IntenCT: Efficient Multi-Target Counting and Tracking By Binary Proximity Sensors

Yongcai Wang
Renmin University
Beijing China

Lei Song
IIIS, Tsinghua University
Beijing, China

Zhaoquan Gu
The University of Hong Kong
Hong Kong, China

Deying Li
Renmin University
Beijing, China

***Abstract*—Binary proximity sensors (BPS) is a generic model for many non-collaborative, presence detecting sensor. It outputs "1" when one or more targets are presenting in its sensing range and "0" otherwise. It cannot tell the number of targets nor the targets' identities in its sensing range. But for its privacy protection and device-free properties, BPS-based tracking has attracted great attentions. However, multiple target counting and tracking (MTCT) by BPS network remains very challenging. Existing approaches generally rely on trajectory decomposition, which suffer association complexity issue and can hardly provide accurate results.**

To address these challenges, this paper presents an novel intensity-based counting and tracking approach, called *IntenCT*, which tracks the evolvement of the multi-targets' probabilistic density distribution overtime, without the complexity of enumerating the multiple targets' trajectories. Then, clustering algorithms on the density distribution are proposed to find the target groups, and count the targets in each group by calculating the integral of the density distribution in the group region. At last, the trajectories of the separable targets in each group are estimated using K-means and a motion consistency model.

Extensive analysis and simulations show that IntenCT has quadratic complexity which is very efficient; provides the current best known multi-target counting lower bound; and tracks the multi-targets more accurately than the existing approaches.

I. Introduction

Binary proximity sensor (BPS) is an abstract model for a large category of sensors including infrared[2], microwave, magnetic, pressure sensors, etc. It outputs "1" when one or more targets are within its sensing range and "0" otherwise. It cannot distinguish or count targets, nor provides moving direction or location information. But it can be widely seen in buildings, such as the infrared triggers for automatic lights.

Tracking targets by BPS have attracted great research attentions [8][14][16], because the great advantage of *privacy protection and user-device-free*, which are highly desired properties in practical tracking applications such as location-based services. The seminal works [8] and [14] presented solutions to track one target by a network of BPS.

But significant difficulties will be encountered in counting and tracking multiple targets due to the sensing ambiguity of the BPS. Existing works generally relied on trajectory decomposition using the motion consistency of users (the minimum variance of speed), to assign the most likely trajectory to each estimated target, by Hidden Markov Model [5], cluster based particle filter[15], and multiple pairs shortest path [18] algorithms. But since the target number, target identity, and the target trajectories are unknown, the trajectory association incurs high combinational complexity, which makes the existing methods inefficient, and unsatisfied in accuracy.

This paper investigates an *intensity*-based approach to address the complexity problem, where intensity is defined as the first moment (mean or expectation) of the multi-target posterior density distribution in the area of interest (AOI). It tracks the expected target distribution overtime without trajectory association and voting. In particular, the targets' dynamic models can be approximated by a linear Gaussian model, so that the intensity of each target can be modeled by a Gaussian distribution. Then the multi-target tracking problem can be converted to tracking the evolvement of intensity distribution, i.e., the evolvement of Gaussian Mixtures (GM).

By knowing the distributionof GM, a density-based clustering algorithm is then proposed to cluster the GMs. Each cluster identifies a disjoint group of targets. Benefited by the properties of the probability distribution, the expected number of targets in each group can be estimated efficiently by calculating the summation of the weights of Gaussian components in the group. After estimating the number of targets in each group, an K-means based algorithm was proposed to further partition the group into K sub-clusters, where K equals to the estimated target number. The centroid of each sub-cluster is therefore the location of a target. At last, the trajectory of each target can be referred using motion consistency information.

We show that the proposed IntenCT framework has quadratic complexity to the number of targets, which is very efficient. Extensive simulations show that, IntenCT provides the currently best known target counting lower bound, which is tightly close to the target number ground truth. It also provides good multiple target tracking performances. The overall solution shows the feasibility to use low cost, privacy protected binary sensors to provide locating services to multiple, device-free, anonymous targets, while achieving good accuracy and good efficiency.

The rest of this paper is organized as follows. Problem model and backgrounds were introduced in Section 2. The IntenCT framework was introduced in Section 3. Augmentation methods were presented in Section 4. Performance evaluations were conducted in Section 5. The paper was concluded with remarks in Section 6.

978-1-5090-1732-4/16/$31.00 ©2016 IEEE

II. BACKGROUND AND PROBLEM MODEL

A. Preliminaries

Let's consider to track unknown number of targets in an 2-D area of interest (AOI) by a network of n BPS sensors. Each sensor has a sensing radius R. A sensor outputs "1" when one or multiple targets are within its sensing region and outputs "0" otherwise. We assume the AOI is fully covered by the sensing regions of sensors. Sensor locations are assumed known by offline calibration [12], and sensors are assumed timely synchronized, which can be carried out by network synchronization protocols[6].

The states of all the sensors at a time instance, is called a *snapshot*, which is denoted by $S_k = \{s_{1,k}, \cdots, s_{n,k}\}$. A sensor's state $s_{i,k}$ is "0" or "1", where k is a time index and i indicates the ith sensor. The multi-target counting and tracking (MCMT) problem is therefore to infer the number of the targets and their motion trajectories from period 1 to K based on the given sensor snapshots $\{S_1, \cdots, S_K\}$. We assume perfect media access control and routing protocols to support the data communication and allow targets entering or leaving the AOI during this period, which will be discussed in Section IV.

B. Related Work

Tracking by BPS network has attracted great attentions.

1. *For one target case*, Kim. et al. [8] presented the first path estimation approach by a geometric method. Shrivastava et al. [14] presented to use the index of geometric patches to represent the targets' locations. They showed that the locating error for tracking one target could be up to $\Omega(\frac{1}{\rho R^{d-1}})$, where ρ is the sensor density; R is the sensing radius and d is the space dimension. The problem becomes difficult in the case of tracking multiple targets. When the target number is known, Wang et al.[18] reduced the problem to a trajectory disaggregation problem and used multiple pairs shortest path algorithm to search the most possible trajectory by exploiting the walking speed variance to disaggregate the target trajectories.

In case the target number is unknown, the MCMT problem becomes much more difficult. Estimating the target number lower bound becomes the main approximation method. Singh et al.[15] presented the target number lower bound in one diminutional network. It showed that if the "ON" sensors could be partitioned into at most X positively independent sets, the number of targets is not less than the cardinality of X. Song et al. [16] presented a target number lower bound in 2-D space. They firstly find the isolated "ON" sensor island separated by the "OFF" sensors; then showed that the lower bound of the target number in each island equals to the minimum number of cliques partitioning the UDG formed by the "ON" sensors in the island. Li et al. [9] improved the lower bound given by Song et al.[16] by temporal dependency between the snapshots and proposed an upper bound of the target number by assuming the minimum separation distance among targets.

For target tracking, Singh. et al. [15] proposed clustered particle filter to divide the particles into n_c clusters, so that the particles in each cluster track the location of one target respectively. FindingHuMo [5] proposed adaptive Hidden Markov Model (HMM) to disaggregate the motion trajectories of the multiple targets. Song et al. [16] proposed multiple color particle filter to jointly optimize the target number and their trajectories. Cao et al.[3][4] investigated the problem of tracking a group of target, which is composed by a set of targets moving correlatively (e.g. moving together). They proposed to detect the convex hull of targets and showed the worst case tracking accuracy. They proposed centroid-based algorithm[3] and circle covering based algorithm[4] to locate the convex hull of the target group.

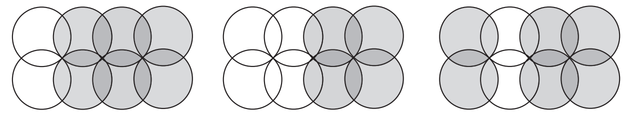

Fig. 1. A simple example to illustrate the computation complexity of the problem

C. Existing challenges.

Above approaches showed the difficulty of tracking unknown number of targets by BPS network. Existing approach generally adopted a ***counting*-then-*tracking*** procedure, which suffers the association complexity. Let's look at a simple example as shown in Fig.1.

The example shows three consecutive snapshots captured by eight BPS sensors. At $t = 1$, the two sensors in the first column is triggered "ON". The estimated lower bound of the target number is 1 and there is no upper bound. Let's consider there are m targets. They may either form one, two or three groups, distributed in the three "On" patches to trigger the sensor states. Even if we set the target number upper bound to 5, we can calculate that there are more than 150 cases of target distributions satisfying the sensor states. The snapshot at $t = 3$ provides the same number of location combinations. The snapshot at $t = 2$ is more complex, which provides more combinations. So the total possible combinations of trajectories are $150^3 (> 3 \times 10^6)$ in this simple example. We can see the main challenges lie on:

1) The uncertainty of the target number, which leads to uncertain trajectory combinations.

2) The trajectory numbers increase exponentially with time, which makes the trajectory disaggregation inefficient.

3) Associating targets to groups and trajectories is difficult, because the sensors cannot distinguish targets.

4) Lack of method to deal new target entering/leaving, which are general scenarios in practice.

D. Intensity Propogation Model

To address above challenges, an intensity propagation based approach was investigated. The concept of "Intensity" was firstly proposed by Mahler in [11], in which, intensity was defined as the first moment of multi-target posterior density distribution, which is analogous to the first-moment approach of Kalman filter in the single-target case.

In multiple target case, let's denote the location space of targets by a finite set $\mathcal{X}$. Let M_k be the number of targets at time k. The locations of these targets are denoted by $X_k = \{\mathbf{x}_{i,k}, i \in \{1, M_k\}\}$ on $\mathcal{X}$. Note that the observations of binary sensors at time k form a snapshot S_k. The optimal multi-target Bayes filter [17] updates the multi-target posterior density based on $S_{1:k}$ (which means $\{S_1, \cdots, S_k\}$), with the following recursion:

$$\begin{aligned} & p_{k|k-1}(X_k|S_{1:k-1}) \\ & = \int f_{k|k-1}\left(X_k|X\right) p_{k-1}\left(X|S_{1:k-1}\right) \mu_s\left(dX\right) \end{aligned} \tag{1}$$

where $f_{k|k-1}(X_k|X_{k-1})$ is the dynamic model of state transition; $\mu_s(dX)$ is a measurement of area in 2-D space given dX.

But equation (1) calculates integration on high dimension space X_k, which is computationally intractable. *Intensity* is to provide a computationally tractable sub-optimal alternative to the Bayesian recursion.

The first-order moment of $p_k(X_k)$ is a nonnegative function $v(x)$, where x is a point $\in \mathcal{X}$, such that for each region $\mathcal{S} \subseteq \mathcal{X}$, the integral of $v(x)$ over $\mathcal{S}$ gives the expected number of targets in $\mathcal{S}$ [11][17], which is denoted by $M_{\mathcal{S}}$:

$$M_{\mathcal{S}} = \int_{\mathcal{S}} v(x)dx \tag{2}$$

The local maxima of the intensity $v(x)$ indicate the points with the highest local concentration of the expected number of targets. The region around local maxima indicates high probability of target presence in the region. Integral over the region will provide information about the target number in the group. Further, the target locations in the group can be inferred by further processing the intensity distribution.

When the targets are moving, the intensity of multiple targets change overtime. The binary readings of sensors can be thought as observations to the multi-target intensity.

Let's denote the multi-target intensity at time k by $v_k(x)$; the intensity at $k-1$ by $v_{k-1}(x)$. The probability that a target is at ξ at time $k-1$, but moves to x at time k is denoted by $\phi_{k|k-1}\left(x|\xi\right)$. Suppose $\gamma_k\left(x\right)$ is the probability of new target appears at x at time k (e.g. when x is a doorway), then the intensity propagates from $k-1$ to k by the following recursion [11]:

$$v_{k|k-1}\left(x\right) = \int \phi_{k|k-1}\left(x|\xi\right) v_{k-1}\left(\xi\right) d\xi + \gamma_k\left(x\right) \tag{3}$$

Let s be a sensor's binary reading at time k. $g(s|x)$ is the likelihood function of a target is at x and the observation is s. Then by Bayesian recursion:

$$v_k\left(x\right) = \sum_{s \in S_k} \frac{g\left(s|x\right) v_{k|k-1}\left(x\right)}{\int g\left(s|\xi\right) v_{k|k-1}\left(\xi\right) d\xi} \tag{4}$$

The summation is for each sensor, and the posterior intensity is a function on the single target space. This makes the recursion in (3)(4) much efficient than that in (1). Note that in this model, it is assumed that each target evolves and generates observations independently.

III. IntenCT

The idea of intensity propagation is investigated to address the difficulty of tracking multiple targets in BPS networks, which leads to IntenCT. IntenCT follows a **tracking-then-counting** procedure. The tracking step propagates the intensity distribution; the counting step counts the number of targets in each group and further locates the targets. We introduce its framework in this section.

A. GM Model

1) Dynamic model of targets.: We assume each target follows a linear Gaussian dynamic model:

$$f_{k|k-1}(x|\xi) = \mathcal{N}(x; F_{k-1}\xi, Q_{k-1}), \tag{5}$$

where $\mathcal{N}(\cdot; m, P)$ denotes a Gaussian density with mean m and covariance P. $F_k(\cdot)$ is the state transition matrix.

2) Sample the Binary Observation.: Each sensor reports a binary reading at time k, which forms a snapshot reading at time k. Based on the readings of "ON" sensors, we generate N_s random points in the sensor's sensing region, to be used as the observations of the sensor to the targets' locations. For simplicity of calculation, we assume the location observations follow an linear Gaussian model:

$$g_k(s|x) = \mathcal{N}(s; Hx, R_k) \tag{6}$$

s is an observation to the target's location at time k. H is observation matrix, which is set to a unit diagonal matrix. Note that when $N_s = 1$, the sensor's coordinates are used as the location sample. Although the observation error is not strictly normally distributed, because BPS network is not expected to provide accurate location observation, the assumption of Gaussian observation model is reasonable and acceptable in most cases.

B. IntenCT Framework

Based on above model, an Intensity-based target Couting and Tracking framework (IntenCT) was designed. The overview of the algorithm is shown in Fig.2. The intensity propagation recursion is based on GM-PHD filter[17]. The target grouping, counting, and locating are carried out by two-steps:

1) Density based clustering to find the target groups;
2) K-means clustering in each group to locate the targets.

Outputs of IntenCT at time k include 1) the number of targets, 2) the groups' locations, and 3) the locations of the targets.

C. Intensity Propagation

We revised GM-PHD filter to incorporate the BPS network model to track the propagation of intensity. The intensity is represented by the weighted sum of Gaussian Mixtures (GM). Benefited by the simplicity of the GM model, the key points in the recursion is therefore to update the weights, means, covariances, and the numbers of the GM components based on the sensor observations.

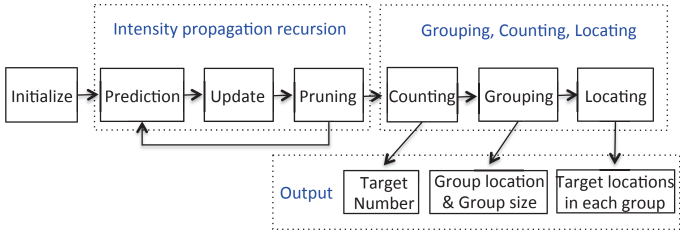

Fig. 2. Algorithm overview of IntenCT

1) Initialization: The algorithm initializes by generating $J_0 \geq 0$ Gaussian mixtures in the AOI. So that the initial intensity is:

$$v_0(x) = \sum_{i=1}^{J_0} \omega_0^{(i)} \mathcal{N}(x; m_0^{(i)}, P_0^{(i)}) \tag{7}$$

where $m_0^{(i)}$ is the mean of a Gaussian, which is randomly generated, and $P_0^{(i)}$ is the uniformly generated covariance. The time index k is set to 0.

2) Prediction: The prediction follows equation (3). But note that the integration can be calculated discretely in GM model. So the predicted intensity at time k is calculated as:

$$v_{k|k-1}(x) = v_{S,k|k-1}(x) + v_{\beta,k|k-1}(x) + \gamma_k(x) \tag{8}$$

where $v_{S,k|k-1}(x)$ is the predicted intensity of existing targets defined in (16); $v_{\beta,k|k-1}(x)$ is the intensity of newly spawned targets (e.g. targets separated from a group) defined in (17); $\gamma_k(x)$ is the intensity of newly appeared targets defined in (18). For clarity of presentation, the detailed formula are given in Appendix A. Note that, at time k, the number of GM components after prediction is denoted by $J_{k|k-1}$, which is at most $J_{\gamma,k} + J_{k-1}(1 + J_{\beta,k})$.

3) Update: Intensity update follows the equation (4). Note that using GM model, the posterior intensity at k is updated by updating the number, weights, means and covariances of GMs. Let's denote the intensity updated by observation s at time k by:

$$v_k(x, s) = \sum_{j=1}^{J_{k|k-1}} \omega_k^{(j)}(s) \mathcal{N}(x; m_{k|k}^{(j)}(s), P_{k|k}^{(j)}) \tag{9}$$

The formula of $\omega_k^{(j)}(s)$, $m_{k|k}^{(j)}(s)$, $P_{k|k}^{(j)}$ are given in (19)(20) in Appendix A. Note that the number of GM components after updating at k is at most $J_k = (1 + |S_k|)J_{k|k-1}$.

4) Pruning: Most of the GM components in J_k have very low weights and some of them are very close. To reduce the computational complexity, a pruning step is proposed to merge the small-weight and close-by GMs. In particular, the GM components who have weights less than threshold T were truncated, and components with distance less than U were merged.

If there are N_p out of J_k components have weights less then T, these N_p components will be truncated. The weights of the remaining components are enlarged by $\omega_k^{(i)} = \omega_k^{(i)} \frac{\sum_{j=1}^{J_k} \omega_k^{(j)}}{\sum_{j=N_p+1}^{J_k} \omega_k^{(j)}}$ to keep the sum of GM weights unchanged.

Then merging is carried out on the remained Gaussian components. From the component with the highest weight, denoted by i, all the other components within i's Mahalanobis radius U are found, denoted by set L. GMs in L are merged into one component l. The parameters of l are calculate by: *weight* : $\tilde{\omega}_k^{(l)} = \sum_{j \in L} \tilde{\omega}_k^{(j)}$; *mean location* $\tilde{m}_k^{(l)} = \frac{1}{\tilde{\omega}_k^{(l)}} \sum_{j \in L} \tilde{\omega}_k^{(j)} m_k^{(j)}$; and *covariance* $\tilde{P}_k^{(l)} = \frac{1}{\tilde{\omega}_k^{(l)}} \sum_{j \in L} \omega_k^{(j)} \left(P_k^{(j)} + (\tilde{m}_k^{(l)} - m_k^{(j)})(\tilde{m}_k^{(l)} - m_k^{(j)})^T \right)$. The merging process repeats to check the GM with the second largest weight, until at most $J_{k,max}$ components are left.

D. Grouping, Counting and Locating

The GM distribution reveals the expected target distribution. After pruning, density-based clustering algorithm is proposed to find the target groups; count the group size, and to locate the targets in each group.

1) Grouping: GM model provides both location mean and covariance, which enables to estimate Bhattacharyaa distance [1] between two Gaussians, which is defined as:

$$d_{i,j} = \frac{1}{8}(m_i - m_j)^T P^{-1} (m_i - m_j) + \frac{1}{2} \ln \left(\frac{\det P}{\sqrt{\det P_i \det P_j}} \right) \tag{10}$$

where $P = (P_i + P_j)/2$. Then a Bhattacharyaa distance-based clustering algorithm is designed to cluster the Gaussians to find the target groups.

We exploit DBSCAN[7], a density-based clustering algorithm to partition the GMs into adaptive number of clusters. Given the $J_{k,max}$ mean locations of Gaussians, DBSCAN groups together points that are closely packed, i.e., points with enough density. It has two parameters to define the density requirement, i.e., Eps and $MinPts$. They specify that at least $MinPts$ neighbors need to present within the Eps radius of a point in a cluster.

In parameter selection, we set $MinPts$ to N_s, i.e., the expected number of Gaussians generated by an ON sensor. Eps is set to 3, because a Gaussian preserves more than 95% of its energy within the region of Mahalanobis distance ≤ 3. This setting makes sure the Gaussians in different groups have very weak overlapping energy. Let N_G be the result number of groups and G_i contains the Gaussians in the ith group.

2) Counting: From equation (2), the number of targets in each group can be estimated by the integration of intensity in the group region. Under GM model, integration can be much simplified by calculating the summation of the weights of the Gaussians in the group like (11).

$$K_i = \sum_{j \in G_i} \omega_k^{(j)} \Big/ \rho \pi R^2 N_s, \forall i \in [1, \cdots, N_G] \tag{11}$$

Note that (11) is actually providing a lower bound of the target number in the group. Its correctness will be proved in Theorem 2.

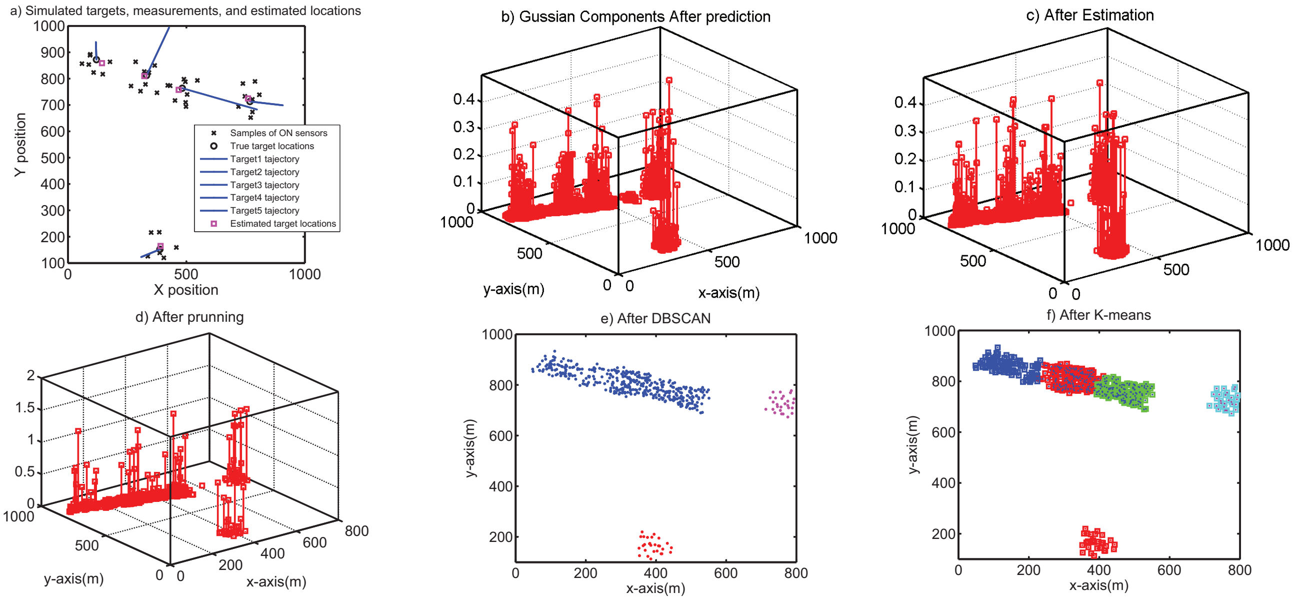

Fig. 3. An illustration of a target locating iteration by IntenCT

3) Locating: Let K_i be the estimated number of targets in the ith group. Then we find K_i representative locations in the group region to approximate the locations of the targets in the group. Since the Gaussians generated by a target tend to be in the vicinity centered by the target, we proposed a K-means based algorithm to determine the locations of the K_i targets.

Given the mean locations of Gaussians $\{m_k^{(j)}, j \in G_i\}$ in G_i and the estimated target number K_i, the task is to divide the locations into K_i sub-clusters, denoted by $\{T_1, \cdots, T_{K_i}\}$, by minimizing a square error function:

$$\{T_1, \ldots, T_{K_i}\} = \underset{\{T_1,\ldots,T_{K_i}\}}{\arg\min} \sum_{l=1}^{K_i} \sum_{m_j \in T_l} \left(m_j - \frac{\sum_{v \in T_l} m_v}{|T_l|} \right)^2 \tag{12}$$

The location of the lth ($l \in \{1, \cdots, K_i\}$) target in G_i is then given by the centroid of the lth sub-cluster:

$$\mu_l = \frac{\sum_{v \in T_l} m_v}{|T_l|}, l = \{1, \cdots, K_i\}, i = \{1, \cdots, N_G\} \tag{13}$$

Although finding the optimal classification in (12) is NP-hard, sub-optimal classification can be efficiently achieved using Lloyds-based K-means algorithm[10]. The algorithm for the *Grouping* and *Locating* steps are listed in Algorithm 1.

Algorithm 1 Grouping and Locating

Require: $\{\omega_k^{(j)}, m_k^{(j)}, P_k^{(j)}\}; J_k; D; N_s$
Ensure: $n_k; n_g; \{u_l, l = 1, \cdots, n_k\}$
1. Grouping $\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$
$Class := \text{DBSCAN}\left(\{m_k^{(j)}\}, \{P_k^{(j)}\}, N_s, D\right)$
$N_G := max\_class\_index(Class)$
for $i = 1; i <= N_G; i++$ **do**
$G_i := \{j \in \{1, \cdots, J_k\} | Class(j) == i\}$
end for
2. Locating $\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$
for $i = 1; i <= N_G; i++$ **do**
$K_i := \sum_{j \in G_i} \omega_k^{(j)} \Big/ \rho \pi R^2 N_s$
end for
Initialize target locations: $X_k = \emptyset$
for $i = 1 : 1 : N_G$ **do**
$points\_in\_G_i := \left\{ m_k^{(l)} \in \{m_k^{(1)}, \cdots, m_k^{(J_k)}\} | l \in G_i \right\}$
$Centroids := \text{KMEANS}(points\_in\_G_i, K_i)$
$X_k := X_k \cup Centroids$
end for

IV. Illustration and Property Analysis

A. Illustration

For better understanding of the IntenCT framework, an illustration was given in Fig.3 to interpret the key steps. The BPS network was deployed in an 1000*1000 area to track the mobile targets. The sensing radius of sensors was $50m$ and the distance between each sensor was $40m$. The figures were captured from the iteration when $k = 49$. In Fig.3a), the blue lines are the true trajectories of five targets. The stars are random samples generated by the ON sensors in this iteration ($N_s = 2$); the circles are targets' true locations; and the pink squares are the estimated locations of the targets. Fig.3b) shows the Gaussian components after the prediction step. Fig.3c) shows the Gaussians after estimation. Note that only the weights and the mean locations are plotted for clarity. The estimation step sharpened the Gaussian mixture of the prediction step. There were many Gaussian components after the estimation step. Fig.3d) shows the pruning result.

The Gaussians were truncated and merged. At most J_{max} Gaussians were kept after pruning.

Fig.3e) changes the angle to the top-view. The Gaussians after pruning were grouped by DBSCAN algorithm into three groups. Then the size of each group was counted and K-means algorithm was applied in each group. Fig.3f) shows the clustering results after K-means. The Gaussian components were successfully divided into five clusters. The centroid of each cluster is then used to estimate the location of a representative target, as indicated by the pink squares shown in Fig.3a). Note that the covariances of the Gaussians are embedded in the Bhattacharyaa distance, and the weights are used in integral calculating. We can see the locating is rather accurate.

B. Property Analysis

Theorem 1 (Quadratic complexity of intensity propagation): The complexity of intensity propagation recursion in each iteration is bounded by $O\left((nN_sJ_{max})^2\right)$, where n is the target number; R is sensing radius; ρ is sensor density; N_s is the number of samples generated by an "ON" sensor; and J_{max} is the number of Guassians preserved in pruning.

Proof 1: The number of GMs generated after update is at most $J_k = (1 + |S_k|)J_{k|k-1}$. The most computationally intensive step is to merge the J_k GMs in the pruning step. For each component in J_k, we need to find all its neighbors within a threshold. The complexity is $O((J_k)^2)$. Since $J_{k|k-1} = J_{\gamma,k} + J_{k-1}(1 + J_{\beta,k}) \leq J_{\gamma,k} + J_{max}(1 + J_{\beta,k})$ and $J_{\gamma,k}$ and $J_{\beta,k}$ are constants which are much smaller than J_{max}, so $J_k \leq c(1+|S_k|)J_{max}$, where c is a constant. Further, The expected value of $|S_k|$ is upper bounded by $n\rho\pi R^2 N_s$, in the case the targets are well separated. So $O((J_k)^2)$ has complexity $O\left((nN_sJ_{max})^2\right)$.

Theorem 2 (Target number lower bound): The expected number of targets in the AOI at time k is lower bounded by:

$$n_k = \sum_{i=1}^{J_k} \omega_k^{(i)} \Big/ \rho\pi R^2 N_s \tag{14}$$

Proof 2: When sensor density is ρ and sensing radius is R, the expected number of sensors triggered "ON" by a target is $\rho\pi R^2$. Since each "ON" sensor generates N_s particles, one target is expected to generate $\rho\pi R^2 N_s$ particles. Since some targets maybe overlapping or very close, leading to unobservable by sensors, the total number of particles generated by n targets is less than $\rho\pi R^2 N_s n$. Since the integral of intensity equals to the expected number of particles in AOI, therefore, $n \geq \frac{\int_{\mathcal{X}} v_k(x)dx}{\rho\pi R^2 N_s}$. Under Gaussian model, the integration can be calculated as $\sum_{i=1}^{J_k} \omega_k^{(i)}$. Therefore, the inequality (14) provides a lower bound for the expected number of targets in the AOI.

Theorem 3 (Locating complexity): The complexity of grouping and locating steps is $O(J_{max}^2 + 2nJ_{max}I)$, where n is the target number and I is the number of iterations in the implementation of K-means.

Proof 3: The computation complexity of DBSCAN is $O(N^2)$, where $N = J_{max}$ in this problem. The complexity of Lloyds implementation of K-means is $O(NkId)$, where $N = J_{max}$; k is at most n; $d = 2$; and I is the number of iteration in the K-means algorithm implementation.

V. Augmentation

The theorems show IntenCT is very efficient. We can further augment the algorithms for more accurately target counting and trajectory connection.

A. Augment Target Counting

In real environments, target cannot appear or disappear suddenly. We call this phenomenon *target number continuity*. In addition, targets can leave only at some special locations, called *exits*. New targets can only be spawned from existing groups, or enters from some specific locations called *entrances*. The number continuity, the leaving and appearing conditions can be leveraged to augment the target counting and locating.

1) Characteristics of counting error: The counted target number is smaller than the ground truth in two cases: (i). When some targets move into close vicinity of each other. (ii). When some targets move close to the boundary, in which some part of the Gaussian distribution is beyond the integral scope. Such counting error can be addressed by the target number continuity. In most cases, number of targets in the AOI is static. Even if some targets move to be overlapped with others, i.e. be unobservable, we should maintain the *the maximum target number ever seen* as the estimated target number, except that we make sure some targets has left from the exit locations.

2) *Feasibility of new target appearing.* In addition, if the target number estimated at k, i.e, $n_k > n_{k-1}$, we should check whether the appearance of the new target is feasible. In specific, we check whether there is an exist groups in V_{max}-radius of the target or whether the target is in the V_{max}-vicinity of the entrance locations. If both are negative, we judge the target appearance is an error. In practice, we can specify some area of AOI as doors for target entering or exiting.

B. Connecting Trajectories

IntenCT achieves efficiency by avoiding target trajectory association. But if target trajectories are desired, the target trajectories can also be estimated by a post-processing method. Since we have the estimated locations from time 1 to k, although the target identities are not known, we can approximate the target trajectories by motion consistency and location continuity. Suppose we got n_{k-1} location estimates at time $k-1$, i.e., $\{X_{k-1,1}, \cdots, X_{k-1,n_{k-1}}\}$ and got n_k location estimates at time k, i.e., $\{X_{k,1}, \cdots, X_{k,n_k}\}$. The possible number of trajectory associations from $k-1$ to k is $n_k n_{k-1}$. We present method to connect locations at $k-1$ to locations at k to form n_k trajectories.

1) Setup association set: At first, targets' movements are bounded by velocity bound. We assume a target's maximum velocity is V_{max}. For any $X_{k-1,i}$, $i = \{1, \cdots, n_{k-1}\}$,$j = \{1, \cdots, n_k\}$ it checks $||X_{k-1,i} - X_{k,j}||_2 < V_{max}$. If $X_{k,j}$ satisfies above inequality, it will be added to the possible association set of $X_{k-1,i}$; After this checking, the association set for $X_{k-1,i}$ is denoted by $\mathbf{A}_i$.

2) Motion consistency: To find the most possible association for $X_{k-1,i}$ from $\mathbf{A}_i$, let's check the consistency of motion. A cost function was defined to check the variance of velocity from $k-1$ to k.

$$\forall X_{k,j} \in \mathbf{A}_i, C_{i,j} = \|(X_{k,j} - X_{k-1,i}) - (X_{k-1,i} - X_{k-2,i})\|_2 \tag{15}$$

from which, $X^*_{k,j} \in \mathbf{A}_i$ with the smallest consistency cost will be connected to $X_{k-1,i}$ as its next location. If $k = 1$, i.e., if historical velocity is not known, the closest point in $\mathbf{A}_i$ will be chosen. If two points select the same $X_{k,j}$, the one with the larger consistency cost releases its choice to select the one with the second smaller cost for one-to-one mapping.

3) When target number is not equal: if $n_k > n_{k-1}$, the association for $\{X_{k-1,i}\}$ is the same as $n_k = n_{k-1}$. Unassociated new targets are detected as $\{X_{k,new}\}$. If $n_k < n_{k-1}$, the unassociated targets at $k-1$ are detected as left targets and denoted by $\{X_{k-1,left}\}$. They will be further checked by the counting augmentation method.

Algorithm 2 Trajectory Connection and Counting Augmentation

Require: $\{\mathbf{X}_{k-1}\}$, $\{\mathbf{X}_k\}$, $\{m_{ex}\}, \{m_{en}\}, n_k, n_{k-1}, V_{max}$
Ensure: $\{X_{k,j}\}, n_k$

```
1. Trajectory connection ·········
for i = 1; i <= n_{k-1}; i++ do
    A_i := {X_{k,j} ∈ X_k | |X_{k-1,i} − X_{k,j}||_2 < V_max}
    for j ∈ A_i do
        j* := argmin_{j∈A_i} ||(X_{k,j} − X_{k-1,i}) − (X_{k-1,i} − X_{k-2,i})||_2
    end for
    Connect X_{k-1,i} to X_{k,j*};
    conflict_resolvation() ; Detect {X_{k,new}} or {X_{k-1,left}}
end for
2. Counting Augmentation ·········
if n_k > n_{k-1} then
    for j ∈ {X_{k,new}} do
        Accept j, n_k=n_{k-1}+1, if ∃k ∈ {m_en}, s.t.||X^j_{k,new} −
        m^k_en|| ≤ V_max or ∃i ∈ {X_{k-1}}, s.t.||X^j_{k,new} − X_{k-1,i}|| ≤
        V_max
    end for
end if
if n_k < n_{k-1} then
    for j ∈ {X_{k-1,left}} do
        Remove j, n_k = n_{k-1} − 1, if ∃k ∈ {m_out}, s.t.
        ||X^j_{k-1,left} − m^k_en|| ≤ V_max
    end for
end if
```

Algorithm 2 gives the algorithm sketch for trajectory connection and counting augmentation. Note that the possible locations for target entering and leaving are denoted by sets $\{m_{ex}\}, \{m_{en}\}$. The sets $\{X_{k,new}\}$ denotes the unassociated newly estimated locations at k and $X_{k-1,left}$ denotes the unassociated locations at $k-1$ which is considered as the locations of target leaving. The association and the augmented processes have low complexity. Wrong trajectory may happen when two targets suddenly change their velocities together. But such case is difficult for all motion consistency based trajectory decomposition methods.

VI. PERFORMANCE EVALUATION

Performances of IntenCT were evaluated from three aspects: 1) efficiency, 2) counting accuracy and 3) tracking accuracy.

A. Simulation settings

Simulations were conducted by Matlab2013. In simulations, BPS networks were deployed in an AOI of $2000(m)$ * $2000(m)$ to track n mobile targets, in a grid topology with the grid size $d = \frac{1}{\sqrt{\rho}}$. So that the sensor density is $\rho = \frac{1}{d^2}$ without considering the boundary effect. The sensing radius of each sensor is R. We set $d < \sqrt{\pi R^2}$ in all simulations, i.e., $\rho > \frac{1}{\pi R^2}$ to keep the fully coverage requirement.

Each target follows the linear dynamic model:

$$F_k = \begin{bmatrix} 1 & 0 & \Delta & 0 \\ 0 & 1 & 0 & \Delta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad Q_k = \sigma^2 \begin{bmatrix} \frac{\Delta^4}{4} & 0 & \frac{\Delta^3}{2} & 0 \\ 0 & \frac{\Delta^4}{4} & 0 & \frac{\Delta^3}{2} \\ \frac{\Delta^3}{2} & 0 & \Delta^2 & 0 \\ 0 & \frac{\Delta^3}{2} & 0 & \Delta^2 \end{bmatrix}$$

where Δ is the time step length, which is set to 1. $\sigma = 5(m/s^2)$ is the standard derivation of processing noise. Each target has a random starting location and moves towards a random destination following the above dynamic motion model. Each "ON" sensor generates N_s random location estimations in its "ON" region, where the "ON" region is the union of the "ON" patches. All the patch areas are initialized by the geographical method in [16].

B. Efficiency Performance

1) Computation cost: As a major benefit of IntenCT, we firstly evaluate the efficiency of IntenCT. The running times of 100 step simulations for experiments of different number of targets were evaluated. Although the execution time maybe a biased metric for evaluating the efficiency because of the different implementation details, the comparison across different number of targets showed the trend of the computation cost increment. Fig.4 shows the running time results when the target number varied from 1 to 35. We set $J_{max} = 100$ and 200 respectively in the two series of experiments. **We can see that the computation cost increased as an quadratic function of n in both experiments.** This highly coincides our complexity analysis.

2) Number of Generated GMs. In addition, we also investigated the maximum number of Gaussian components generated in each iteration in different experiments to have an insight to the computation complexity. Fig.5 shows the most number of GMs generated in different experiments. The most number of generated GMs increased almost linearly with n. This confirms the complexity analysis in Theorem 1.

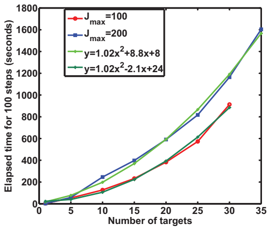

Fig. 4. Running time.

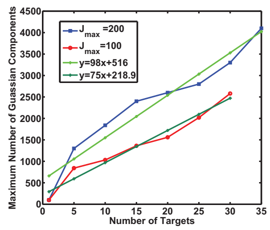

Fig. 5. Most number of generated GMs

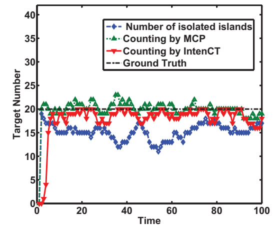

Fig. 6. Count 20 targets.

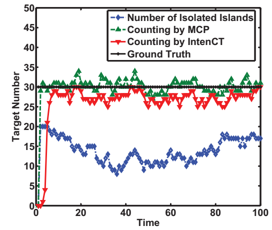

Fig. 7. Count 30 targets.

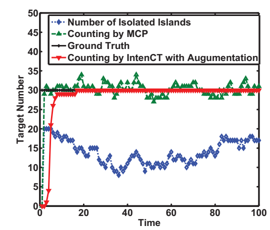

Fig. 8. Counting with augmentation.

C. Counting Accuracy

The counting accuracy was compared with *1) the ground true of the target number; 2) the previously best known target number lower bound [16] calculated by the minimum clique partition (MCP) of the UDG formed by the ON sensors; 3) the number of isolated islands separated by the OFF sensors*. We evaluated the cases when augmentation method was used or not used respectively.

1) Without Counting Augmentation: At first, two experiments were conducted without counting augmentation, in which 20 targets and 30 targets were generated respectively at arbitrary locations and moved towards random designations. In IntenCT, the target number was counted using Equation (14). The lower bound of the target number given by MCP was calculated PTAS approximation method in [16]. Fig.6 and Fig.7 compare the counting accuracy performance. We can see that IntenCT provides a very tight lower bound. The calculated MCP lower bound may sometimes exceeds the ground truth, because the MCP was calculated by a PTAS approximation, which was not the exact MCP. Calculating the exact MCP is NP-hard. Therefore, IntenCT provide an easily calculated, accurate lower bound for the target number. Both of them performed much better than the method of using the number of isolated island.

2) With Counting Augmentation: Then we evaluated how the augmentation method can improve the counting accuracy. For the same 30 targets experiment, we considered only the boundary points of the AOI were feasible exit locations. The target number continuity were exploited to enhance the target counting accuracy. The counting results were shown in Fig.8. We can see the counting accuracy was dramatically improved than the case without augmentation, which approached the ground truth very well.

D. Location and Tracking Accuracy

1) Location Error Metric: An Optimal Sub-Pattern Assignment (OSPA) metric [13] which was widely used in evaluating the locating accuracy of multiple target tracking was applied to evaluate the location accuracy. Given the estimated location $X = \{x_1, \cdots, x_n\}$, ground truth location $Y = \{y_1, \cdots, y_m\}$, the OSPA metric is: $\bar{d}_p^c(X,Y) := \left(\frac{1}{n}\left(\min_{\pi\in\Pi_n}\sum_{i=1}^{m} d^c(x_i, y_{\pi(i)})^p + c^p(n-m)\right)\right)^{\frac{1}{p}}$ if $m \leq n$; and $\bar{d}_p^c(X,Y) = \bar{d}_p^c(Y,X)$ if $m > n$, where $c > 0$ and $\infty > p \geq 1$ are two constants; Π_n is the set of permutations on $\{1, 2, \cdots, n\}$. Note that a penalty to the counting error was included in OSPA.

2) Location Accuracy Comparing with MCP-centroid: MCP-centroid method in [16] calculated the centroid of the minimum cliques formed by the minimum clique partition. We compared the OSPA performances of IntenCT based tracking results and that of MCP-centroid based tracking results in different experiments. In an experiment of tracking 15 targets using sensors of radius 60m and grid distance 60m, the OSPA performances were shown in Fig.9. The corresponding location estimation results and the ground truth of target trajectories were illustrated in Fig.10. It can be seen that IntenCT provides accurate location estimation to the multiple targets; shows better location accuracy than MCP-centroid method.

To better understand how the locating accuracy was affected by the sensor density and target number, Fig.11 shows the cumulated probability distribution (CDF) of the locating accuracies in tracking different number of targets. We evaluated the cases when target number was 10, 20, 30 respectively. Each experiments were run ten times and the CDFs of OSPA were evaluated. It can be seen that IntenCT's performed better then MCP-centroid in all the cases. When $n = 10$, the location error of IntenCT has more than 93% probability to be less than 20m. Fig.12 shows the CDFs of location error against the sensor density. In the experiments, the grid size, i.e., distance between neighboring sensor was set 40m, 60m, 80m respectively with the sensor radius being 60m. We can see IntenCT has better accuracy when sensors are denser. This accorded our general knowledge. IntenCT performed better than MCP-centroid method in all these settings.

2) Tracking Accuracy:

Fig.13 evaluated the performance of the proposed trajectory connection algorithm, i.e, Algorithm 2. The experiment is to track ten targets. We can see that the target trajectories were rater correctly constructed. The trajectories of different targets were assigned different colors.

VII. CONCLUSION

This paper presented IntenCT, an efficient multi-target counting and tracking method using binary proximity sensors. It evolves *intensity*, i.e., the first moment of the probability density distribution of the targets to address the association complexity in multiple target tracking by BPS networks. We

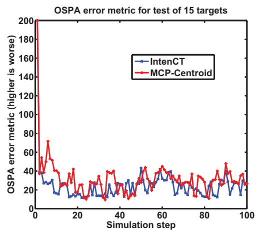

Fig. 9. OSAP in locating 15 targets

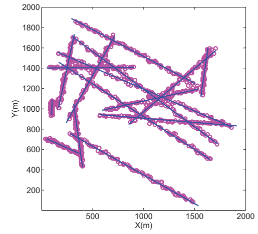

Fig. 10. Location vs. ground truth

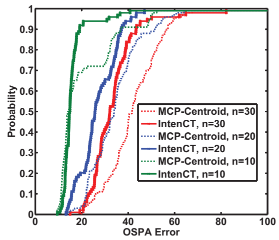

Fig. 11. CDF of OSPA

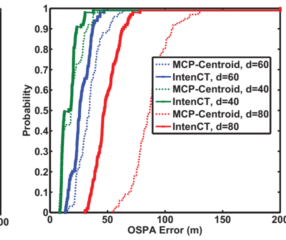

Fig. 12. CDF of OSPA

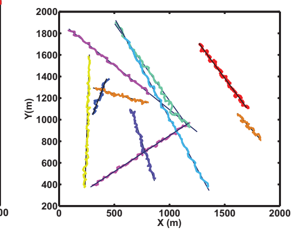

Fig. 13. Trajectory connecting results

showed that the proposed IntenCT framework had quadratic complexity to the number of targets, i.e., $O((nN_sJ_{max})^2)$. Extensive experiments showed that it provided an easily calculated, tight lower bound to the target number. Augmentation methods to improve the counting accuracy and to estimate target trajectories were presented. The experiments showed that the augmented counting approached the target number ground truth very well. Its tracking performance also outperformed the state-of-the-art MCP-centroid method in almost all experiments. In future, since IncenCT is very efficient, it is feasible to develop practical BPS tracking systems, such as infrared sensor based systems, in practice.

VIII. Appendix

A. Formula in prediction step

$$v_{S,k|k-1}(x) = \sum_{j=1}^{J_{k-1}} \omega_{k-1}^{j} N(x; m_{S,k|k-1}^{(j)}, P_{S,k|k-1}^{(j)}) \quad (16)$$

$$v_{\beta,k|k-1}(x) = \sum_{j=1}^{J_{k-1}} \sum_{l=1}^{J_{\beta,k}} \omega_{k-1}^{(j)} \omega_{\beta,k}^{(l)} \mathcal{N}(x; m_{\beta,k|k-1}^{(j,l)}, P_{\beta,k|k-1}^{(j,l)}) \quad (17)$$

$$\gamma_k(x) = \sum_{i=1}^{J_{\gamma,k}} \omega_{\gamma,k}^{(i)} \mathcal{N}(x; m_{\gamma,k}^{(i)}, P_{\gamma,k}^{(i)}) \quad (18)$$

where the close-form formula for $m_{S,k|k-1}^{(j)}$, $P_{S,k|k-1}^{(j)}$, $m_{\beta,k|k-1}^{(j,l)}$, $P_{\beta,k|k-1}^{(j,l)}$ are given in [17].

B. Formula in update step

The weights, mean locations and covariances are updated following GM-PHD filter [17]:

$$\omega_k^{(j)}(s) = \frac{\omega_{k|k-1}^{(j)} \mathcal{N}(s; H_k m_{k|k}^{(j)}, R_k + H_k P_{k|k-1}^{(j)} H_k^T)}{\sum_l^{J_{k|k-1}} \omega_{k|k-1}^{(l)} \mathcal{N}(s; H_k m_{k|k}^{(l)}, R_k + H_k P_{k|k-1}^{(l)} H_k^T)} \quad (19)$$

$$\begin{aligned} m_{k|k}^{(j)}(s) &= m_{k|k-1}^{(j)} + K_k^{(j)} \left(s - H_k m_{k|k-1}^{(j)} \right), \\ P_{k|k}^{(j)} &= \left[I - K_k^{(j)} H_k \right] P_{k|k-1}^{(j)}, \quad (20) \\ K_k^{(j)} &= P_{k|k-1}^{(j)} H_k^T \left(H_k P_{k|k-1}^{(j)} H_k^T + R \right)^{-1} \end{aligned}$$

IX. Acknowledgement

This paper was partially supported by the Fundamental Research Funds for the Central University, and the Research Funds of Remin University of China, 2015030273; The National Natural Science Foundation of China, No. 61202360; Hong Kong scholars program

References

[1] http://en.wikipedia.org/wiki/bhattacharyya.
[2] Passive infrared sensor, Mar. 2013. Page Version ID: 547502393.
[3] D. Cao, B. Jin, and J. Cao. On group target tracking with binary sensor networks. In *MASS'08*, pages 334–339, 2008.
[4] D. Cao, B. Jin, S. K. Das, and J. Cao. On collaborative tracking of a target group using binary proximity sensors. *Journal of Parallel and Distributed Computing*, 70(8):825–838, Aug. 2010.
[5] D. De, W.-Z. Song, M. Xu, C.-L. Wang, D. Cook, and X. Huo. FindingHuMo: real-time tracking of motion trajectories from anonymous binary sensing in smart environments. In *ICDCS'12*, pages 163–172, 2012.
[6] J. Elson, L. Girod, and D. Estrin. Fine-grained network time synchronization using reference broadcasts. *SIGOPS Oper. Syst. Rev.*, 36(SI):147–163, Dec. 2002.
[7] M. Ester, H.-p. Kriegel, J. S, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231. AAAI Press, 1996.
[8] W. Kim, K. Mechitov, J.-Y. Choi, and S. Ham. On target tracking with binary proximity sensors. In *IPSN'05*, pages 301–308, Apr. 2005.
[9] T. Li, Y. Wang, L. Song, and H. Tan. On target counting by sequential snapshots of binary proximity sensors. In *EWSN'14*, 2014.
[10] S. Lloyd. Least squares quantization in PCM. *IEEE Trans. Inf. Theor.*, 28(2):129–137, Sept. 2006.
[11] R. Mahler. Multitarget bayes filtering via first-order multitarget moments. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1152–1178, Oct. 2003.
[12] A. Savvides, C.-C. Han, and M. B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. MobiCom '01, pages 166–179, New York, NY, USA, 2001. ACM.
[13] D. Schuhmacher, B.-T. Vo, and B.-N. Vo. A consistent metric for performance evaluation of multi-object filters. *IEEE Transactions on Signal Processing*, 56(8):3447–3457, Aug. 2008.
[14] N. Shrivastava, R. M. U. Madhow, and S. Suri. Target tracking with binary proximity sensors: fundamental limits, minimal descriptions, and algorithms. SenSys '06, pages 251–264, New York, NY, USA, 2006. ACM.
[15] J. Singh, U. Madhow, R. Kumar, S. Suri, and R. Cagley. Tracking multiple targets using binary proximity sensors. IPSN '07, pages 529–538, New York, NY, USA, 2007. ACM.
[16] L. Song and Y. Wang. Multiple target counting and tracking using binary proximity sensors: Bounds, coloring, and filter. MobiHoc '14, pages 397–406, New York, NY, USA, 2014. ACM.
[17] B.-N. Vo and W.-K. Ma. The gaussian mixture probability hypothesis density filter. *IEEE Transactions on Signal Processing*, 54(11):4091–4104, Nov. 2006.
[18] C. Wang, D. De, and W.-Z. Song. Trajectory mining from anonymous binary motion sensors in smart environment. *Knowledge-Based Systems*, 37:346–356, Jan. 2013.