

Article

# A LS-SVM based Measurement Points Classification Algorithm for Adjacent Targets in WSNs

Xiang Wang <sup>1</sup>, Zong-Min Zhao <sup>1,\*</sup> , Tao Wang <sup>1</sup>, Zhun Zhang <sup>1</sup>, Qiang Hao <sup>1</sup> and Xiao-Ying Li <sup>2</sup>

<sup>1</sup> School of Electronic and Information Engineering, Beihang University, Beijing 100191, China; wxiang@buaa.edu.cn (X.W.); wt860122@buaa.edu.cn (T.W.); microzhun@buaa.edu.cn (Z.Z.); haoqiang1994@buaa.edu.cn (Q.H.)

<sup>2</sup> Institute of Medical Information, Chinese Academy of Medical Sciences, Beijing 100020, China; lixiaoying@imicams.ac.cn

\* Correspondence: zhaozongmin@buaa.edu.cn

Received: 12 November 2019; Accepted: 14 December 2019; Published: 16 December 2019



**Abstract:** In wireless sensor networks (WSNs), the problem of measurement origin uncertainty for observed data has a significant impact on the precision of multi-target tracking. In this paper, a novel algorithm based on least squares support vector machine (LS-SVM) is proposed to classify measurement points for adjacent targets. Extended Kalman filter (EKF) algorithm is firstly adopted to compute the predicted classification line for each sampling period, which will be used to classify sampling points and calculate observed centers of closely moving targets. Then LS-SVM algorithm is utilized to train the classified points and get the best classification line, which will then be the reference classification line for the next sampling period. Finally, the locations of the targets will be precisely estimated by using observed centers based on EKF. A series of simulations validate the feasibility and accuracy of the new algorithm, while the experimental results verify the efficiency and effectiveness of the proposal.

**Keywords:** wireless sensor networks (WSNs); measurement origin uncertainty; localization and tracking; least square support vector machine (LS-SVM)

## 1. Introduction

Nowadays, with the development of precision sensors to monitor physical or environmental conditions, the applications of WSNs have been motivated by various industrial and consumer systems, as well as military usages [1–3]. Among the increasing research and applications of WSNs, an important field is target localization and tracking [4,5]. Generally, the location problem in WSNs falls into two categories: range-free or range-based method [6]. While the former, depending on a kind of communication device (such as ZigBee, WiFi, Bluetooth, RF, and so on), has exhibited the advantage of easy implementation without additional ranging sensors, the latter, tracking targets based on the distance measurement sensors and angle sensor, demonstrates much better positioning accuracy. In this paper, we will mainly focus on the range-based localization method.

Broadly speaking, the traditional methods of localization and tracking contain Kalman filter (KF), extended-KF (EKF), Unscented-KF and so on [7–9]. The basic idea of these algorithms is to construct suitable state equations and predict the location of targets for the next moment. Meanwhile, the observed values will also be obtained from sensor systems, which are then utilized to estimate target positions combined with predicted values. These algorithms are well suited to the case where one object corresponds to one measurement per scan. However, in WSNs, it is quite common for one target to generate multiple measurements, which is generally called extended target. There are many reasons for this phenomenon, such as the improvement of sensor resolution, the target being close to the sensor,

and multiple sensors working together. In WSNs, there exist some special features for acquisition of observation values, which is a problem for the measurement of uncertain origin [10]. Although multiple sensors have been applied to observe several targets in WSNs, it is still not clear whether a measurement point belongs to a specific target or not, especially for adjacent targets. Obviously, in order to get precise observation and estimation value for targets, it is a major requisite to accurately classify these sampling points.

In general, there are two kinds of state-of-the-art methods dealing with sampling points for respective targets: with or without the probability model [11–15]. While the computational complexity of the former increases as the model becomes more complicated, the latter saves a large number of complex calculations and includes two different methods: clustering and classification algorithms. Clustering is a process of unsupervised machine learning without the training of sample data, and group target data set by designing a suitable distance metric [16,17]. Besides, the clustering algorithm is fairly effective and suitable for acquiring observations in WSNs. However, in cases where targets move closely and their sampling points overlap each other, the performance of the clustering algorithm would dramatically decrease due to different distributions of these overlapped points. In contrast, the classification algorithms have favorable stability, and the most representative one among them is SVM algorithm. As SVM avoids the traditional process from generalization to deduction, it largely simplifies the usual classification problem. Especially, LS-SVM optimizes the algorithm solution and improves computing efficiency [18,19]. Nevertheless, being a supervised machine learning algorithm, LS-SVM trains the sample data before classification and has been rarely used in applications with dynamic systems, such as tracking problems in WSNs. In this paper, a novel maneuvering targets classification algorithm is developed by combining LS-SVM and EKF to classify sampling points, which are then used to calculate the observed centers for targets and evaluate the positions of targets based on the EKF algorithm.

The remainder of this paper is organized as follows. Section 2 mainly describes the related works. In Section 3, we describe the principles of the proposed algorithm in detail. Section 4 is about the simulation that verifies the classification and tracking performance, followed by the experiment and discussion about the proposal in Section 5. At last, Section 6 concludes this paper.

## 2. Related Work

Recently, with the improvement of sensor resolution, the problem of extended target recognition and positioning has attracted a lot of research attention. One of the key issues is how to identify two adjacent extended targets and locate them separately.

Maximum Likelihood Probabilistic Multi-Hypothesis Tracker (ML-PMHT) can reliably discriminate between the target and clutter, based on the calculation of peak point's PDF in ML-PMHT log-likelihood ratio (LLR) due to clutter and also the peak point's Probability Density Function (PDF) in LLR due to the target. In order to track the original target from measurement points generated by both the original target and interfering target, Steven Schoenecker replaced the clutter with a second, independently parameterizable target, on the basis of ML-PMHT [20]. However, sometimes the peak of the LLR surface that it generates can merge with the original target peak, and the ability to track the original target is limited, especially when an interfering target is strong enough or close enough. Based on previous work, [21] developed a statistical method to watch how close two targets were to each other, and the system still had the ability to recognize them, by checking whether the LLR peaks of the two targets coincided. Li proposed an improved partitioning algorithm for a Gaussian inverse Wishart probability hypothesis density (GIW-PHD) filter to solve the problem where the sub-partitioning algorithm failed to handle cases where targets were of different sizes, and the Mahalanobis distances was employed to distinguish among measurement cells of different sizes for extended targets [22]. However, this approach seems to not be sensitive to either differences in target size or target maneuvering.

As the applications of extended targets quickly increase, the research on target location is gradually developed for the method based on clustering. The probability-based method is used to classify each measurement point, and the great number of equivalent measurement points will lead to very large computational complexity. For this issue, adaptive weight Fuzzy C-Means clustering (AWFCM) algorithm was used to track multi-targets under the cross tracks situation [23], where a distance in new metric space was defined to restrain the error range of data association clustering centers for samples with noise points and cross tracks. Besides, the improved weights dependent on dots' density were introduced to reduce the negative effect of imbalanced data sets. Another method of Clustering-Based Multiple Hypothesis Multi-target Localization was presented for the same purpose, which divided a large number of hypotheses into several independent clusters, as well as hypothesis update, hypothesis pruning and global hypothesis generation were completed independently in each cluster [24]. Additionally, a clustering method-based Evolutionary Kernel Fuzzy C-Means clustering (EKFCM) algorithm was adopted to identify and locate targets oriented to multi-target tracking in wireless sensor networks [13], which firstly applied the clustering number recognition algorithm to filter out outliers and then calculate initial cluster centers based on the density of each measurement data. At last, the density factors of these measurement data were fused into the Gaussian kernel function to improve the accuracy of the cluster center, and the accurate position of each target at the current moment was calculated according to the predictive position and measurement data set of the corresponding cluster.

Relatively speaking, the clustering algorithm could quickly and effectively classify the measurement points and track the measurement points belonging to each extended target, so as to solve the problem of measurement origin uncertainty. The accuracy of the clustering algorithm to classification of measurement points will directly affect the accuracy of the target location. However, when two objects are close to each other, the overlapping phenomenon of measurement points will result in a classification error of the clustering algorithm and affect the positioning accuracy. In order to solve this problem, we use the classification method based on LSSVM to classify the measurement points for adjacent targets, which will further improve the accuracy of the target location.

### 3. The Proposed Algorithm

In this section, we will describe the proposed algorithm in detail.

#### 3.1. Algorithm Overview

As shown in Figure 1, the whole process of our new algorithm consists of four portions: initialization, predicting the classification line, obtaining the best classification line, and target tracking. Initialization provides the initial values for EKF and the original sampling points for LS-SVM algorithm, as well as the best classification line. The predicted positions of targets will be calculated based on the EKF algorithm and then applied to predict the classification line. Besides, the measurement points sampled by sensors in WSNs with noise points filtered out by using some simple filter would be classified by the predicted classification line. Subsequently, the geometric centers of these classified points will be regarded as observed centers of targets, which could be adopted to estimate the positions of targets based on EKF algorithm. Meanwhile, the classified points will also be devoted to train the best classification line based on LS-SVM algorithm, which will be taken as the reference classification line for the next sampling period.

Here, to facilitate the description of the algorithm, we make a few points in advance. Firstly, the classification line is used hereinafter to replace the classification surface, as it is a straight line in a 2-D area. Secondly, because the sampling points always contain valid measurement points of targets and also invalid noise points, we will filter out the noise points far from targets based on the density of the sampling points. Thus, the sampling points can be considered as valid measurement points after filtering. Finally, since it is the most common case that two targets move closely compared with more targets, we will take two targets as the example to show the principle of the proposed algorithm.

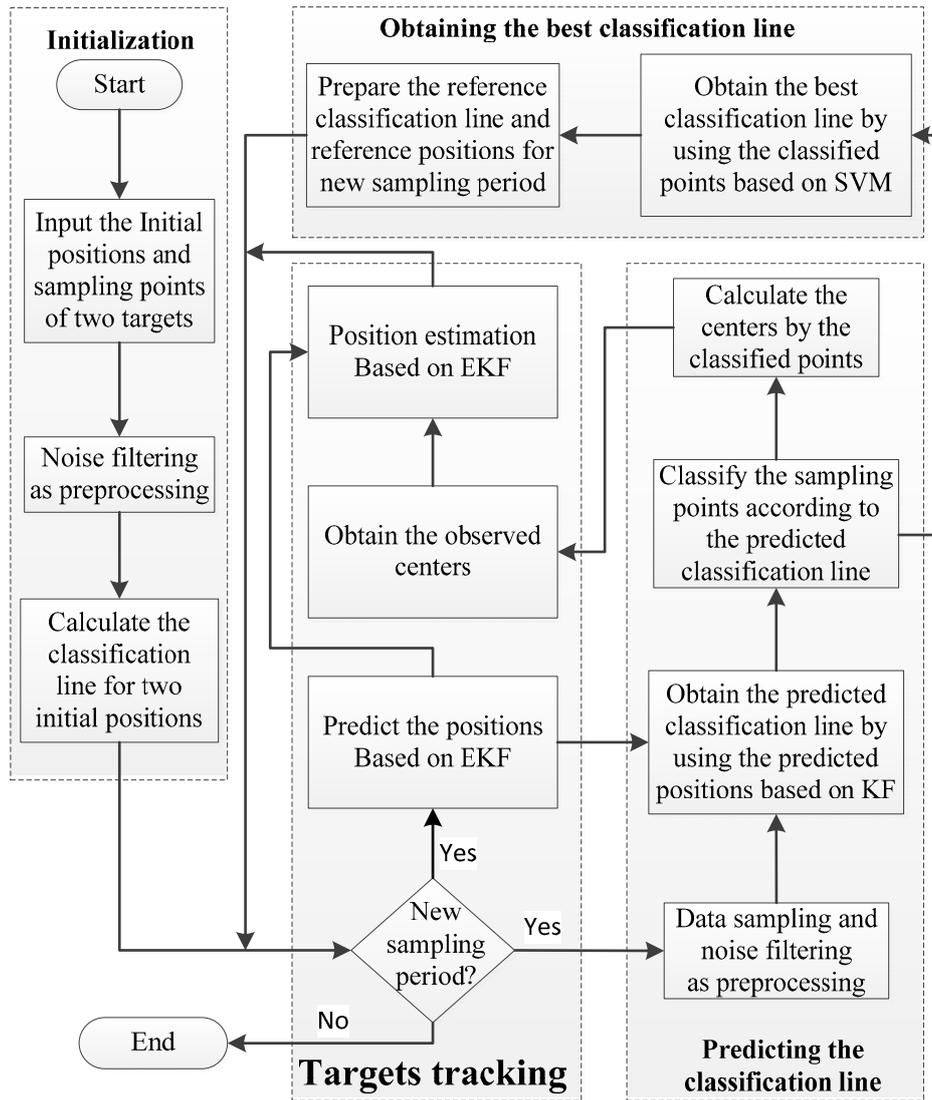


Figure 1. Flowchart of the proposed algorithm.

### 3.2. EKF algorithm for Target Tracking

In order to guarantee the algorithm works well under nonlinear systems, EKF algorithm is utilized to track targets. Firstly, we build a state model for each target:

$$\mathbf{Z}(k) = [\mathbf{x}(k) \ \dot{\mathbf{x}}(k) \ \mathbf{y}(k) \ \dot{\mathbf{y}}(k)]^T \quad (1)$$

where  $x(k)$  and  $y(k)$  represent the x-axis and y-axis positional coordinates of the target at time  $k$  respectively,  $\dot{x}(k)$  and  $\dot{y}(k)$  indicate the speed of the target along the x-axis and y-axis directions at moment  $k$ . According to EKF, the state equation of the target is:

$$\mathbf{Z}(k+1) = \Phi(k+1|k)\mathbf{Z}(k) + \mathbf{G}(k)\mathbf{W}(k) + \phi(k) \quad (2)$$

where  $\Phi(k+1|k)$  represents the state transition matrix, and  $\mathbf{G}(k)$  and  $\mathbf{W}(k)$  denote the noise driving matrix and input white noise. Here,  $\phi$  is a nonrandom term. Besides, the initial value is set as:

$$\mathbf{Z}(0) = E[\mathbf{Z}(0)] \quad (3)$$

The prediction equation is:

$$\widehat{\mathbf{Z}}(k+1|k) = f(\widehat{\mathbf{Z}}(k|k)) \quad (4)$$

and the prediction covariance matrix is:

$$\mathbf{P}(k+1|k) = \Phi(k+1|k)\mathbf{P}(k|k)\Phi^T(k+1|k) + \mathbf{Q}(k+1) \quad (5)$$

where  $\mathbf{Q}$  is the variance of  $\mathbf{W}$ , and the initial value of the prediction covariance is:

$$\mathbf{P}(0) = \text{var}[\mathbf{Z}(0)] \quad (6)$$

The predicted positions of targets will be adopted to predict the classification line, and the observed centers  $\mathbf{O}(k+1)$  of targets will be obtained from Section 3.3. The estimation equation is:

$$\widehat{\mathbf{Z}}(k+1|k+1) = \widehat{\mathbf{Z}}(k+1|k) + \mathbf{K}(k+1)\left[\mathbf{O}(k+1) - h\widehat{\mathbf{Z}}(k+1|k)\right] \quad (7)$$

$$\mathbf{K}(k+1) = \mathbf{P}(k+1|k)\mathbf{H}^T(k+1)\left[\mathbf{H}(k+1)\mathbf{P}(k+1|k)\mathbf{H}^T(k+1) + \mathbf{R}(k+1)\right]^{-1} \quad (8)$$

$$\mathbf{P}(k+1) = [\mathbf{I} - \mathbf{K}(k+1)\mathbf{H}(k+1)]\mathbf{P}(k+1|k) \quad (9)$$

$$\mathbf{H}(k) = \left. \frac{\partial h}{\partial \mathbf{X}(k)} \right|_{\mathbf{x}(k)=\widehat{\mathbf{x}}(k)} \quad (10)$$

where  $h$  denotes the observation matrix.

### 3.3. Predicting the Classification Line

In this subsection, we use the predicted positions of targets from Section 3.2 to predict the classification line, which will be devoted to classify sampling points, calculate observed centers, and also estimate positions of targets. Figure 2 shows the geometric relationship between the best classification line at time  $k$  and the predicted classification line at moment  $k+1$ . Here,  $A_{k|k}$  and  $B_{k|k}$  are the geometric centers of classified points, and  $l_{k|k}$  is the best classification line at time  $k$ , which is also the reference classification line at  $k+1$ .  $C_k$  is the intersection point between  $l_{k|k}$  and connection line  $A_{k|k}B_{k|k}$ . Similarly,  $A_{k+1|k}$  and  $B_{k+1|k}$  are the predicted positions of targets at time  $k+1$ .  $l_{k+1|k}$  is the predicted classification line to be calculated below, with  $C_{k+1}$  the intersection point between  $A_{k+1|k}B_{k+1|k}$  and  $l_{k+1|k}$ . The specific process holds six steps.

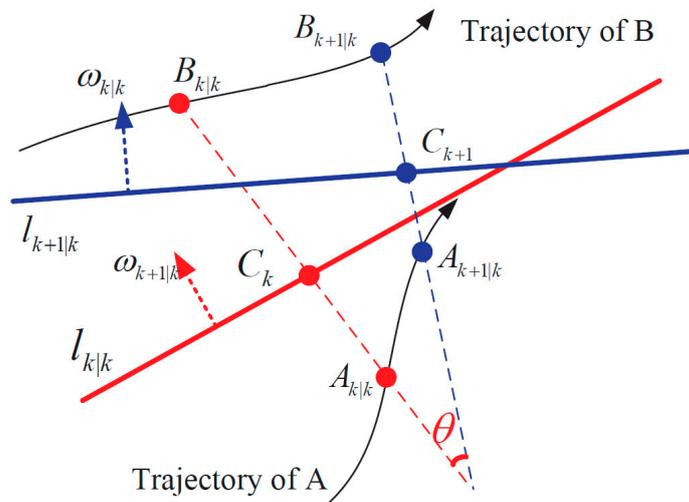


Figure 2. Transformation for the classification line.

**Step 1:** Input the initial parameters.

The initial parameters mainly refer to the best classification line and observed centers at the very beginning of the algorithm. For the following sampling periods, the best classification line at time  $k$  will be regarded as the reference classification line at moment  $k + 1$ . Later, after predicting targets' positions at time  $k + 1$  based on EKF, the predicted classification line at time  $k + 1$  will be obtained by transforming the reference classification line according to the positional relationship between predicted targets positions at time  $k + 1$  and observed targets centers at  $k$  (through following steps 2–5).

**Step 2:** Calculate the intersection point  $C_k$ .

The intersection point  $C_k(x_{k|k}^{(C)}, y_{k|k}^{(C)})$  could be calculated by solving the straight line equations below:

$$\omega_{k|k} \cdot [x_{k|k}, y_{k|k}]^T + b_{k|k} = 0 \quad (11)$$

$$y_{k|k} = y_{k|k}^{(A)} + \frac{y_{k|k}^{(B)} - y_{k|k}^{(A)}}{x_{k|k}^{(B)} - x_{k|k}^{(A)}} [x_{k|k} - x_{k|k}^{(A)}] \quad (12)$$

where (11) is the equation of the best classification line  $l_{k|k}$  with normal vector  $\omega_{k|k}$ , and (12) is the equation of line  $A_{k|k}B_{k|k}$ . We would also compute the scaling factor  $\alpha_s$ , which denotes the proportion of point  $C_k$  between  $A_{k|k}$  and  $B_{k|k}$ .

$$\alpha_s = \frac{y_{k|k}^{(C)} - y_{k|k}^{(B)}}{y_{k|k}^{(A)} - y_{k|k}^{(B)}} \quad (13)$$

**Step 3:** Compute the predicted intersection point  $C_{k+1}$ .

Assume that the relative position of predicted classification line  $l_{k+1|k}$  between two targets is persistent for two adjacent moments. Thus, the scaling factor  $\alpha_s$  for time  $k + 1$  will be equal to the one at moment  $k$ . According to predicted positions of targets  $A_{k+1|k}$  and  $B_{k+1|k}$ , we will calculate the predicted intersection point  $C_{k+1}(x_{k+1|k}^{(C)}, y_{k+1|k}^{(C)})$ :

$$\begin{bmatrix} x_{k+1|k}^{(C)} \\ y_{k+1|k}^{(C)} \end{bmatrix} = \alpha_s \begin{bmatrix} x_{k+1|k}^{(A)} \\ y_{k+1|k}^{(A)} \end{bmatrix} + (1 - \alpha_s) \begin{bmatrix} x_{k+1|k}^{(B)} \\ y_{k+1|k}^{(B)} \end{bmatrix} \quad (14)$$

**Step 4:** Calculate the normal vector of  $l_{k+1|k}$ .

The predicted classification line  $l_{k+1|k}$  would be transformed from the best classification line  $l_{k|k}$  at time  $k$ . Thus, the normal vector of  $l_{k+1|k}$  could be obtained by getting the rotation angle  $\theta$  of two lines  $AB_{k|k}$  and  $AB_{k+1|k}$ , which equals to the rotation angle between the normal vector of  $l_{k|k}$  and normal vector of  $l_{k+1|k}$ .

$$\cos \theta = \frac{\overrightarrow{AB_{k|k}} \cdot \overrightarrow{AB_{k+1|k}}}{\|\overrightarrow{AB_{k|k}}\| \|\overrightarrow{AB_{k+1|k}}\|} \quad (15)$$

$$\sin \theta = \frac{\overrightarrow{AB_{k|k}} \times \overrightarrow{AB_{k+1|k}}}{\|\overrightarrow{AB_{k|k}}\| \|\overrightarrow{AB_{k+1|k}}\|} \quad (16)$$

here  $\overrightarrow{AB_{k|k}}$  denotes the vector of point  $A_{k|k}$  to point  $B_{k|k}$ , and  $\overrightarrow{AB_{k+1|k}}$  presents the vector of point  $A_{k+1|k}$  to point  $B_{k+1|k}$  respectively. Then, the normal vector of  $l_{k+1|k}$  is:

$$\omega_{k+1|k} = \omega_{k|k} \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \quad (17)$$

**Step 5:** Obtain the predicted classification line.

The predicted classification line  $l_{k+1|k}$  can be described as below:

$$\boldsymbol{\omega}_{k+1|k} \cdot [x_{k+1|k}, y_{k+1|k}]^T + b_{k+1|k} = 0 \quad (18)$$

where  $b_{k+1|k}$  will be derived by taking point  $C_{k+1}(x_{k+1|k}^{(C)}, y_{k+1|k}^{(C)})$  into (18).

**Step 6:** Calculate the observed centers of targets.

After obtaining the predicted classification line, each sampling point is put into the following formula for classification.

$$M(\mathbf{X}_i) = \text{sgn}(\boldsymbol{\omega}_{k+1|k} \cdot [x_{k+1,i}, y_{k+1,i}]^T + b_{k+1|k}) \quad (19)$$

The sampling points will be classified into two groups, the geometric center of each group is supposed to be the observed center for each target, which could be adopted to estimate the positions of targets based on EKF (see Section 3.2).

The method for predicting the classification line is presented in Algorithm 1 below.

---

**Algorithm 1** Predicting the classification line

---

Input: best classification line and observed centers of targets at time  $k$ ;

Output: predicted classification line and observed centers of targets at moment  $k + 1$ ;

- 1: Input the best classification line and observed centers of targets at time  $k$  as initial parameters, get the reference classification line and predicted targets positions  $A_{k+1|k}$  and  $B_{k+1|k}$  at moment  $k + 1$ ;
  - 2: Calculate the intersection point  $C_k$  and scaling factor  $\alpha_s$  according to the straight line equations;
  - 3: Compute the predicted intersection point  $C_{k+1}$  on the basis of scaling factor  $\alpha_s$  and predicted positions of targets  $A_{k+1|k}, B_{k+1|k}$ ;
  - 4: Calculate the normal vector of  $l_{k+1|k}$  based on positions of targets at time  $k$  and  $k + 1$ ;
  - 5: Obtain the predicted classification line at moment  $k + 1$  with intersection point  $C_{k+1}$  and normal vector of  $l_{k+1|k}$ ;
  - 6: Calculate the observed centers of targets at time  $k + 1$  after classifying the sampling points with the predicted classification line, which will be utilized to track targets based on EKF.
- 

### 3.4. Training the Best Classification Line

By using the predicted classification line to classify the sampling points, we could effectively find the observed centers of targets at each time. However, the predicted classification line obtained at time  $k$  may sometimes bring some classification errors for next moment  $k + 1$ . Subsequently, the following classification is affected by the accumulated error before, namely misclassification. In order to reduce the impact of the accumulated error, before the closing algorithm for the current sampling period  $k$  and beginning of the new period  $k + 1$ , we will retrain the data set and get the best classification line of classified points, which will be regarded as the reference classification line for the next sampling period  $k + 1$ . In order to reduce computational complexity, we use the LS-SVM to train the best classification line. The LS-SVM turns the inequality constraints of the original SVM algorithm into equality constraints, thus the solving quadratic programming is replaced by solving linear equations [18]. According to the classified sampling point set  $\{x_i, y_i, z_i\}_{i=1}^l$  with input data  $\{x_i, y_i\}$  and corresponding binary class labels  $z_i \in \{-1, +1\}$ , the LS-SVM classifier is obtained by reformulating the minimization problem as:

$$\min_{\boldsymbol{\omega}} \Phi(\boldsymbol{\omega}, e) = \frac{1}{2}(\boldsymbol{\omega} \cdot \boldsymbol{\omega}) + \gamma \frac{1}{2} \sum_{i=1}^l e_i^2 \quad (20)$$

subject to

$$z_i(\boldsymbol{\omega} \cdot [x_i, y_i]^T + b) = 1 - e_i, i = 1, \dots, l \quad (21)$$

where  $e_i$  denotes the slack variable,  $\gamma$  presents the weight to balance best hyperplane and minimum deviation, and  $\cdot$  presents the scalar product of two vectors. Subsequently, the Lagrangian function is constructed as:

$$\mathbf{L}(\boldsymbol{\omega}, b, e, \boldsymbol{\alpha}) = \Phi(\boldsymbol{\omega}, \mathbf{e}) - \sum_{i=1}^l \alpha_i [z_i(\boldsymbol{\omega} \cdot [x_i, y_i]^T + b) - 1 + e_i] \quad (22)$$

where  $\alpha_i \in R$  denote the Lagrange multipliers. The conditions for optimality are

$$\left\{ \begin{array}{l} \frac{\partial \mathbf{L}}{\partial \boldsymbol{\omega}} = 0 \rightarrow \boldsymbol{\omega} = \sum_{i=1}^l \alpha_i [x_i, y_i], \\ \frac{\partial \mathbf{L}}{\partial b} = 0 \rightarrow \sum_{i=1}^l \alpha_i = 0, \\ \frac{\partial \mathbf{L}}{\partial e_i} = 0 \rightarrow \alpha_i = \gamma e_i, i = 1, \dots, l, \\ \frac{\partial \mathbf{L}}{\partial \alpha_i} = 0 \rightarrow z_i(\boldsymbol{\omega} \cdot [x_i, y_i]^T + b) + e_i - 1 = 0, i = 1, \dots, l. \end{array} \right. \quad (23)$$

By solving the linear Equations (23), we can get the best classification line. The method for training the best classification line is shown in Algorithm 2.

---

**Algorithm 2** Training the best classification line

---

Input: predicted classification line and observed measure point set  $\{x_i, y_i\}_{i=1}^l$  at time  $k + 1$ ;

Output: best classification line at moment  $k + 1$ ;

- 1: Input the predicted classification line and observed measure point set  $\{x_i, y_i\}_{i=1}^l$  at time  $k + 1$ ;
  - 2: Classify the observed measure point set  $\{x_i, y_i\}_{i=1}^l$  by using the predicted classification line according to Equation (19), which will get the classified sampling point set  $\{x_i, y_i, z_i\}_{i=1}^l$  with the binary class labels  $z_i \in \{-1, +1\}$ ;
  - 3: Substituting  $\{x_i, y_i, z_i\}_{i=1}^l$  into Equation (23) to compute the parameter  $\boldsymbol{\omega}$  and  $b$ , and finally get the best classification line at moment  $k + 1$ .
- 

#### 4. Algorithm Simulation and Validation

In this section, the classification and tracking performance of the presented method will be simulated for algorithm validation.

##### 4.1. Parameter Selection

The parameter  $\gamma$ , which is shown in (20), is needed to be determined before program running. Theoretically, a bigger  $\gamma$  will increase the difficulty of calculation, while a smaller one will reduce the balance effect. In order to determine a proper  $\gamma$ , we simulated a classification by using the measurement points of two targets in different overlapping ranges and optimized the value of  $\gamma/2$  from 1 to 10000. The overlap range is set from 0% to 30% for simulation, with the simulation result listing in Figure 3. The subfigure above shows the influence of the selection of  $\gamma$  on the running time and classification accuracy under 10% of the overlapped range, and the subfigure below lists the test under 30% of the overlapped range. Clearly, we get a balance between running time and classification accuracy when the  $\gamma/2$  will be assigned within 1000.

##### 4.2. Simulation of Classification Performance

In order to verify the classification performance of the proposed algorithm, we firstly simulate the classification of the two targets within 11 groups of continuous sampling periods, which is shown in Figure 4. Here,  $L$  represents the best classification line for each sampling period. The whole simulation process involves two different motion states of targets: parallel state (L2–L5) and crossing state (L9–L10). Figure 4 reveals that the proposed algorithm can basically distinguish the sampling points of two

targets, even though two targets are sometimes very close to each other. Besides, once two targets move separately (L6–L8, L11), the best classification line will effectively reduce accumulative error.

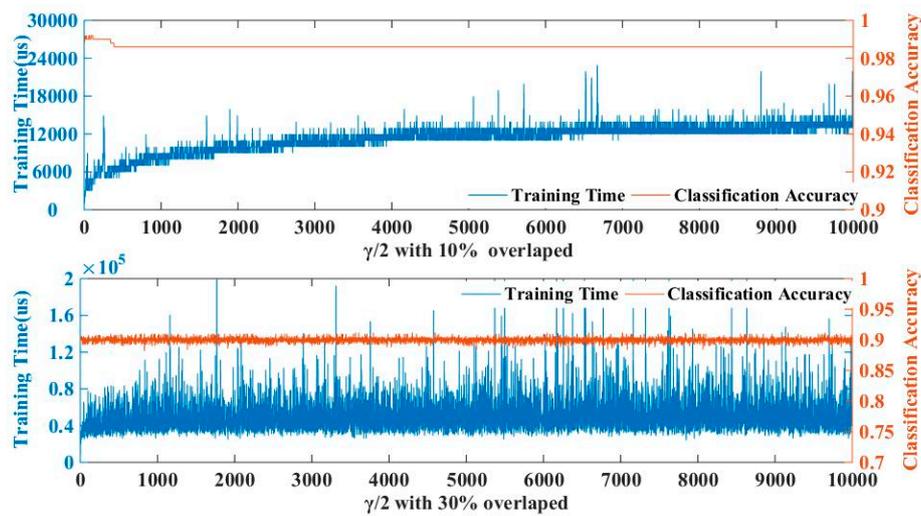


Figure 3. Parameter Selection for  $\gamma$ .

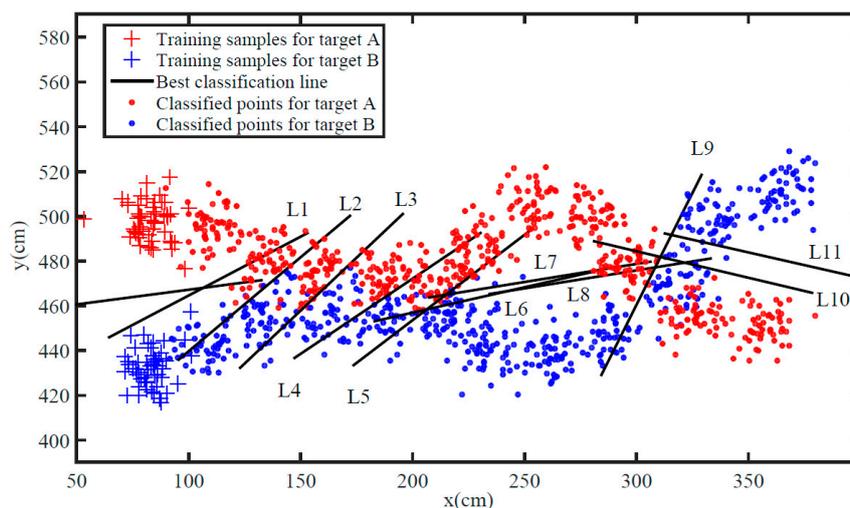


Figure 4. The classification process.

Moreover, a Monte Carlo simulation with 5000 s was taken to analyze the classification performance. We have recorded the number of misclassified points and center distance between two targets. Figure 5 reports the probabilities of misclassification over center distances of two targets. As the radius of targets is set to 20 cm, the classification performance of our proposed algorithm becomes increasingly promising while the center distance is bigger than 30 cm, especially 50 cm (while two targets completely separate from each other).

#### 4.3. Simulation of Tracking Performance

In order to evaluate the performance of the proposed algorithm for tracking adjacent targets, several simulations have been done and compared with some existing algorithms, namely K-means [16], Fuzzy C-means(FCM) [14] and EKFCM [13] clustering algorithm, which are all combined with the same tracking EKF algorithm. Besides, the root mean square error (RMSE) and Mean RMSE (MMSE) of the position are chosen as performance metrics [25].

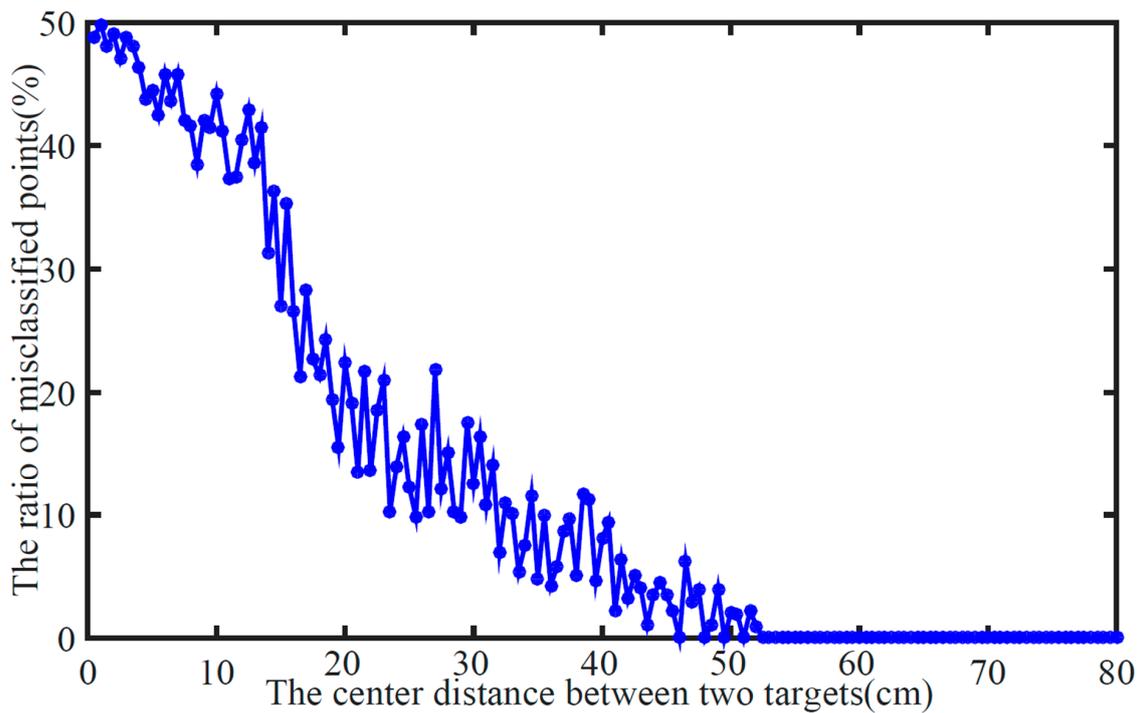


Figure 5. The misclassification probability.

The RMSE and MMSE of position are respectively defined as follows:

$$RMSE = \sqrt{\frac{1}{T} \sum_{k=1}^T ((x_k^s - \hat{x}_k^s)^2 + (y_k^s - \hat{y}_k^s)^2)} \quad (24)$$

$$MMSE = \sqrt{\frac{1}{MT} \sum_{s=1}^M \sum_{k=1}^T ((x_k^s - \hat{x}_k^s)^2 + (y_k^s - \hat{y}_k^s)^2)} \quad (25)$$

where  $(x_k^s, y_k^s)$  and  $(\hat{x}_k^s, \hat{y}_k^s)$  represent true and estimated positions at time  $k$  of the  $s$ -th Monte Carlo simulation, and  $M = 100$  and  $T = 1000$  s are the total number of Monte Carlo runs and simulation time separately. Figure 6 shows the RMSEs of K-means, FCM, EKFCM and the proposed algorithm, with Table 1 illustrating their MMSE. It is obvious that the RMSE of the proposed algorithm is almost smaller than the ones of K-means, FCM and EKFCM algorithms in each Monte Carlo run, and the MMSE of our proposal is reduced at least 20% and 10% compared with K-means and FCM algorithm. Compared with EKFCM algorithm, the MMSE of the proposed method is reduced almost 10%. Besides, we calculated the average running time of each algorithm shown in Table 1, where the proposed method takes a little longer time to solve SVM equations rather than the addition-based clustering algorithm. Therefore, in terms of feasibility and accuracy of tracking adjacent targets, the provided algorithm will be more suitable than K-means, FCM and EKFCM.

Table 1. MMSE of the proposed and existing methods for two targets.

Target	K-Means	FCM	EKFCM	Proposed Method
A (cm)	15.7163	13.7165	13.7481	12.1711
B (cm)	15.8078	14.1137	13.4989	12.4182
Improvement rate (%)	21.99	11.65	9.75	-
Average Run Time (s)	0.1534	0.1674	0.1871	0.2162

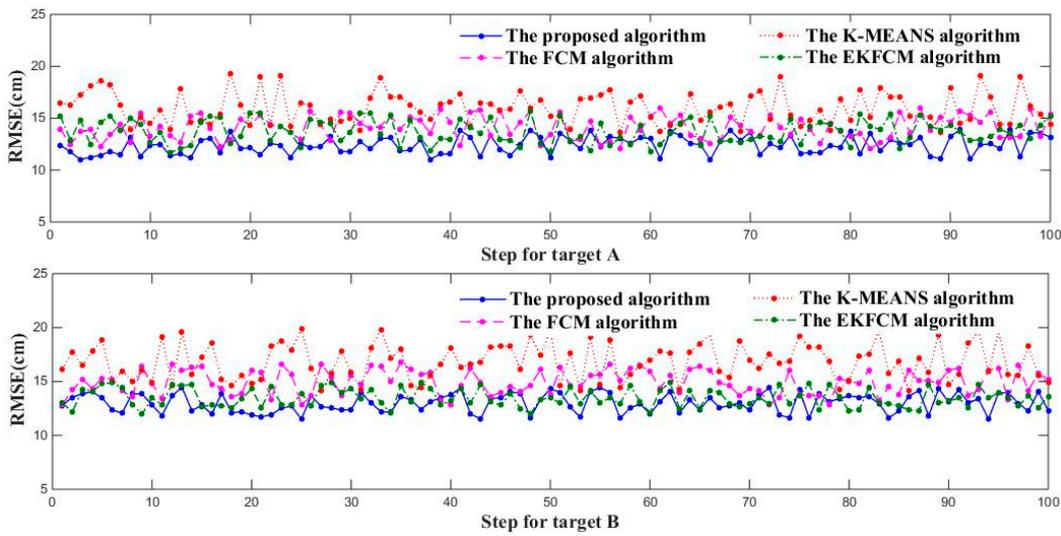


Figure 6. RMSEs across 100 Monte Carlo simulations.

5. Experimental Evaluation

Besides algorithm simulation, a series of experiments are conducted to validate the efficiency and effectiveness of our proposal. In this section, we firstly introduce the implementation of experiments and then analyze and discuss experimental results.

5.1. Experimental Implementation

A WSN platform with 12 sensor nodes was set up (Figure 7) covering a surveillance area of 7 by 7 square meters. The positional coordinate for sensor node is SN1 (0,0), SN2 (0,2.33), SN3 (0,4.67), SN4 (0,7), SN5 (2.33,7), SN6 (4.67,7), SN7 (7,7), SN8 (7,4.67), SN9 (7,2.33), SN10 (7,0), SN11 (4.67,0), and SN12 (2.33,0) in meters. Vertically, the sensor nodes are all placed at the height of 1.15 m.

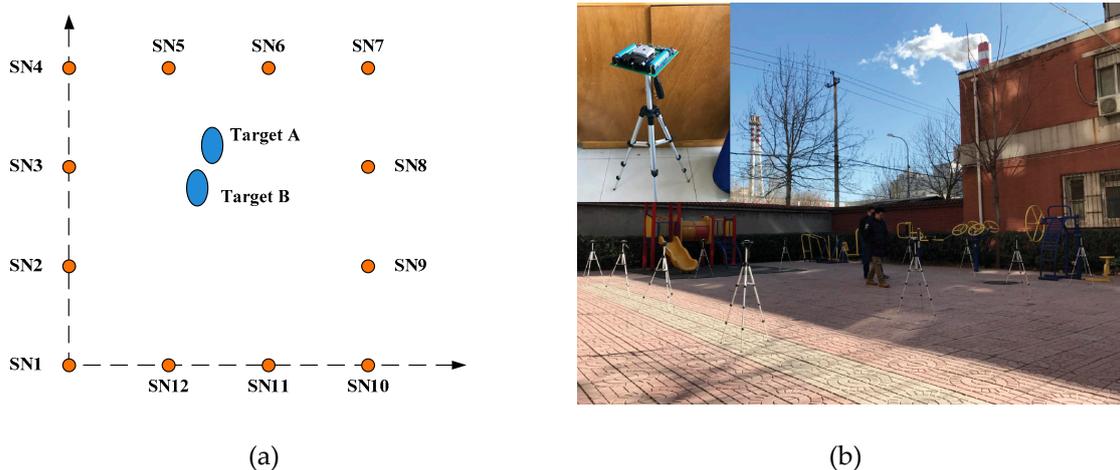


Figure 7. Experimental setup: the distribution of the sensor. (a) The distribution of sensors; (b) The experimental scene.

The sensor node consists of five parts: infrared ranging (IR) sensor, angle control module, wireless communication module, control module, and power supply module, which is shown in Figure 8. Specifically, the effective detection distance of the IR sensor is up to 5 meters, and the angle control module drives a stepping motor with a step angle of 7.2°. Here, a sampling period is defined as the time required by the sensor node to turn half a circle. Each sampling period is 0.5 s and includes 25

sampling windows. All the sampling data will be sent to the host computer by wireless communication module with model ESP8266. The control module here is hired to initialize and manage other modules. Two people are the targets of the practical experiment. Once beginning, they enter the surveillance region with a speed of approximately 1m/s and walk along the predefined routes. During the whole experiment, they will move closely with each other and even cross several times. While two persons keep walking, the sensor nodes sample target positions and then update the data to the host computer, which will be analyzed by the proposed algorithm to track adjacent targets.

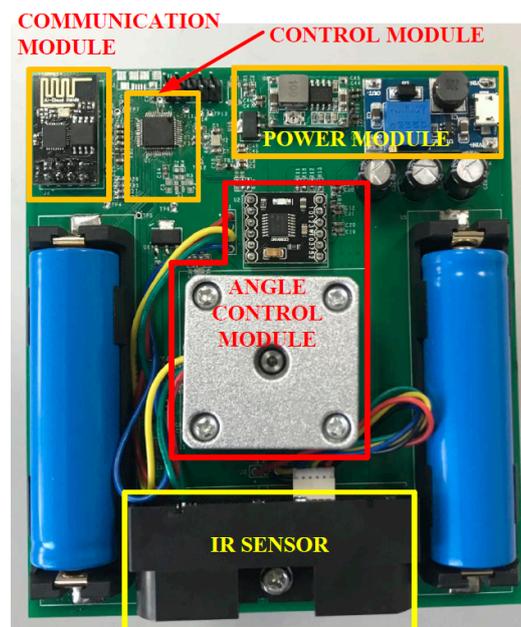


Figure 8. The details of the sensor.

Besides, in order to compare the tracking efficiency of these algorithms, it is necessary to obtain true positions of each target. Therefore, a kind of device for recording time and imprinting on the ground is installed on the shoes of every person. When people walk, a mark will be printed on the ground for each footstep as well as the exact time will be recorded by the device. After the experiment is completed, all the marks will be detected and the center between the two adjacent marks is perceived as the true position of target. Then, the true positions will be coupled with the recorded time. Finally, the actual positions for each sampling time can be calculated by the interpolation method according to the detected positions and recorded time.

## 5.2. Experimental Analysis

Based on the above experimental setup, sequence data were collected from two closely moving persons and then used to estimate the target trajectory by the presented algorithm, K-means, and FCM methods separately, see Figure 9.

We counted the positional errors (PE) between the locations from each algorithm and truth locations, as shown in Figure 10, which shows the PE of the new algorithm, and found that they are basically less than that of other methods. In order to compare the performance of these algorithms more intuitively, RMSE was calculated and shown in Table 2. Besides, we have done 60 experiments and collected 60 trajectories which include the trajectory shown in Figure 9. We have calculated the RMSE of these trajectories and shown them in Figure 11. At last, we also calculate the MMSE for these trajectories, which is shown in Table 3. It demonstrates that MMSE of the new algorithm is still lower than that of others, which keeps consistent with the results of the simulation. Moreover, under Matlab R2014b with Intel(R) Core(TM) i5-4590 CPU at 3.3 GHz as well as 8 GB RAM, the average run time of tracking is 0.328s, which will satisfy the real-time requirement of our system.

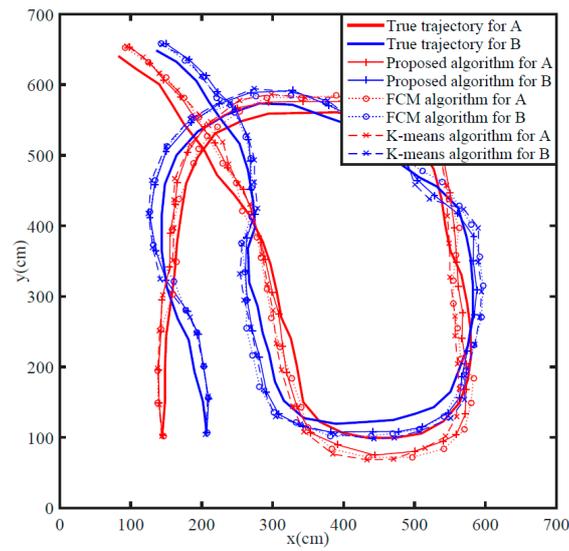
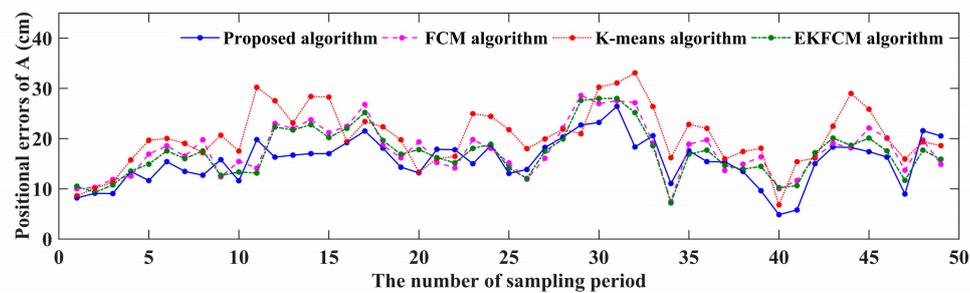
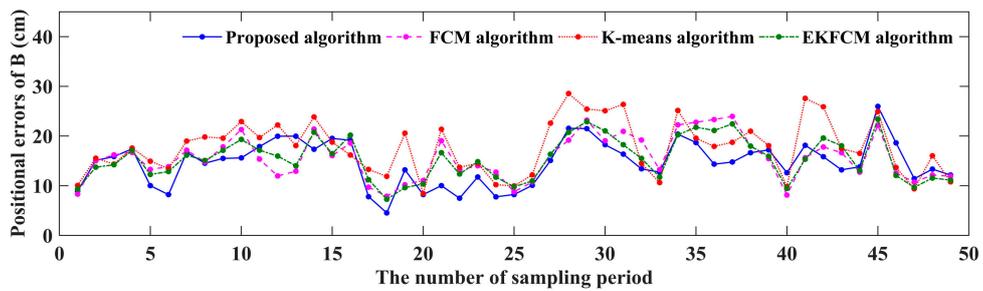


Figure 9. Tracking comparison among the algorithms.



(a)

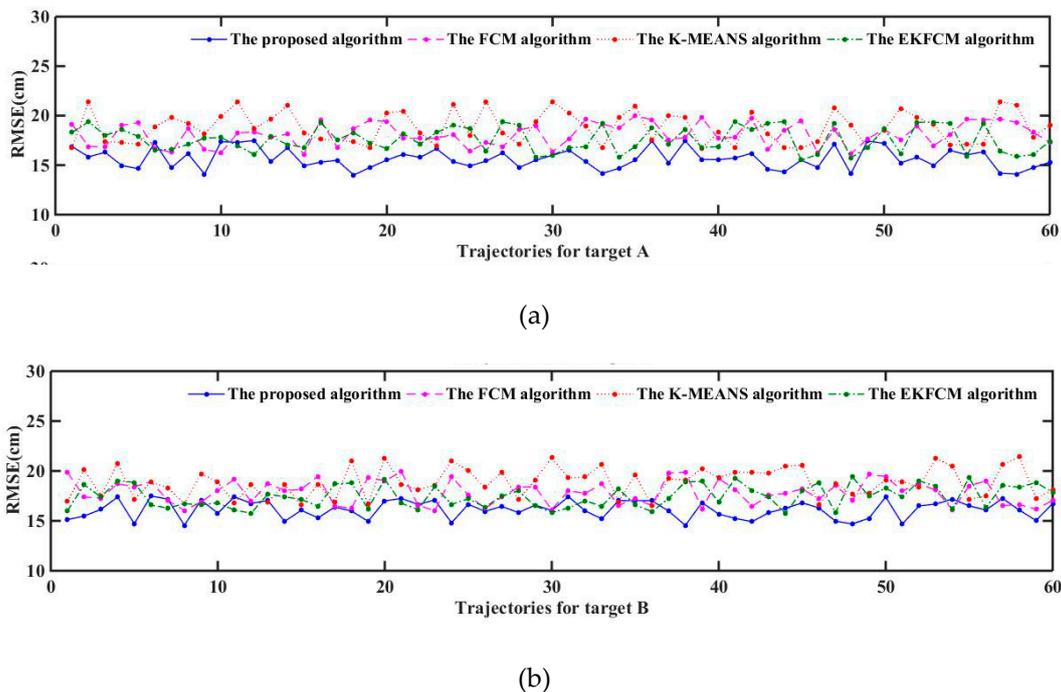


(b)

Figure 10. Positional errors for these the algorithms. (a) Positional errors with these algorithms for target A; (b) Positional errors with these algorithms for target B.

Table 2. RMSE (cm) of proposed and existing methods for the experiment.

Target	K-Means	FCM	EKFCM	Proposed Method
A	20.9915	19.1967	17.8418	16.9961
B	18.2747	16.0483	15.9778	15.4728



**Figure 11.** RMSE of 60 trajectories for targets A and B. (a) RMSE of these algorithms for Target A; (b) RMSE of these algorithms for Target B.

**Table 3.** MMSE of proposed and existing methods for the experiment.

Target	K-Means	FCM	EKFCM	Proposed Method
A (cm)	18.8562	18.1433	17.6606	15.6904
B (cm)	18.9090	17.8874	17.4856	16.1786
Improvement rate (%)	15.61	11.54	9.32	-

## 6. Conclusions and Future Work

This paper mainly focuses on the classification problem among the sampling points of two adjacent targets. The proposed algorithm combined EKF and LS-SVM methods to obtain the best classification line for each sampling period, which was then devoted to precisely classify sampling points and calculate the observed centers of targets, as well as estimate the positions for closely moving targets based on EKF. Several simulations were conducted to demonstrate the feasibility and accuracy of the presented algorithm for tracking two closely moving targets in WSNs. Moreover, in contrast with state-of-the-art K-means, FCM and EKFCM methods, experimental and quantitative results further validate the efficiency and effectiveness of our proposal. In the near future, algorithm optimization and its extension to track more than two targets will be fully investigated.

**Author Contributions:** Conceptualization, Z.-M.Z.; Methodology, Z.-M.Z.; Software, Z.-M.Z. and T.W.; Validation, T.W.; Formal analysis, X.-Y.L.; Investigation, Z.Z. and Q.H.; Resources, X.W.; Data curation, Z.Z. and Q.H.; Writing—original draft preparation, Z.-M.Z.; Writing—review and editing, X.-Y.L.; Visualization, Z.-M.Z.; Supervision, X.W.; Project administration, X.W.; Funding acquisition, X.W.; All authors read and approved the final manuscript.

**Funding:** This research was funded by National Natural Science Foundation of China (grant number 60973106), National Natural Science Foundation of China (grant number 81571142), Key Project of National Natural Science Foundation of China (grant number 61232009). And the APC was funded by Key Project of National Natural Science Foundation of China, grant number 61232009.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Chhaya, L.; Sharma, P.; Bhagwatikar, G.; Kumar, A. Wireless sensor network based smart grid communications: Cyber attacks, intrusion detection system and topology control. *Electronics* **2017**, *6*, 5. [[CrossRef](#)]
2. Mois, G.; Folea, S.; Sanislav, T. Analysis of three IoT-based wireless sensors for environmental monitoring. *IEEE Trans. Instrum. Meas.* **2017**, *66*, 2056–2064. [[CrossRef](#)]
3. Pau, G.; Salerno, V.M. Wireless sensor networks for smart homes: A fuzzy-based solution for an energy-effective duty cycle. *Electronics* **2019**, *8*, 131. [[CrossRef](#)]
4. Tomic, S.; Beko, M.; Dinis, R. 3-D target localization in wireless sensor networks using RSS and AoA measurements. *IEEE Trans. Veh. Technol.* **2016**, *66*, 3197–3210. [[CrossRef](#)]
5. ZDeshpande, N.; Grant, E.; Henderson, T.C. Target localization and autonomous navigation using wireless sensor networks—A pseudogradient algorithm approach. *IEEE Syst. J.* **2013**, *8*, 93–103. [[CrossRef](#)]
6. Jiang, J.A.; Zheng, X.Y.; Chen, Y.F.; Wang, C.H.; Chen, P.T.; Chuang, C.L.; Chen, C.P. A distributed RSS-based localization using a dynamic circle expanding mechanism. *IEEE Sens. J.* **2013**, *13*, 3754–3766. [[CrossRef](#)]
7. Kalman, R.E. A new approach to linear filtering and prediction problems. *J. Basic Eng.* **1960**, *82*, 35–45. [[CrossRef](#)]
8. Yang, Y.; Fan, X.; Zhuo, Z.; Wang, S.; Nan, J.; Xu, Y. Amended Kalman filter for maneuvering target tracking. *Chin. J. Electron.* **2016**, *25*, 1166–1171. [[CrossRef](#)]
9. Julier, S.J.; Uhlmann, J.K. Unscented filtering and nonlinear estimation. *Proc. IEEE* **2004**, *92*, 401–422. [[CrossRef](#)]
10. Bar-Shalom, Y.; Marcus, G. Tracking with measurements of uncertain origin and random arrival times. *IEEE Trans. Autom. Control.* **1980**, *25*, 802–807. [[CrossRef](#)]
11. Choi, M.E.; Seo, S.W. Robust multitarget tracking scheme based on Gaussian mixture probability hypothesis density filter. *IEEE Trans. Veh. Technol.* **2015**, *65*, 4217–4229. [[CrossRef](#)]
12. Jiang, L.; Singh, S.S.; Yıldırım, S. Bayesian tracking and parameter learning for non-linear multiple target tracking models. *IEEE Trans. Signal Process.* **2015**, *63*, 5733–5745. [[CrossRef](#)]
13. Wang, T.; Wang, X.; Zhao, Z.; He, Z.; Xia, T. Measurement data classification optimization based on a novel evolutionary kernel clustering algorithm for multi-target tracking. *IEEE Sens. J.* **2018**, *18*, 3722–3733. [[CrossRef](#)]
14. He, X.; Wang, T.; Liu, W.; Luo, T. Measurement data fusion based on optimized weighted least-squares algorithm for multi-target tracking. *IEEE Access* **2019**, *7*, 13901–13916. [[CrossRef](#)]
15. Zhang, Z.; Fu, K.; Sun, X.; Ren, W. Multiple target tracking based on multiple hypotheses tracking and modified ensemble Kalman filter in multi-sensor fusion. *Sensors* **2019**, *19*, 3118. [[CrossRef](#)]
16. Ahmed, M.N.; Yamany, S.M.; Mohamed, N.; Farag, A.A.; Moriarty, T. A modified fuzzy c-means algorithm for bias field estimation and segmentation of MRI data. *IEEE Trans. Med. Imaging* **2002**, *21*, 193–199. [[CrossRef](#)]
17. Kanungo, T.; Mount, D.M.; Netanyahu, N.S.; Piatko, C.D.; Silverman, R.; Wu, A.Y. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 881–892. [[CrossRef](#)]
18. Zhang, S.; Zhao, S.; Sui, Y.; Zhang, L. Single object tracking with fuzzy least squares support vector machine. *IEEE Trans. Image Process.* **2015**, *24*, 5723–5738. [[CrossRef](#)]
19. Suykens, J.K.; Lukas, L.; Van Dooren, P.; De Moor, B.; Vandewalle, J. Least squares support vector machine classifiers: A large scale algorithm. In Proceedings of the European Conference on Circuit Theory and Design, ECCTD; Citeseer: Stresa, Italy, 1 September 1999; Volume 99, pp. 839–842.
20. Schoenecker, S.; Willett, P.; Bar-Shalom, Y. Trackability: Resolvability of two closely-spaced targets. International Society for Optics and Photonics. In Proceedings of the Signal Processing, Sensor/Information Fusion, and Target Recognition XXV, Baltimore, MD, USA, 17–21 April 2016; Volume 9842, p. 984205.
21. Schoenecker, S.; Willett, P.; Bar-Shalom, Y. Resolution limits for tracking closely spaced targets. *IEEE Trans. Aerosp. Electron. Syst.* **2018**, *54*, 2900–2910. [[CrossRef](#)]
22. Li, P.; Ge, H.; Yang, J. Adaptive measurement partitioning algorithm for a Gaussian inverse Wishart PHD filter that tracks closely spaced extended targets. *Radioengineering* **2017**, *26*, 573–580. [[CrossRef](#)]
23. Wang, X.; Guo, R.; Jha, N.K.; Wolf, W. Multi-sensor targets data association and track fusion based on novel AWFCM. *Int. J. Nonlinear Sci. Numer. Simul.* **2009**, *10*, 811–822. [[CrossRef](#)]

24. Wang, H.; Liu, G.; Zhang, D.; Han, K.; Jiang, M. Clustering based multiple hypothesis multi-target localization. In Proceedings of the 2017 4th International Conference on Information Science and Control Engineering (ICISCE), Changsha, China, 21–23 July 2017; pp. 898–901.
25. Huang, Y.; Zhang, Y.; Li, N.; Wu, Z.; Chambers, J.A. A novel robust student's-t-based Kalman filter. *IEEE Trans. Aerosp. Electron. Syst.* **2017**, *53*, 1545–1554. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).