using different training data. The best values (search increments) of all algorithms in the first network are:

PPLCA: $k - -3(1), K - -4(1)$;
LCA: $k - -3(1), K - -4(1)$;
Semi-LRKCCA: $k - -3(1), K - -21(1), KW - -0.3162(10^{0.5}); \epsilon_X - -0.001(10^1), \gamma_X - -10(10^1), \epsilon_Y - -0.01(10^1), \gamma_Y - -10(10^1)$;
LeMan: $k - -3(1), K - -4(1), KW - -0.1(10^0.5), \gamma_A - -0.0032(10^{0.5}), \gamma_I - -0.001(10^{0.5})$.

The best values (search increments) of all algorithms in the second network are:

PPLCA: $k - -15(1), K - -10(1)$;
LCA: $k - -15(1), K - -10(1)$;
Semi-LRKCCA: $k - -11(1), K - -5(1), KW - -0.01(10^{0.5}); \epsilon_X - -0.001(10^1), \gamma_X - -0.001(10^1), \epsilon_Y - -0.001(10^1), \gamma_Y - -100(10^1)$;
LeMan: $k - -14(1), K - -12(1), KW - -0.01(10^{0.5}), \gamma_A - -0.01(10^0.5), \gamma_I - -0.3162(10^0.5)$.

Here, kernel width (KW) means specify width parameter for a kernel function. For simpleness and workability of algorithms, we suppose all the values of KW are same in different kernel functions in the one algorithm and one network environment. Regularizes γ_A and γ_I in LeMan respectively control the complexity of the functions [18] and are carefully tuned.

It is worth pointing out again that the number of the parameters in PPLCA is much smaller than those in Semi-LRKCCA and LeMan, for which, except for the k or/and the K, there are additional kernel and regularization parameters needed to be adjusted, accordingly, leading to heavy cost during establishing the mapping.

Fig. 1. Network deployment of fire alarming WSN of ancient city wall.

C. Comparison in Fire alarming WSN of ancient city wall

1) System deployment: It is very important to protect the ancient city walls which are hundreds (even thousands) of years old. However, installing the modern fire protection facilities would be possible to damage the historical sites. Luckily, due to the convenient deployment and powerful function of wireless sensors, we can deploy the WSN on and in the ancient city walls without any damage. If some sensors monitor the high temperature or smoke, they will send the fire alarm. Thus, it is very important to hold the position of each sensor. However, in many large scale ancient building complexes, some factors, for example, heavy raining would make the signal transmitted in a short range, thick city wall or landslide would make the failure transmission of signal or GPS, etc., make it difficult to gather completely paired data. As a consequence, for such a scenario, how to achieve high localization accuracy with such the data is a key task in this project.

In this experiment, the network is deployed in 99 square meters region over which sensors are randomly distributed (Fig. 1). We adopt the products of Crossbow as the hardware platform, its Mica2 as sensors, MIB520 as the gateway, and then assemble the Mote, computer and gateway as the base station which manages the localization. Four APs are deployed uniformly in this network, meaning that each RSS vector is 4-dimensional. There are 850 samples collected in total. As shown in Fig. 1, those sensors are spread on the floor or seat stones. Due to that the deployed sensors lie basically in the same level floor, we characterize the physical positions just using two-dimension coordinates.

2) Experimental results: The localization performance is illustrated in Fig. 2, from which we can observe that the mean error distances of each algorithm increase with the ratio of missing data increased, and PPLCA outperforms all the compared algorithms in all the missing cases of the two spaces (from no missing to 50% missing). For the other algorithms, from Fig. 2, we can find that they perform differently in localization in different missing cases and none of them can always dominate: LCA has shorter mean error distances than both Semi-LRKCCA and LeMan when the missing ratios of data are small, for example, RSS missing ratio $\leq 15\%$ with no position missing (Fig. 2(a)) and RSS missing ratio $\leq 7\%$ with 10% position missing (Fig. 2(b)) but yields lower mean error distances in other missing cases for which, Semi-LRKCCA leads ahead contrarily. Naturally LeMan seats between them.

D. The WSN in HKUST

1) System deployment: In this sub-section, we perform a series of location estimation experiments with the data coming from HKUST. The network set-up is equipped with an IEEE 802.11b wireless network in the 2.4GHz frequency bandwidth. All data are collected using an IBM laptop computer with an external Linksys Wireless-B USB network adapter [24].

Different from the network in the previous sub-section, it is deployed in a three dimensional indoor space, in which all non-AP sensors are distributed on the same floor and 8 APs are set on the up-middle-bottom three floors. Besides, the number of sensors is more, and thus these differences naturally form a different topology for this deployment. In this experiment, there are 2,999 samples collected in total.

2) Experimental results: Applying all the same algorithms to the network, the performance comparisons for all the missing ratios of data (from 0% to 50%) are illustrated in Fig. 3, from which we can observe that the mean error distances of all the algorithms increase gradually as the missing (position and RSS) ratios increase, PPLCA still outperforms the other algorithms in all the missing cases and the performance of the other algorithms is similar to those in the first network, except for some points of Semi-LRKCCA's.

E. Discussion on the above two experiments

From above two sets of experimental results in both sub-section C and D, we can observe that (1) on the whole, PPLCA has lower mean error distances than Semi-LRKCCA, LCA and LeMan. (2) LCA can keep its shorter mean error distances than Semi-LRKCCA and LeMan when the unpaired ratios are small, but rises when the unpaired ratio becomes larger due to just using paired data. (3) Semi-LRKCCA behaves relatively badly when the unpaired ratios are small, but relatively well as the ratio becomes larger. In addition, it needs tune so-many parameters and this undoubtedly lowers its stability and applicability. (4) LeMan lies in mean error distances in-between Semi-LRKCCA and LCA since it is designed just for the case that the number of RSSs is greater than or equal to the number of physical positions.

VI. Conclusion

In this paper, we mainly solve the problem of localization in complicated WSNs, which results from the incompletely paired (RSS, Position) data due to some undesired reasons. To attack the problem, we present a novel algorithm, which is called PPLCA, and successfully apply it to the WSN localization. Our algorithm can more accurately estimate physical location of an unknown sensor in a complex environment due to directly embedding the local structures or geometrical topologies of those networks into both the objective and constraints of the LCA simultaneously, and greatly reduce the time and energy resources to enhance its practical applicability due to linear computation and only needing to tune less parameters involved. Experiments on two different environments show

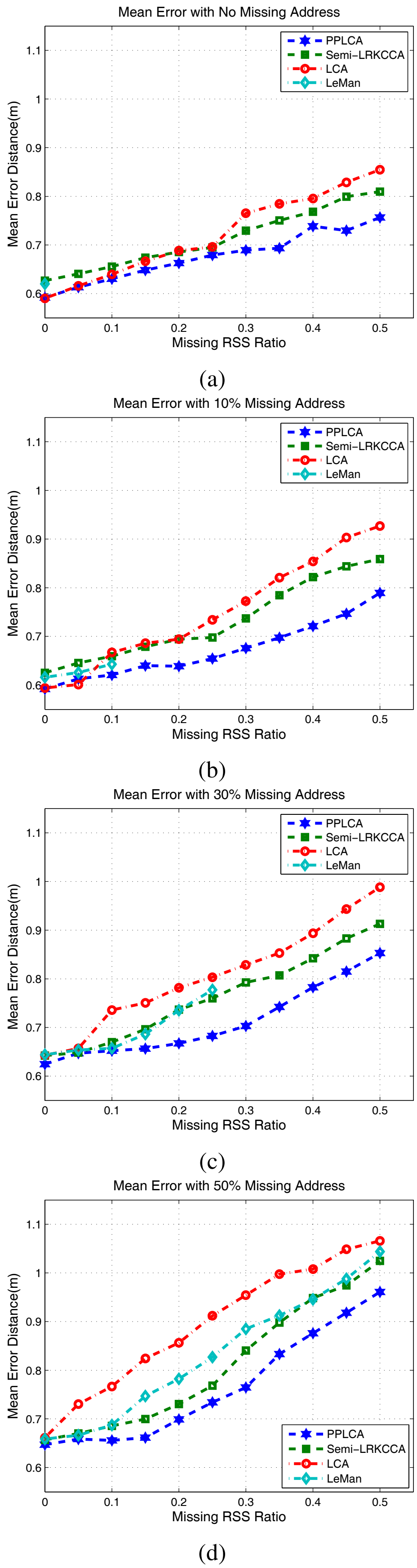

Fig. 2. Mean error distances vs. the ratios of missing data—(a), (b), (c), (d): missing physical position ratios are 0%, 10%, 30%, 50%.

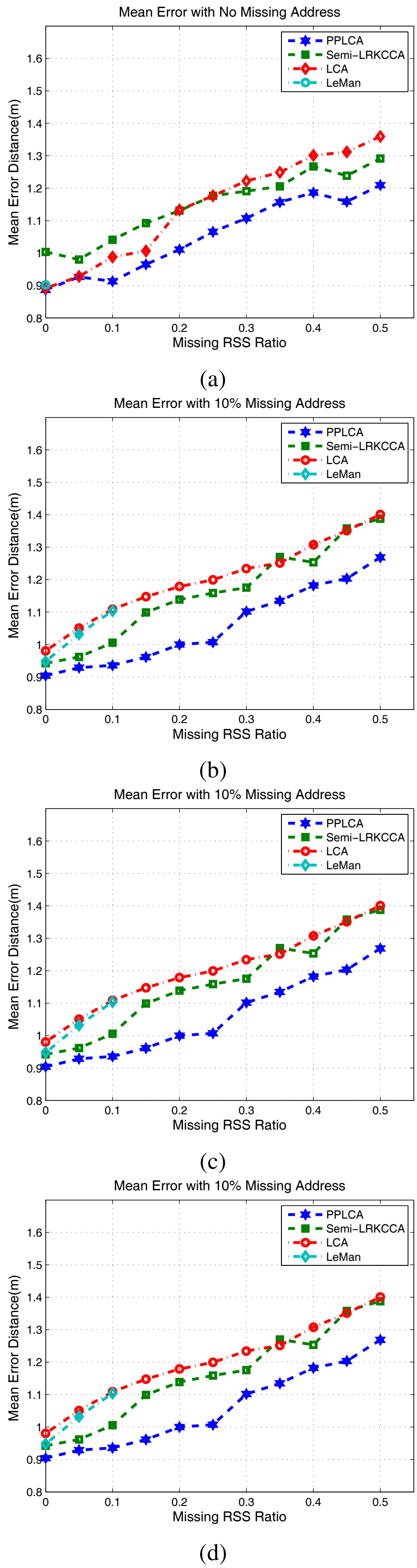

Fig. 3. Mean error distances vs. the ratios of missing data—(a), (b), (c), (d): missing physical position ratios are 0%, 10%, 30%, 50%.

its better performance in localization than those compared algorithms.

Our future work is about trajectory prediction of mobile nodes. We will also consider the networks under much complicated or easily-attacked circumstance, where the mapping need to be updated in some dynamic WSNs or the information is easy to be juggled even if gathered completely.

ACKNOWLEDGMENT

We thank very much Department of Computer Science of the Hong Kong University of Science and Technology for providing the experimental data and professor Yi Zhuang for scientific discussion. This work was partly supported by National Natural Science Foundations of China under grant Nos. 60973097 , Nos. 60803049 and Natural Science Foundations of Jiangsu Province SBK201122668.

REFERENCES

[1] J. Bruck, J. Gao, and A. Jiang, "Localization and routing in sensor networks by local angle information," *ACM Trans. Sensor Networks*, vol. 5, pp. 181–192, 2005.

[2] B. Dil, S. Dulman, and P. Havinga, "Range-based localization in mobile sensor networks," *Lecture Notes in Computer Science*, vol. 3868, pp. 164–179, 2006.

[3] D. Niculescu and B. Nath, "DV based positioning in ad hoc networks," *J. Telecommun. Syst.*, vol. 22, pp. 267–280, 2003.

[4] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, "Localization from mere connectivity," in *Proc. 4th ACM Int'l Symp. on Mobile Ad Hoc Networking & Computing*, pp. 201–212, 2003.

[5] R. Stoleru, T. He, J. A. Stankovic, and D. Luebke, "Spotlight, a high accuracy, low-cost localization system for wireless sensor networks," in *Proc. 3rd ACM Conf. on Embedded Networked Sensor Systems*, pp. 13–26, 2005.

[6] J. Krumm, G. Cermak, and E. Horvitz, "RightSPOT, a novel sense of location for a smart personal object," in *Proc. Ubicomp*, pp. 36–43, 2003.

[7] L. Girod and D. Estrin, "Robust range estimation using acoustic and multimodal sensing," in *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, pp. 1312–1320, 2001.

[8] D. W. Chin and Y. L. Han, "Hybrid TOA/AOA estimation error test and non-line of sight identification in wireless location," *Wireless Commun. and Mobile Comput.*, vol. 9, no. 6, pp. 859–873, 2009.

[9] S. P. Joseph and J. W. Anthony, "Localization of networks using various ranging bias models," *Wireless Commun. and Mobile Comput.*, vol. 8, no. 5, pp. 553-562, 2008.

[10] X. L. Li, H. C. Shi, and Y. Shang, "A sorted RSSI quantization based algorithm for sensor network localization," in *Proc. 2005 11th Int'l Conf. on Parallel and Distributed Systems*, pp. 557–563, 2005.

[11] X. L. Nguyen and M. I. Jordan, "Kernel-based learning approach to ad hoc sensor network localization," *ACM Trans. Sensor Networks*, vol. 1, no. 1, pp. 134–152, 2005.

[12] J. J. Pan, Y. Qiang, H. Chang, and D. Y. Yeung, "A manifold regularization approach to calibration reduction for sensor-network based tracking," *American Association for Artificial Intelligence*, vol. 1, pp. 988–993, 2006.

[13] C. Gentile, "Distributed sensor location through linear programming with triangle inequality constraints," *IEEE Trans. Wireless Commun.*, vol. 6, no. 7, pp. 2572–2581, 2007.

[14] A. J. Weiss and J. S. Picard, "Network localization with biased range measurements," *IEEE Trans. Wireless Commun.*, vol. 7, no. 1, pp. 298–304, 2008.

[15] C. Q. Wang, J. M. Chen, and Y. X. Sun, "Sensor network localization using kernel spectral regression," *Wireless Commun. and Mobile Comput.*, vol. 10, pp. 1045–105, 2010.

[16] A. T. Duc, N. XuanLong, and N. Thinh, "Localization problem in sensor networks, the machine learning approach," in *Localization Algorithms and Strategies for Wireless Sensor Networks, Monitoring and Surveillance Techniques for Target Tracking*, IGI Global, chapter XII, 2009.

[17] Y. Zhang, L. Zhang, and X. M Shan, "Robust distributed localization with data inference for wireless sensor network," in *Proc. ICC 2008*, pp. 3125–3129.

[18] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization, a geometric framework for learning from labeled and unlabeled examples," *J. Machine Learning Research*, vol. 7, pp. 2399–2434, 2006.

[19] J. J. Gu and S. C. Chen, "Manifold-based canonical correlation analysis for wireless sensor network localization," *Wireless Commun. and Mobile Comput.*, DOI: 10.1002/ wcm. 1071, 2011.

[20] M. B. Blaschko, C. H. Lampert, and A. Gretton, "Semi-supervised Laplacian regularization of kernel canonical correlation analysis," *Lecture Notes in Artificial Intelligence*, vol. 5211, pp. 133–145, 2008.

[21] T. Melzer, M. Reiter, and H. Bischof, "Appearance models based on kernel canonical correlation analysis," *Pattern Recognition*, vol. 36, pp. 1961-1971, 2003.

[22] J. F. Hair, R. E. Anderson, R. L. Tatham, and W. C. Black, "Canonical correlation analysis," chapter 8, *Multivariate Data Analysis*, 5th edtion, 1998.

[23] S. Taylor and J. Cristianini, *Kernel Methods for Pattern Analysis*, chapter 2, pp. 168–176. Cambridge University Press, 2004.

[24] J. J. Pan, J. T. Kwok, Q. Yang, and Y. Q. Chen, "Multidimensional vector regression for accurate and low-cost location estimation in pervasive computing," *IEEE Trans. Knowledge and Data Engineering*, vol. 9, pp. 1181–1193, 2006.

[25] J. T. Tenenbaum, V. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319–2323, 2000.

[26] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proc. 14th Int'l Joint Conf. on Artificial Intelligence*, vol. 2, pp. 1137–1143, 1995.

Jingjing Gu received her B.Sc. degree from Nanjing University of Aeronautics and Astronautics (NUAA), China, in 2001, and received the Ph.D. in computer science from NUAA, China, in 2011. Her research interests focus on Wireless Network and Data Analysis.

Songcan Chen received the B.Sc. degree in mathematics from Hangzhou University (now merged into Zhejiang University) in 1983.In December 1985, he completed the M.Sc. degree in computer applications at Shanghai Jiaotong University and then worked at Nanjing University of Aeronautics and Astronautics (NUAA) in January 1986 as an assistant lecturer. There he received a Ph.D. degree in communication and information systems in 1997. Since 1998, as a full professor, he has been with the Department of Computer Science and Engineering at NUAA. His research interests include pattern recognition, machine learning and neural computing. In these fields, he has authored or coauthored over 120 scientific journal papers.

Tingkai Sun received his B.Sc. degree from Nanjing University, Nanjing, China, in 1996. In 1999 he completed the M.Sc. degree in Nanjing University, and in 2007 he received his Ph.D. degree in computer applications from Nanjing University of Aeronautics and Astronautics. Now as an associate professor he works at School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing, China. His current research interests include pattern recognition with its applications, multivariant data analysis and machine learning.