

Article

## Target Tracking and Classification from Labeled and Unlabeled Data in Wireless Sensor Networks

Jaehyun Yoo and Hyoun Jin Kim \*

Department of Mechanical and Aerospace Engineering, Seoul National University, 599 Gwanangno, Gwanak-gu, Seoul KS013, Korea; E-Mail: yjh5455@gmail.com

\* Author to whom correspondence should be addressed; E-Mail: hjinkim@snu.ac.kr;  
Tel.: +82-2-880-1907.

External Editor: Leonhard M. Reindl

Received: 17 October 2014; in revised form: 24 November 2014 / Accepted: 2 December 2014 /  
Published: 11 December 2014

---

**Abstract:** Tracking the locations and identities of moving targets in the surveillance area of wireless sensor networks is studied. In order to not rely on high-cost sensors that have been used in previous researches, we propose the integrated localization and classification based on semi-supervised learning, which uses both labeled and unlabeled data obtained from low-cost distributed sensor network. In our setting, labeled data are obtained by seismic and PIR sensors that contain information about the types of the targets. Unlabeled data are generated from the RF signal strength by applying Gaussian process, which represents the probability of predicted target locations. Finally, by using classified unlabeled data produced by semi-supervised learning, identities and locations of multiple targets are estimated. In addition, we consider a case when the labeled data are absent, which can happen due to fault or lack of the deployed sensor nodes and communication failure. We overcome this situation by defining artificial labeled data utilizing characteristics of support vector machine, which provides information on the importance of each training data point. Experimental results demonstrate the accuracy of the proposed tracking algorithm and its robustness to the absence of the labeled data thanks to the artificial labeled data.

**Keywords:** low-cost sensor network; multi-target tracking; semi-supervised learning; Gaussian process

---

## 1. Introduction

Wireless sensor networks (WSNs) consist of small, low-power sensor nodes that collect, process, and communicate the sensor data. WSNs are suitable to monitor environmental phenomena widely distributed in space and time. This paper focuses on the problem of localization and classification of multiple targets moving within the area of deployed sensor nodes, which plays an important role in surveillance application [1], environmental monitoring [2–4], and traffic monitoring [5].

In particular, this paper considers low-cost sensors such as acoustic, magnetic, seismic, ultrasonic sensors and radio signal strength as shown in Figure 1, which consume low power. The low-cost sensors limit the direct adaptation of existing multi-target tracking algorithm due to following reasons. First, many multi-target tracking algorithms, such as Gaussian mixture probability hypothesis density filter [6], Markov chain Monte Carlo data association [7,8], assume that sensors measure location or both range and angle of targets, for example by using RFID [9], sonar, and LiDAR [10]. Generally, low-cost sensors such as RSSI measure relative distance to the target, not location. Therefore, without employing some additional data-association algorithms, it is difficult to directly adopt those algorithms [6–10] in our setup. Second, popular conventional localization algorithms such as time of arrival (TOA), time difference of arrival (TDOA) and angle of arrival (AOA) need expensive hardware [11,12] (e.g., laser) that is not available for the low-cost sensor nodes. Third, the memory and computational power in the low-cost sensor network are limited. There are existing classification methods such as fast Fourier transform (FFT) [13,14] and feature-aided tracking [15–17] in order to extract feature points from raw data. These need heavy computational load and large memory, which are intractable for low-cost sensor nodes without additional digital signal co-processors.

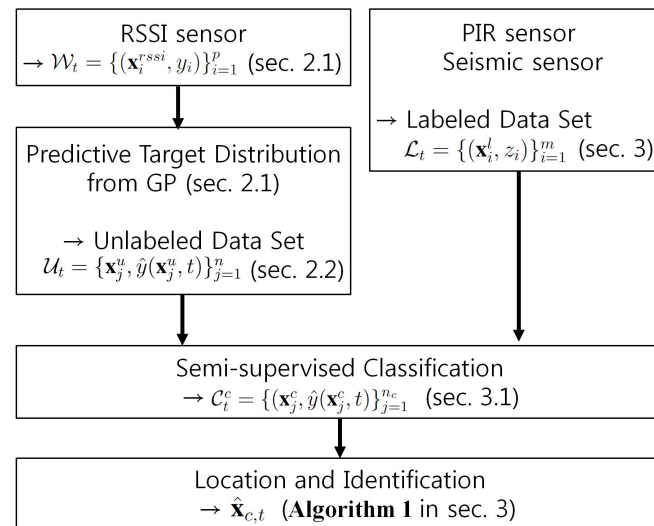
**Figure 1.** Experimental setup (clockwise): 4 m × 4 m size surveillance area, deployed sensor nodes on a tripod, ground and aerial targets.



In order to meet the requirement of the multi-target tracking using low-cost sensor networks, we propose a new tracking algorithm based on semi-supervised learning as described in Figure 2. When a few labeled data are available, the semi-supervised learning [18] can efficiently utilize large amount of unlabeled data. In our experimental scenario, labeled data are obtained by deployed passive infrared

(PIR) sensors and seismic sensors. This setup is suitable for semi-supervised learning because labeled data are sparsely obtained over space from those sensors due to their narrow sensing range relative to the wide surveillance area.

**Figure 2.** Overview of the localization and classification algorithm.



While labeled data present type information of heterogeneous targets, unlabeled data contain information about locations of targets. In the proposed framework, unlabeled data are generated from predictive target distribution that represents the probability of predicted target locations. To obtain the distribution from RSSI measurements, we employ Gaussian process (GP), which is a non-parametric machine learning technique. Due to its flexibility in modeling complex expressions using a small number of parameters, GP has been widely used to approximate signal strength [19].

Unlabeled data generated by GP contain information of target locations. Then, unlabeled data are classified via semi-supervised learning along with the small amount of labeled data, which provide information of both locations and identifications of targets. In other words, we can track targets using GP and semi-supervised learning in sequence.

We also propose a technique to deal with absence of labeled data, which can occur due to sensors fault, lack of the deployed sensors, or communication failure. We overcome this problem by propagating support vectors of each time instant, *i.e.*, essential data points during classification, to the next time step and defining those that match with the incoming unlabeled data as artificial labeled data. This allows to maintain target tracking regardless of the existence of the labeled data.

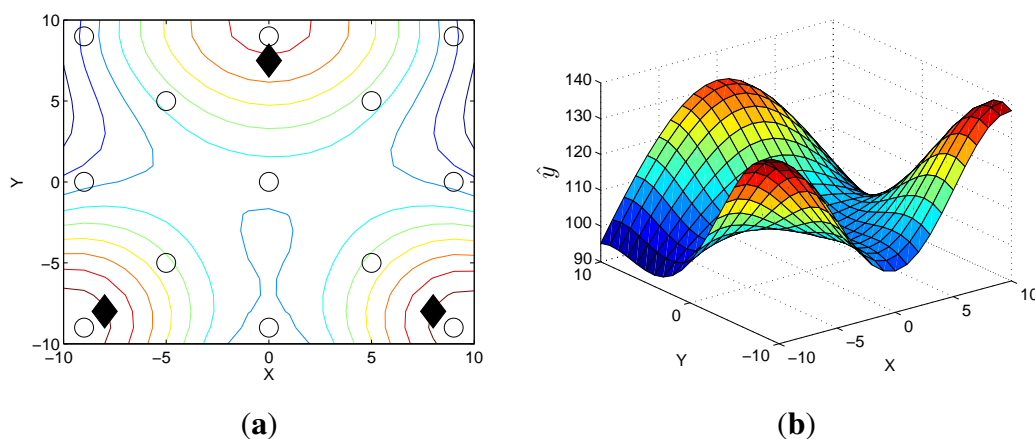
The proposed algorithm is applied to an experiment for the tracking of moving aerial and ground vehicles, as shown in Figure 1. From the experiments with an actual wireless sensor network in a  $4\text{ m} \times 4\text{ m}$  surveillance area, our algorithm achieves good localization results. Moreover, the suggested algorithm is robust to the absence of the labeled data thanks to the artificial labeled data.

The rest of this paper is organized as follows. Section 2 introduces RSSI-based predictive target distribution using a Gaussian process. In Section 3, the tracking algorithm using semi-supervised learning is described. Section 4 shows the experimental results of tracking aerial and ground targets. Finally, Section 5 is devoted to concluding remarks.

## 2. RSSI-Based Predictive Target Distribution

In received signal strength indication (RSSI)-based localization, the signal strength emitted by targets is measured by many sensors spatially distributed near the targets [20]. With the known locations of sensors, these measurements are used to make predictive target distribution that represents the probability of the predicted target locations. Figure 3 is a distribution example where 13 sensor nodes measure the RSSIs emitted from three targets.

**Figure 3.** Predictive target distribution is described where (a) deployed sensor nodes (circles) measure the RSSIs emitted from the three targets (diamonds) and these measurements are used to make the distribution; and (b) the 3-D distribution using a Gaussian process is shown. In the RSSI-based distribution, the higher value on the  $z$ -axis denotes higher probability of the predicted locations of the targets.



Section 2.1 describes how the predictive target distribution is obtained by using a Gaussian process (GP). Section 2.2 shows how unlabeled data are generated from the constructed predictive target distribution.

### 2.1. Gaussian Process

Gaussian process for regression (GP) is a non-parametric regression model that defines a Gaussian distribution over output value for input value, conditioned on training data. The advantage of GP is its flexibility in modeling complex expressions under noise using a small number of learning parameters.

Let  $y_1, \dots, y_p$  be the measurement set where  $y_i$  is the RSSI measurement obtained from the target nearest to the  $i^{th}$  node and  $p$  is the number of RSSI sensor nodes. Also, sensor locations are defined as  $\mathbf{x}_1^{rssi}, \dots, \mathbf{x}_p^{rssi}$ . Then,  $\mathcal{W} = \{(\mathbf{x}_i^{rssi}, y_i)\}_{i=1}^p$  is the training set obtained from

$$y_i = g(\mathbf{x}_i^{rssi}) + \varepsilon, \quad (1)$$

where the Gaussian noise  $\varepsilon$  is drawn from normal distribution  $\mathcal{N}(0, \sigma_{gp}^2)$ . In the GP model, the joint distribution over the noisy training outputs  $\mathbb{Y} = \{y_1, \dots, y_p\}^T$  is a function of the training inputs  $\mathbb{X} = \{\mathbf{x}_1^{rssi}, \dots, \mathbf{x}_p^{rssi}\}$ , with the form

$$\mathbb{Y} \sim \mathcal{N}(0, K(\mathbb{X}, \mathbb{X}) + \sigma_{gp}^2 I), \quad (2)$$

where  $K(\mathbb{X}, \mathbb{X})$  is  $p \times p$  kernel matrix whose  $(i, j)^{th}$  element is  $k(\mathbf{x}_i, \mathbf{x}_j)$ . The squared exponential is a commonly used kernel function, given by

$$k(\mathbf{x}_i, \mathbf{x}_j) = \theta_1 \exp \left( -\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\theta_2} \right) \quad (3)$$

where  $\theta_1$  and  $\theta_2$  are hyperparameters of the kernel.

After the training, the GP estimates the output value of the Gaussian process for the input value that is an interested location  $\mathbf{x}_* = [X, Y]^T \in \chi$ , where  $\chi$  is a user-defined set of the interested location and its cardinality is  $v = |\chi|$ . The output value has the mean  $\hat{y}(\mathbf{x}_*)$  and variance  $\hat{\sigma}^2(\mathbf{x}_*)$ :

$$g(\mathbf{x}_*) \sim \mathcal{N}(\hat{y}(\mathbf{x}_*), \hat{\sigma}^2(\mathbf{x}_*)) \quad (4)$$

where

$$\begin{aligned} \hat{y}(\mathbf{x}_*) &= K(\mathbf{x}_*, \mathbb{X})^T (K(\mathbb{X}, \mathbb{X}) + \sigma_{GP}^2 I)^{-1} \mathbb{Y} \\ \hat{\sigma}^2(\mathbf{x}_*) &= k(\mathbf{x}_*, \mathbf{x}_*) - K(\mathbf{x}_*, \mathbb{X})^T (K(\mathbb{X}, \mathbb{X}) + \sigma_{GP}^2 I)^{-1} K(\mathbf{x}_*, \mathbb{X}) \end{aligned}$$

We note that  $\hat{y}(\mathbf{x}_*, t)$  represents the GP estimation at time step  $t$ .

**Example 1.** Figure 3b is an example of the distribution obtained by the GP where  $\chi$  is defined as  $\{(X, Y) : X, Y \in \{-10, \dots, 10\}\}$  and the values on the  $z$ -axis are the estimated means. When using the RSSI measurements as the input of the GP and the corresponding locations as the output of the GP, the distribution represents the predicted sensor measurements over the space as illustrated in Figure 3b. More specifically, the estimated  $\hat{y}(\mathbf{x}_*)$  indicates the predicted RSSI measurement of an artificial sensor that does not exist at  $\mathbf{x}_*$ . Because the RSSI value is larger when the target is closer, the value on the  $z$ -axis of the distribution,  $\hat{y}$ , implies the probability with which the targets are located at  $(X, Y)$ .

## 2.2. Unlabeled Data Generation

We highlight again that larger value on the  $z$ -axis of the predictive target distribution denotes higher probability of the predicted locations of targets. In this manner, we derive a dataset  $\mathcal{U}_t = \{(\mathbf{x}_j^u, \hat{y}(\mathbf{x}_j^u, t))\}_{j=1}^n$  for  $\mathbf{x}^u \in \mathbf{x}$  from the predictive target distribution satisfying  $\hat{y}(\mathbf{x}_j^u, t) \geq y_{th}$  where  $y_{th}$  is a user-defined threshold. Thus, the dataset  $\mathcal{U}_t$  are correlated with the predictive locations of the targets at time  $t$ . Depending on the value of  $y_{th}$ , the amount of unlabeled data is decided. As described in Figure 3b, the value larger than 130 can be considered high probability for the target locations. For this reason, we set  $y_{th} = 130$  in this paper.

We assume that sensor nodes do not know the labels of the RSSI measurements, *i.e.*, the nodes do not know who emitted the RF signal. With this assumption, the dataset  $\mathcal{U}_t$  does not provide any information about identity of the targets. In the following section, we introduce a semi-supervised algorithm to classify the unlabeled data in order to identify the targets.

## 3. Target Tracking Based on Semi-Supervised Learning

Traditional classification algorithms such as supervised learning are inaccurate when only a few labeled data are available. To overcome this problem, semi-supervised classification [21,22] improves the efficiency and performance of supervised learning by using both labeled and unlabeled data.

We assume that  $N$  different targets are identified by the identification label set  $\{1, \dots, N\}$ . There is a small set of labeled data, given by  $\mathcal{L}_t = \{(\mathbf{x}_i^l, z_i(\mathbf{x}_i^l, t))\}_{i=1}^m$ , where  $\mathbf{x}_i^l$  is the location of the  $i^{th}$  labeled data and  $z_i(\mathbf{x}_i^l, t)$  is the observation denoting the identification label of the targets.

We describe the Laplacian support vector machine (LapSVM) algorithm in Section 3.1. The best scenario for semi-supervised classification is where there always exists at least one labeled data for each class while tracking the target. However, this cannot be guaranteed due to sensor fault, lack of deployed sensor, or communication failure. Considering a case where there is no labeled data at some time instants, we propose incremental learning for target tracking in Section 3.2.

### 3.1. Laplacian Support Vector Machine (LapSVM)

In the graph-based semi-supervised classification algorithm, the geometry of data is represented by an undirected graph  $\mathcal{G} = (V, E)$  with nodes  $V = \{1, \dots, n + m\}$  and edges  $E$ , where  $n$  and  $m$  are the number of unlabeled and labeled data, respectively. The nodes  $V$  represent the training data, and the edges  $E$  denote the similarities between the nodes. These similarities are given by the connection strength from a node  $i$  to another node  $j$ , which is encoded in the element  $W_{ij}$  of a weight matrix  $W$ . A typical weight matrix is given by the a Gaussian function of the Euclidean distance between data, with length scale  $\theta_3$ :

$$W_{ij} = \begin{cases} \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\theta_3}\right) & \text{if } (i, j) \in E \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where training dataset  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{n+m}\} = \{\mathbf{x}_1^u, \dots, \mathbf{x}_n^u, \mathbf{x}_1^l, \dots, \mathbf{x}_m^l\}$  consists of unlabeled and labeled data points. An edge  $(i, j)$  between the nodes  $\mathbf{x}_i$  and  $\mathbf{x}_j$  is linked by the  $k$  nearest neighbors. To assign the weights to the edge set of  $\mathcal{G}$ , the graph Laplacian matrix is defined as

$$L = D - W \quad (6)$$

where  $D$  is a diagonal matrix with entries  $D_{ii} = \sum_{j=1}^{n+m} W_{ij}$ .

The semi-supervised learning algorithm outputs an  $(n + m)$ -dimensional real-valued vector  $\mathbf{f}^* = [f(\mathbf{x}_1), f(\mathbf{x}_2), \dots, f(\mathbf{x}_{n+m})]^T$ , where  $f(\mathbf{x}_j) = \sum_{i=1}^{n+m} \alpha_i^* k(\mathbf{x}_i, \mathbf{x}_j)$ . In the LapSVM algorithm, optimal  $\alpha^* = [\alpha_1, \dots, \alpha_{n+m}]^T$  is given by

$$\alpha^* = (2\gamma_A I + 2\frac{\gamma_I}{(n+m)^2} LK)^{-1} J^T Z^{(l)} \beta^* \quad (7)$$

where  $Z^{(l)}$  is an  $m$ -by- $m$  diagonal matrix given by  $Z_{ii}^{(l)} = z_i$  with  $z_i \in \{-1, +1\}$  and  $J$  is an  $m$ -by- $(m + n)$  matrix given by  $[I_{m,m} \ O_{m,n}]$  where  $I_{m,m}$  is an  $m$ -by- $m$  identity matrix and  $O_{m,n}$  is an  $m$ -by- $n$  rectangular matrix with all zeros. Tuning variables  $\gamma_A$  and  $\gamma_I$  are weight parameters, and  $K$  is the kernel matrix defined in Equation (3).

The optimal  $\beta^*$  in order to get the optimal  $\alpha^*$  in Equation (7) is obtained by the following quadratic program:

$$\beta^* = \max_{\beta \in \mathcal{R}^m} \sum_{i=1}^m \beta_i - \frac{1}{2} \beta^T Q \beta \quad (8)$$



subject to:

$$\sum_{i=1}^m z_i \beta_i = 0 \quad (9)$$

$$0 \leq \beta_i \leq \frac{1}{m} \text{ for } i = 1, \dots, m, \quad (10)$$

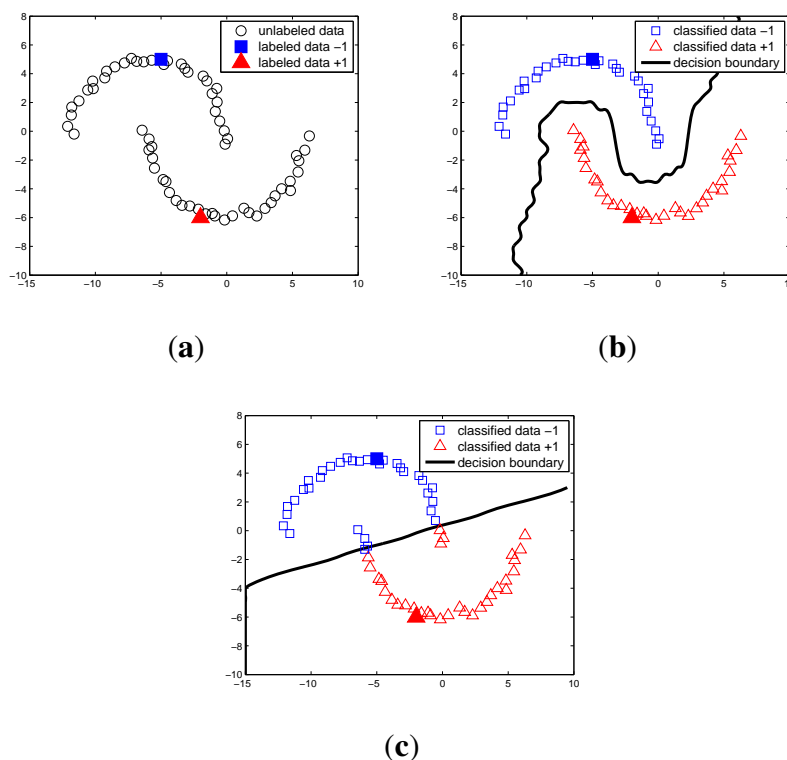
where

$$Q = Z^{(l)} J K \left( 2\gamma_A I + 2 \frac{\gamma_I}{(n+m)^2} L K \right)^{-1} J^T Z^{(l)} \quad (11)$$

In the semi-supervised learning, the unlabeled data help finding the decision boundary more accurately, assuming that data points of each class tend to form a cluster (called cluster assumption [21]). In other words, if points are in the same cluster, they are likely to belong to the same class.

**Example 2.** Figure 4 illustrates the two-moon classification problem. In order to show the usefulness of unlabeled data, we prepare only one labeled data for each class. In the semi-supervised algorithm, the adjacent points in a high-density data region are classified into the same class. This gives a correct decision boundary, in contrast to the supervised learning shown in Figure 4c. The influence of the unlabeled data in this algorithm is handled by  $\gamma_I$  in Equation (7). When  $\gamma_I = 0$ , Equation (7) gives zero coefficients over the unlabeled data, so the algorithm becomes standard supervised SVM. On the other hand, as  $\gamma_I$  increases, the effect of unlabeled data also increases. In this paper, we set  $\gamma_I = 10^3$  and  $\gamma_A = 0.1$ .

**Figure 4.** Two moon data classification where (a) 64 unlabeled and 2 labeled data before classification; (b) result of the LapSVM semi-supervised learning; (c) result of the standard SVM.



General classification algorithm presets the label value to +1 or −1 and appoints the label of an unlabeled data  $\mathbf{x}^u$  based on the function  $f$ : if  $f(\mathbf{x}^u) > 0$  then the label of  $\mathbf{x}^u$  is +1, or if  $f(\mathbf{x}^u) \leq 0$  then the label of  $\mathbf{x}^u$  is −1. For the multi-label classification problem, it can be solved using “One Against One” or “One Against All” algorithms [23]. Specific descriptions of these algorithms are out of the scope of this paper. Instead, we define  $Q(\mathbf{x})$  as a function mapping the data to the label. Therefore, all data points  $\mathbf{x}_i^u$ ,  $1 \leq i \leq n$  will have labels through the classifier, given by  $Q(\mathbf{x}_i^u) \in \{1, \dots, N\}$ , where  $N$  is the number of identification labels. For example, we consider two types of targets to track, thus  $N = 2$ .

As the result of classification, we define  $\mathcal{C}_t^c$  as classified sets to label  $c$ , given by  $\mathcal{C}_t^c = \{(\mathbf{x}_j^c, \hat{y}(\mathbf{x}_j^c, t) \mid Q(\mathbf{x}_j^c) = c\}_{j=1}^{n_c}$  for  $c \in \{1, \dots, N\}$ , where  $n_c$  is the number of data points that are labeled to  $c$ . The total set and number of unlabeled data,  $\mathcal{U}_t$  and  $n$  described in Section 2.2, can be described by  $n = \sum_{c=1}^N n_c$  and  $\mathcal{U}_t = \bigcup_{c=1}^N \mathcal{C}_t^c$ .

### 3.2. Incremental Time Series Classification

We consider a situation where there is no labeled data after the targets intrude the surveillance field. Most papers assume that initial positions of the intruders are successfully estimated by deploying sensor nodes intensively at edge of the surveillance field. Accordingly, we also assume that the set of labeled data exists at the moment the targets just enter the surveillance area. In this section, we focus on how to perform semi-supervised classification when labeled data is absent after the target intruded.

Target information at the current time step is correlated with the information at the previous time step such as a Markov process model. This leads us to exploit the classified data at the previous time step to create artificial labeled data for the current time step (artificial labeled data is different with the genuine labeled data made by the sensor). Our strategy is to select useful data among the classified datasets at the previous time step and define them as the artificial labeled data for the current time step.

The support vector machine framework offers a natural method to select such useful data. The classification rule is based on the decision boundary that depends on the support vector,  $x_i$  with  $|\alpha_i| > 0$  in Equation (7), in the SVM algorithm. In other words, the support vectors are representative points of the classification result. Moreover, once the data is well trained, most of the  $\alpha$ 's become zero and only a few support vectors have nonzero value of the coefficient  $|\alpha|$ . Therefore, the small set of support vectors is suitable to be propagated to the next step and they become candidates for artificial labeled data.

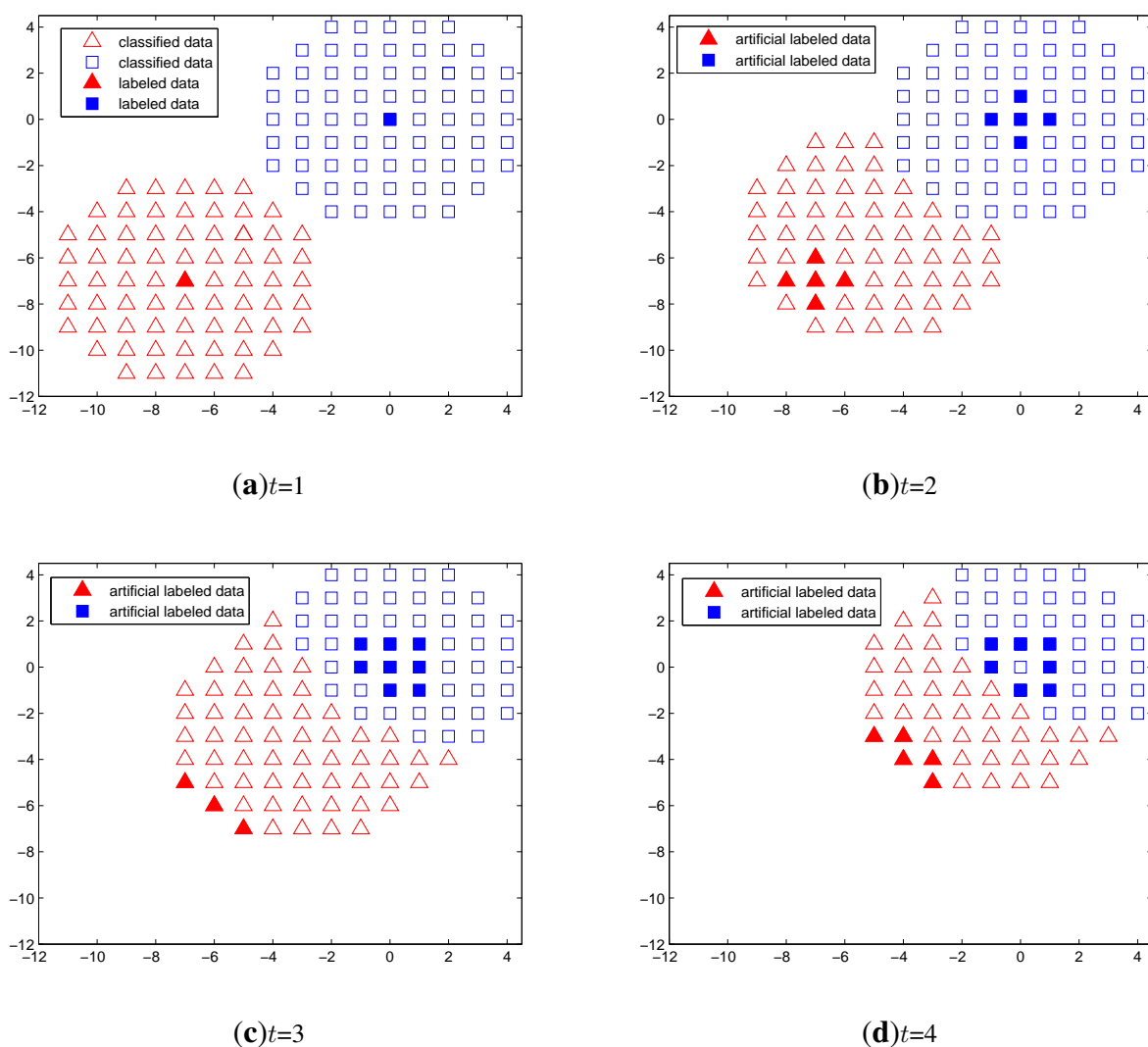
One more consideration in constructing the artificial labeled data is that they have to be updated as time elapses. Some candidates obtained during the previous time step may not correlate with the current data. In order to find correlated points among the candidates to the current data, we check whether the candidates belong to the current unlabeled dataset or not. The candidates satisfying this condition become the artificial labeled data at time  $t$ , given by  $\mathcal{A}_t^c = \{(\mathbf{x}_i^c, z_i(\mathbf{x}_i^c, t-1)) \mid |\alpha_i| > 0, \mathbf{x}_i^c \in \mathcal{C}_t^c \cap \mathcal{C}_{t-1}^c\}_{i=1}^{s_c}$ , where  $s_c$  is the number of the artificial labeled data that are labeled to  $c$ .

**Example 3.** Figure 5 shows a time series classification example when no labeled data exist except in the first time step. In this simulation, we let one target remain at the origin all the time and the other approaches the origin. The unlabeled data are generated over a circular region around the two targets. The expected incorrect outcome is that one classified set is merged into the other set, which



wrongly represents multiple overlapping targets as one identity. However, from the simulation results, the classification using only the artificial labeled data performs well both when the targets are approaching and when the targets are apart from each other.

**Figure 5.** Time series classification when one target approaches the other target at the origin. Unlabeled data is generated over a circular region around the two targets, as marked by blank triangle and square symbols. Among those data, classified points are denoted as filled triangle and square. At the time instant shown in (a), it is assumed that labeled data exists for each target. After that, labeled data is not available from (b) to (d), so semi-supervised classification is performed using only the artificial labeled data.



Concluding this section, we summarize a procedure for multi-target localization and classification into Algorithm 1.

**Algorithm 1** Multi-target localization and classification.

**Step 1:** Collect  $p$  RSSI measurement dataset  $\mathcal{W}_t = \{(\mathbf{x}_i^{RSSI}, y_i)\}_{i=1}^p$

**Step 2:** Collect  $m$  labeled dataset  $\mathcal{L}_t = \{(\mathbf{x}_i^l, z_i)\}_{i=1}^m$

**Step 3:** Generate the predictive target distribution (Section 2.1)

**Step 4:** Obtain unlabeled dataset from the predictive target distribution

$\mathcal{U}_t = \{\mathbf{x}_j^u, \hat{y}(\mathbf{x}_j^u, t)\}_{j=1}^n$ . (Section 2.2)

**Step 5:** If  $t \geq 2$ , generate artificial labeled data  $\mathcal{A}_t^c$ , then renew  $\mathcal{L}_t = \mathcal{L}_t \cup \mathcal{A}_t^c$ . (Section 3.2)

**Step 6:** Execute the LapSVM classification algorithm, then obtain the classified dataset

$\mathcal{C}_t^c = \{(\mathbf{x}_j^c, \hat{y}(\mathbf{x}_j^c, t))\}_{j=1}^{n_c}$  for  $c \in \{1, \dots, N\}$ . (Section 3.1)

**Step 7:** Estimate the positions of the targets  $\hat{\mathbf{x}}_{c,t}$  for  $c = \{1, \dots, N\}$ :

$$\hat{\mathbf{x}}_{c,t} = \mathbf{x}_k^c,$$

$$k = \operatorname{argmax}_j \hat{y}(\mathbf{x}_j^c, t),$$

$$\text{where } \hat{y}(\mathbf{x}_j^c, t) \in \mathcal{C}_t^c \text{ for } j \in \{1, \dots, n_c\}.$$

**Step 8:**  $t \leftarrow t + 1$ , then go to Step1.

## 4. Experiment

This section describes experimental setup, characteristics of sensor data, (*i.e.*, PIR, seismic, RSSI sensors), and tracking results.

### 4.1. Setup

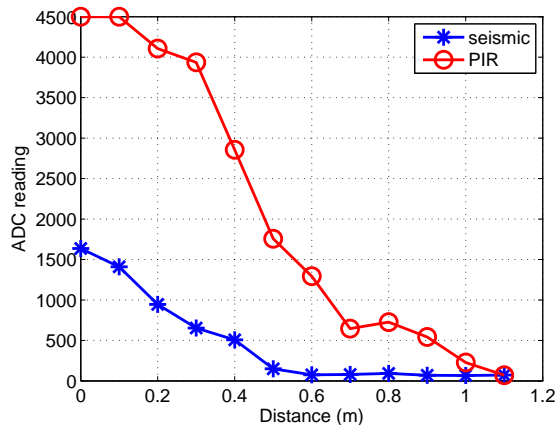
We consider a two-dimensional tracking problem for one ground and one aerial target using a total of 26 sensor nodes. To identify the aerial vehicle, 13 spatially distributed seismic sensors are used to recognize the airflow due to aerial vehicle. The ground vehicle is identified by 13 passive infrared sensors (PIR). In addition, RSSI is available for both types of sensor nodes. The true trajectories of the targets are obtained by a Vicon Motion System that tracks reflective markers attached to each target.

### 4.2. Characteristics of Sensor Data

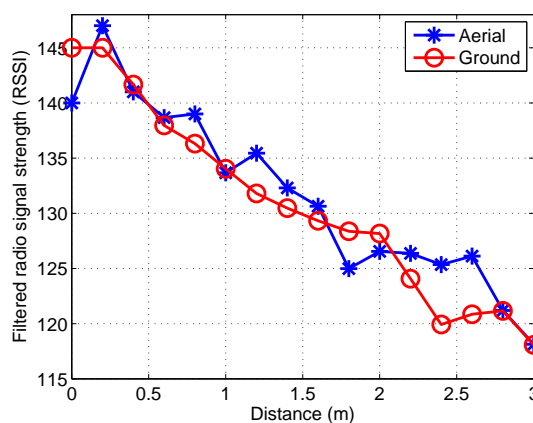
We record the raw ADC readings of one PIR sensor by varying the distance between the node and the ground target. Similarly, the raw ADC of one seismic sensor is measured using the aerial target, shown in Figure 6. We set threshold values to each sensor, *i.e.*, 300 for the seismic sensor and 500 for the PIR sensor. Based on the defined thresholds on readings, the  $i^{th}$  seismic sensor node yields +1 labeled data when a reading is greater than the threshold, *i.e.*,  $z_i = +1$ . Similarly, the  $j^{th}$  PIR sensor node generates −1 labeled data, *i.e.*,  $z_j = -1$ .

Besides, all 26 sensor nodes measure the RSSI. Considering the noisy ADC reading of the RSSI, moving average filter [24] is adopted. Figure 7 shows the relationship between the filtered RSSI measurement and the distance between a node and the ground target and the aerial target, respectively.

**Figure 6.** Raw ADC readings of PIR sensor with respect to the distance between the PIR node and ground target and seismic sensor with respect to the distance between the seismic node and aerial target.



**Figure 7.** RSSI measurements adopting moving average filter with respect to the distance between a node and the ground target and the distance between a node and the aerial target.



#### 4.3. Tracking Results

In our experiments, two targets start from different corners of the surveillance area, crossing at the middle of their pathway. We perform five trials, whose results are summarized in Figure 8. Because the RSSI characteristics are affected by the motion of the flying aerial robot, its tracking results are worse than results of the ground robot.

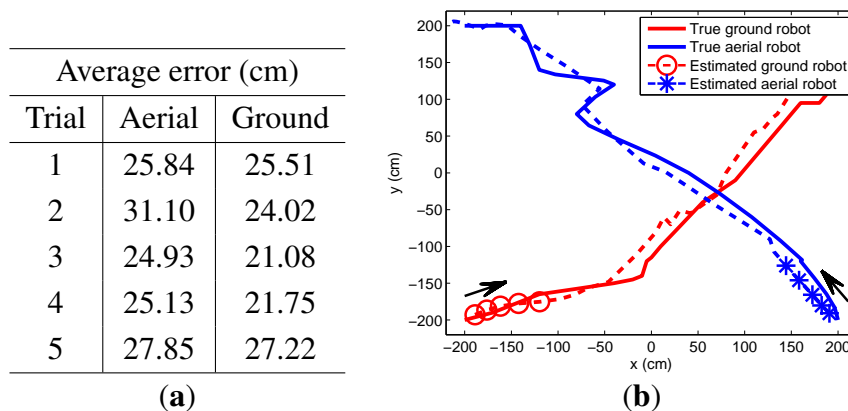
Moreover, we analyze the effect of the artificial labeled data introduced in Section 3.2 with respect to the ratio of existence of the labeled data,  $R$ :

$$R = \frac{\sum_i L(t_i)}{T} \times 100 \quad (12)$$

$$L(t_i) = \begin{cases} 1, & \text{if the labeled data is present at } t_i \\ 0, & \text{if the labeled data is absent at } t_i, \end{cases} \quad (13)$$

where  $t_i$  is discrete time step and  $T$  is total tracking time.

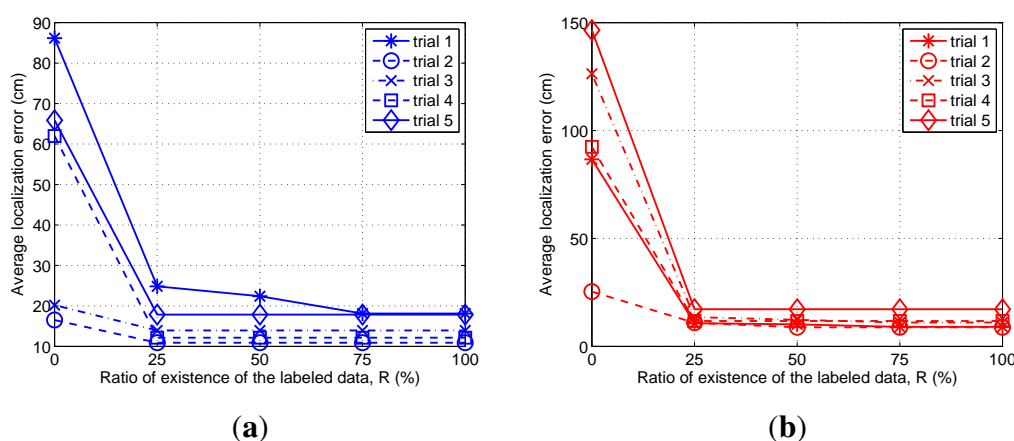
**Figure 8.** Five experimental tracking results for moving ground and aerial targets with (a) average error and (b) true and estimated trajectories of the first trial experiment.



We intentionally remove the labeled data among the data obtained via the experiment up to each 25%, 50%, 75% 100%, and then perform tracking. The results of the average localization error for each aerial and ground targets are summarized in Figure 9.

When there is no labeled data all the time ( $R = 0\%$ ), four of the five results do not indicate the correct trajectories. From 25% to 100% of the existence ratio of the labeled data, the results are almost same as the results for 100% absence ratio. We confirm that the artificial labeled data prevent the tracking performance from getting deteriorated by the lack of the labeled data.

**Figure 9.** Average localization error from five experiments according to the existence ratio of the labeled data. (a) Localization error of the aerial target; (b) Localization error of the ground target.



## 5. Conclusions

This paper proposed a new multi-target localization and classification for low-cost sensor networks. Data configuration that consists of the unlabeled and labeled data obtained from low-cost sensor nodes was well-suited to apply semi-supervised learning. Furthermore, the incremental algorithm was developed with semi-supervised learning to handle the situation where labeled data are absent. The

suggested algorithm was evaluated for the tracking of aerial and ground targets and showed accurate tracking performance and robustness to the absence of labeled data.

## Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Ministry of Science, ICT & Future Planning (MSIP), and the Korea government (MEST).

## Author Contributions

Jaehyun Yoo and Hyoun Jin Kim designed the tracking algorithm; Jaehyun Yoo performed the experiments and analyzed the data; Jaehyun Yoo and Hyoun Jin Kim wrote the paper.

## Conflicts of Interest

The authors declare no conflict of interest.

## References

1. Chen, M.; González, S.; Cao, H.; Zhang, Y.; Vuong, S.T. Enabling low bit-rate and reliable video surveillance over practical wireless sensor network. *J. Supercomput.* **2013**, *65*, 287–300.
2. Song, G.; Zhou, Y.; Ding, F.; Song, A. A mobile sensor network system for monitoring of unfriendly environments. *Sensors* **2008**, *8*, 7259–7274.
3. Wan, J.; Yu, Y.; Wu, Y.; Feng, R.; Yu, N. Hierarchical leak detection and localization method in natural gas pipeline monitoring sensor networks. *Sensors* **2012**, *12*, 189–214.
4. Lloret, J.; Bosch, I.; Sendra, S.; Serrano, A. A wireless sensor network for vineyard monitoring that uses image processing. *Sensors* **2011**, *11*, 6165–6196.
5. Torfs, T.; Sterken, T.; Brebels, S.; Santana, J.; van den Hoven, R.; Spiering, V.; Bertsch, N.; Trapani, D.; Zonta, D. Low power wireless sensor network for building monitoring. *IEEE Sens. J.* **2013**, *13*, 909–915.
6. Vo, B.N.; Ma, W.K. The Gaussian mixture probability hypothesis density filter. *IEEE Trans. Signal Process.* **2006**, *54*, 4091–4104.
7. Oh, S.; Russell, S.; Sastry, S. Markov chain Monte Carlo data association for multi-target tracking. *IEEE Trans. Autom. Control* **2009**, *54*, 481–497.
8. Liu, J.; Chu, M.; Reich, J.E. Multitarget tracking in distributed sensor networks. *IEEE Signal Process. Mag.* **2007**, *24*, 36–46.
9. Shin, J.; Lee, N.; Thrun, S.; Guibas, L. Lazy inference on object identities in wireless sensor networks. In Proceedings of the 4th International Symposium on Information Processing in Sensor Networks, Los Angeles, CA, USA, 25–27 April 2005; p. 23.
10. Nelson, J.K.; Roufarshbaf, H. A tree search approach to target tracking in clutter. In Proceedings of the 12th International Conference on Information Fusion (FUSION'09), Seattle, WA, USA, 6–9 July 2009; pp. 834–841.

11. Bachrach, J.; Taylor, C. Localization in sensor networks. In *Handbook of Sensor Networks: Algorithms and Architectures*; Wiley: Hoboken, NJ, USA, 2005; Volume 1.
12. Vercauteren, T.; Guo, D.; Wang, X. Joint multiple target tracking and classification in collaborative sensor networks. *IEEE J. Sel. Areas Commun.* **2005**, *23*, 714–723.
13. Bilik, I.; Tabrikian, J.; Cohen, A. GMM-based target classification for ground surveillance Doppler radar. *IEEE Trans. Aerosp. Electron. Syst.* **2006**, *42*, 267–278.
14. Duarte, M.F.; Hu, Y.H. Vehicle classification in distributed sensor networks. *J. Parallel Distrib. Comput.* **2004**, *64*, 826–838.
15. Roufarshbaf, H.; Nelson, J.K. Feature-aided initiation and tracking via tree search. In Proceedings of the 2013 Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 3–6 November 2013; pp. 2138–2142.
16. Hong, L.; Cui, N.; Pronobis, M.; Scott, S. Local motion feature aided ground moving target tracking with GMTI and HRR measurements. *IEEE Trans. Autom. Control* **2005**, *50*, 127–133.
17. Brown, A.P.; Sullivan, K.J.; Miller, D.J. Feature-Aided Multiple Target Tracking in the Image Plane. *Proc. SPIE* **2006**, doi:10.1117/12.666143.
18. Chapelle, O.; Schölkopf, B.; Zien, A. *Semi-Supervised Learning*; MIT Press: Cambridge, MA, USA, 2006; Volume 2.
19. Yoo, J.H.; Kim, W.; Kim, H.J. Event-driven Gaussian process for object localization in wireless sensor networks. In Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Francisco, CA, USA, 25–30 September 2011; pp. 2790–2795.
20. Sugano, M.; Kawazoe, T.; Ohta, Y.; Murata, M. Indoor localization system using RSSI measurement of wireless sensor network based on ZigBee standard. *Target* **2006**, *538*, 050.
21. Belkin, M.; Niyogi, P.; Sindhvani, V. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *J. Mach. Learn. Res.* **2006**, *7*, 2399–2434.
22. Guillaumin, M.; Verbeek, J.; Schmid, C. Multimodal semi-supervised learning for image classification. In Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 13–18 June 2010; pp. 902–909.
23. Tsujinishi, D.; Koshiba, Y.; Abe, S. Why pairwise is better than one-against-all or all-at-once. In Proceedings of the 2004 IEEE International Joint Conference on Neural Networks, Budapest, Hungary, 25–29 July 2004; Volume 1.
24. Smith, S.W. *The Scientist and Engineer's Guide to Digital Signal Processing*; California Technical Pub: San Diego, CA, USA, 1997.