Multiple Target Tracking with RF Sensor Networks

Maurizio Bocca, Ossi Kaltiokallio, Neal Patwari, *Member, IEEE*, and Suresh Venkatasubramanian

Abstract—RF sensor networks are wireless networks that can localize and track people (or targets) without needing them to carry or wear any electronic device. They use the change in the received signal strength (RSS) of the links due to the movements of people to infer their locations. In this paper, we consider real-time multiple target tracking with RF sensor networks. We apply radio tomographic imaging (RTI), which generates images of the change in the propagation field, as if they were frames of a video. Our RTI method uses RSS measurements on multiple frequency channels on each link, combining them with a fade level-based weighted average. We introduce methods, inspired by machine vision and adapted to the peculiarities of RTI, that enable accurate and real-time multiple target tracking. Several tests are performed in an open environment, a one-bedroom apartment, and a cluttered office environment. The results demonstrate that the system is capable of accurately tracking in real-time up to four targets in cluttered indoor environments, even when their trajectories intersect multiple times, without mis-estimating the number of targets found in the monitored area. The highest average tracking error measured in the tests is 0.45 m with two targets, 0.46 m with three targets, and 0.55 m with four targets.

Index Terms—Radio tomography, multiple target tracking, device-free localization, wireless sensor networks

1 Introduction

Radio frequency (RF) sensor networks are wireless networks that monitor the changes in the received signal strength (RSS) of the links in order to localize and track people, without requiring them to carry or wear any radio device. In these systems, the RSS measurements can be processed to form images of the changes in the propagation field of the monitored area in presence of moving people and objects - a process named radio tomographic imaging (RTI) [1]. Unlike previous works [1]–[4] which have focused on single target localization and tracking, this work aims at contributing to the growing field of device-free localization (DFL) through RF sensor networks [1], [5] by presenting a system capable of accurately tracking in real-time multiple people (or *targets*) moving in real-world indoor environments where low-power transceivers are deployed.

The potential applications of RF sensor networks are many, including smart buildings and perimeter surveillance, ambient assisted living and residential monitoring [6], breathing detection [7], and security and rescue operations [8]. Compared to other sensing technologies applied in indoor DFL, such as infrared, ultrasonic range finders [9], ultra-wideband (UWB) radios and video cameras, RF sensor networks provide several advantages: they work in the dark and can penetrate smoke and walls; they are less invasive in domestic environments than video camera networks; they are significantly less expensive than UWB transceivers; their installation and maintenance time is minimal.

People moving in an area where wireless transceivers are deployed affect the propagation of the radio signals by shadowing, reflecting, diffracting or scattering a subset of their multipath components [5], [10]–[12]. In open and uncluttered environments, where line-of-sight (LoS) communication among the transceivers is predominant, a person obstructing a link line will generally cause attenuation of the radio signal. This phenomenon has been successfully applied for DFL in several works [1]–[4]. However, in cluttered environments, where multipath propagation is predominant, the change in RSS due to the presence of a human body becomes more unpredictable. As the link line is obstructed, the RSS can also remain constant or increase [12]. In addition, due to the multipath propagation of the radio signals, people can affect the RSS also when located far away from the link line [13].

Most of the research presented so far in the area of DFL with RF sensor networks has considered the situation in which only one target had to be located and tracked [1]–[4]. However, the real-world scenarios in which these systems are to be used often require the localization and tracking of multiple targets. Moreover, the DFL system should be able *a)* to perform the localization and tracking tasks in real-time even when the targets trajectories intersect, and *b)* to correctly estimate the number or targets to be tracked as

- *M. Bocca is with the Department of Electrical and Computer Engineering, University of Utah, Salt Lake City, UT 84103 USA. E-mail: maurizio.bocca@utah.edu.*
- *O. Kaltiokallio is with the Department of Automation and Systems Technology, Aalto University, Helsinki 02150, Finland. E-mail: ossi.kaltiokallio@aalto.fi.*
- *N. Patwari is with the Department of Electrical and Computer Engineering, University of Utah, and also with Xandem Technology, Salt Lake City, UT 84103 USA. E-mail: npatwari@ece.utah.edu.*
- *S. Venkatasubramanian is with the School of Computing, University of Utah, Salt Lake City, UT 84103 USA. E-mail: suresh@cs.utah.edu.*

Manuscript received 21 Nov. 2012; revised 11 June 2013; accepted 5 July 2013. Date of publication 24 July 2013; date of current version 7 July 2014. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier 10.1109/TMC.2013.92

people enter and exit the monitored area. In this work, we tackle these challenges.

The contributions of our work are twofold. First, in the process of estimating RTI images, we apply a novel method that weights the RSS measurements based on the fade level [14] of the frequency channels on which the RSS was measured. Second, we adapt machine vision methods to RTI and use the new methods to process the RTI images in real-time to detect and track the blobs corresponding to real targets. Multiple target tracking with RF sensor networks is made more challenging in that in RTI the targets have to be modeled as points, consisting only of position, velocity and acceleration, neither having physical length or other features such as shape, color, or size. Although the blob size and shape in RTI are loosely related to a person's size, the blob dramatically changes depending on a person's position in the monitored area and on the positions of other people in the area. Thus, the noise in the blob shape overwhelms any attempt to determine a person's features. We invite the reader to view an RTI video at [15] as an example of the capabilities and limitations of this imaging modality. In addition, due to measurement noise and the nonlinear fading caused by the simultaneous presence of multiple targets in the same cluttered indoor environment, spurious blobs (not corresponding to real targets) can appear in the image. For the same reasons, blobs corresponding to real targets can temporarily disappear from the image. These factors dramatically increase the difficulty of multiple target tracking, especially when targets have intersecting trajectories.

We evalute the performance of the multi-target tracking system in three different indoor environments, *i.e.* a $70m^2$ open environment with no obstructions nor objects, a $58m^2$ one-bedroom apartment with internal walls, furniture and various objects, and a $67m^2$ heavily cluttered office environment. The few multi-target tracking methods developed for RSS-based DFL are either non-real time [14], [16], track only two or three people [14], [16]–[19], assume that the number of targets is fixed and known a priori [14], [18]–[20], or do not attempt to track targets with intersecting trajectories [14], [16]–[20]. In an experimental test with four targets having separated trajectories, the method in [21] achieves as low as 0.63 m average accuracy, but consuming 7.6 seconds to process an RTI image. The system in [22] is used to track up to four targets having partially overlapping trajectories (only two of the four targets walk side by side). The best accuracy with four targets is 1.08 m in a $150m^2$ office environment and 1.49 m in a $400m^2$ open environment. In the case of overlapping trajectories, the system estimates correctly the number of targets in the monitored area 80% of times. Our system achieves 0.55 m average accuracy in an experimental test with four targets having intersecting trajectories, consuming only 13.3 milliseconds to process an RTI image, thus enabling tracking in real-time. In addition, the system estimates correctly the number of targets found in the monitored area at least 97% of times, even when the number is not known a priori and varying. The highest average tracking error measured in the tests is 0.45 m with two targets, 0.46 m with three targets, and 0.55 m with four targets. Videos showing the performance of the multi-target tracking system during the tests described in this paper can be found at [15]. Our work demonstrates that RF sensor networks can be used in real-world indoor environments to accurately track in real-time an unknown and varying number of targets even when their trajectories intersect multiple times.

The remainder of the paper is organized as follows. We present the methods used to form radio tomographic images and to track multiple targets in Section 2 and 3, respectively. The experiments carried out are described in Section 4. We introduce the metrics used to evaluate the performance of the system in Section 5. The experimental results are listed and discussed in Section 6. The related work is in Section 7. Conclusions are drawn in Section 8.

2 Multi-Channel RTI

In this section, we describe how the RSS measurements collected on multiple channels are processed in real-time to form radio tomographic images. We deploy R sensors at positions $\{\mathbf{z}_r\}_{r=1,...,R}$. At time instant k, we measure the RSS $r_{l,c}(k)$ in dBm of link l on channel $c \in \{1, ..., K\}$. Our objective is to estimate the change in the propagation field of the monitored area, $\mathbf{x}$, from the RSS measurements collected on all the links of the network.

2.1 Fade Level Estimation

In obstructed environments, radio signals propagate from the transmitter to the receiver via multiple paths. At the receiver, a phasor sum of the waves impinging on the antenna determine the RSS. Depending on the relative phase of each wave, the waves may add constructively or destructively. As a wave's phase is a function of the center frequency and path length, the RSS is a function of the center frequency and position of the communicating devices, an effect called multipath fading [23]. During an initial calibration period performed in static conditions, *i.e.*, when the monitored area is empty, we measure the average RSS of each link l on each different frequency channel c, $\bar{r}_{l,c}$, which can be modeled as:

$$r_{l,c} = \tilde{P}_c - L_{l,c} - S_{l,c} + F_{l,c} - \eta_{l,c}, \quad (1)$$

where $\tilde{P}_c$ is the actual transmit power, in dBm, on channel c, $L_{l,c}$ the large scale path loss, $S_{l,c}$ the shadowing loss, $F_{l,c}$ the fading gain (or *fade level* [14]), and $\eta_{l,c}$ the measurement noise. For the wireless sensors used in the experiments (see Section 4.1), the actual transmit power $\tilde{P}_c$ is different from the nominal one and, most importantly for estimating the fade level, is a function of the frequency channel c, in part due to the difficulty in antenna impedance matching across a wide frequency band [24].

The relation between steady-state, narrow-band fading and the temporal fading statistics of the RSS due to human movement has been described in [14], where the authors define the concept of *fade level*, a continuum between two extremes: *deep fade* and *anti-fade*. For a link in a deep fade, when the link line is obstructed, the RSS, on average, increases. In addition, deep fade links show changes of the RSS even when the person is at some positions far away from the link line. On the contrary, for a link in an anti-fade, when the link line is obstructed, the RSS, on average, decreases. Moreover, anti-fade links show changes of the

RSS only when the person is in the close proximity of the link line.

Due to the limited size and predictable shape of their sensitivity area, the anti-fade links are the most informative for DFL. In [6], [13], the channels used to form RTI images were ranked based on their fade level, from the most anti-fade to the most deep fade. The RSS measurements of the Z most anti-fade channels of each link were then used to form RTI images. However, the results showed that the number Z of frequency channels to be used which provided the highest localization accuracy depended on the deployment area and on the position of the sensors, thus making it impossible to know from the start what value to use for Z. In this work, we use channel diversity and propose a novel approach to overcome the limitation of the method in [13] by considering the RSS measurements of all the channels and weighting them proportionally based on their fade level.

The large scale path loss $L_{l,c}$ and shadowing loss $S_{l,c}$ in (1) both change very slowly with the center frequency. Since the 16 frequency channels available with the ZigBee, 802.15.4-compliant nodes used in the experiments (see Section 4) span over 80 MHz in the 2.4 GHz ISM band, we assume that both do not depend on the frequency channel c. Consequently, $F_{l,c}$ can be calculated as:

$$F_{l,c} = r_{l,c} - \tilde{P}_c + \eta_{l,c}. \tag{2}$$

Thus, the fade level of a link on a frequency channel can not be measured directly due to the measurement noise and also because $L_{l,c}$ and $S_{l,c}$ are not known. In order to *estimate* it, we perform an initial calibration of the system in static conditions, *i.e.*, when the monitored area is empty, and measure the average RSS, $\bar{r}_{l,c}$, of link l on channel c. The lowest $\bar{r}$ measured on the used frequency channels is used as a reference to estimate the fade level of channel c:

$$F_{l,c} = \bar{r}_{l,c} - \min_c \bar{r}_{l,c}. \tag{3}$$

Thus, for the same link l, channel c_1 is in a deeper fade than channel c_2 if $F_{l,c1} < F_{l,c2}$. Note that $F_{l,c} \geq 0$, and that $F_{l,c} = 0$ for one channel c on each link.

2.2 Measurement Model

The RSS measurements collected on different channels are weighted based on the fade level defined in (3), bearing in mind that anti-fade channels provide measurements more informative for localization. The weighted average change in RSS of link l at time k is computed as:

$$y_l(k) = \frac{1}{\sum_{c\in K} F_{l,c}} \sum_{c\in K} F_{l,c} \cdot |\Delta r_{l,c}(k)|, \tag{4}$$

where $\Delta r_{l,c}(k)$ is the difference between the RSS of link l on channel c measured at time k, $r_{l,c}(k)$, and the reference RSS, $\bar{r}_{l,c}$. While in [13] the Δr of the m most anti-fade channels were averaged, in (4) the Δr of each channel is assigned a weight directly proportional to the estimated fade level.

2.3 Image Estimation

The change in RSS is assumed to be a spatial integral of the propagation field of the monitored area. When the propagation field is discretized, some voxels affect the RSS of a specific link, whereas others do not. Thus, the change in RSS of each link is assumed to be a linear combination of the change caused by each voxel:

$$y_l(k) = \sum_{j=1}^{N} w_{lj} x_j + n_l, \tag{5}$$

where x_j is the change in RSS caused by voxel j, w_{lj} the weight of voxel j for link l, N the number of voxels in the discretized image and n_l the noise of link l. The weight w indicates how each voxel of the image affects each link. For this, we use an ellipse model [1], [12], [13] in which the transmitter and receiver are located at the foci of the ellipse. According to this model, a voxel j at position $\mathbf{v}_j$ that is located within the ellipse of link l has its weight w_{lj} set to a constant, which is inversely proportional to the area of the ellipse. Otherwise its weight is set to zero, as follows:

$$w_{lj} = \begin{cases} \frac{1}{A_l} & \text{if } d_{lj}^{tx} + d_{lj}^{rx} < d_l + \lambda \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

where $d_{lj}^{tx} = \|\mathbf{z}_{l_{tx}} - \mathbf{v}_j\|$ and $d_{lj}^{rx} = \|\mathbf{z}_{l_{rx}} - \mathbf{v}_j\|$, $d_l = \|\mathbf{z}_{l_{tx}} - \mathbf{z}_{l_{rx}}\|$, A_l is the area of the ellipse, and λ is its excess path length, *i.e.*, the parameter defining the width of the ellipse.

When all the links of the wireless network are considered, the change in the propagation field of the monitored area is:

$$\mathbf{y} = \mathbf{W}\mathbf{x} + \mathbf{n}, \tag{7}$$

in which $\mathbf{y}$ and $\mathbf{n}$ are $M \times 1$ vectors representing the weighted RSS change and noise of the M wireless links, and $\mathbf{x}$ is the $N \times 1$ change in the propagation field to be estimated, where each element x_j represents change due to the presence of a person in voxel j. The linear model for the change in the propagation field is based on the correlated shadowing models in [25], [26] and on the work in [1].

Since estimating the image vector $\mathbf{x}$ from the links' measurements $\mathbf{y}$ is an ill-posed inverse problem, regularization is required. In this work, we use a regularized least-squares approach [25], [27]:

$$\hat{\mathbf{x}} = \Pi \mathbf{y}, \tag{8}$$

where:

$$\Pi = (\mathbf{W}^T\mathbf{W} + \mathbf{C}_x^{-1}\sigma_N^2)^{-1}\mathbf{W}^T, \tag{9}$$

in which σ_N^2 is the regularization parameter. The *a priori* covariance matrix $\mathbf{C}_x$ is calculated by using an exponential spatial decay:

$$[\mathbf{C}_x]_{ji} = \sigma_x^2 e^{-\|\mathbf{v}_j - \mathbf{v}_i\|/\delta_c}, \tag{10}$$

where σ_x^2 is the variance of voxel measurements, and δ_c is the voxels' correlation distance. The linear transformation Π is computed only once before real-time operation. The calculation of $\hat{\mathbf{x}}$ in (8) requires MN operations and can be performed in real-time.

2.4 Image Denoising

An RTI image representing the situation in which n targets are located in different regions of the monitored area should ideally show n *blobs*, *i.e.*, regions in which the voxels have an intensity much higher than the intensity of the surrounding voxels. However, due to noise in the RSS measurements, modeling inaccuracies, and the simultaneous

presence of multiple individuals and obstructions, the RTI images are often noisy, showing multiple spurious blobs of small size which do not correspond to actual targets. For this reasons, a Gaussian filter is applied on the RTI image. This operation has the effect of reducing the image's high spatial frequency components, filtering out small spurious blobs while simultaneously keeping those larger blobs corresponding to actual targets [28].

The filtering is obtained by convolving the RTI image with an isotropic Gaussian kernel:

$$\mathbf{G}(x,y) = \frac{1}{2\pi\sigma_G^2} e^{-\frac{x^2+y^2}{2\sigma_G^2}}, \tag{11}$$

where $\sigma_G = 1$ m is the standard deviation of the Gaussian kernel which indicates how much the image is filtered (or *blurred*). Since the estimated RTI image $\hat{\mathbf{x}}$ is stored as a set of discrete voxels, we need to produce a discrete approximation of the Gaussian kernel $\mathbf{G}$ to be able to perform the convolution. Thus, the kernel is truncated after $\lfloor r_G/p + 0.5 \rfloor$ voxels, where $r_G = 0.75$ m is the radius of the kernel. The low-pass filtered RTI image $\hat{\mathbf{x}}_G$ is then calculated as:

$$\hat{\mathbf{x}}_G = \hat{\mathbf{x}} * \mathbf{G}, \tag{12}$$

where $*$ represents the convolution operator. As a result, each voxel of the filtered RTI image is a weighted average of the voxel's neighborhood, with the central pixels weighted more than the peripheral ones. The values of σ_G and r_G are chosen so as to fit the size of the blobs corresponding to the targets.

3 Multiple Target Tracking

The methods described in the previous section form radio tomographic images in real-time, *i.e.*, pictures of the change in the propagation field of the monitored area caused by the presence and movements of the people found in it. These images can be considered as *frames* of a video showing the movements of multiple targets. In this section, we describe the methods used to process the RTI images and track multiple targets in real-time. Our objective is to correctly detect the entrance and exit of the targets and estimate their position $\mathbf{v}_t$ by assigning to each of them a voxel of the estimated RTI image.

3.1 Thresholding

After denoising the image, an additional filter is required in order to reduce the size of the set of voxels that go through the clustering process (see Section 3.2) and to preserve only those in the regions occupied by the targets. To this purpose, a dynamic threshold T_t is set as follows.

During the calibration period, *i.e.*, when the monitored area is empty, we calculate the average maximum intensity of the formed RTI images, $\bar{I}_e$. When the monitored area is empty, the threshold T_t is set to $2\bar{I}_e$ to filter the voxels having very low intensity. On the other hand, when targets are being tracked, we derive the minimum intensity, I_{min}, of the targets t in the set of estimated targets $\mathcal{T} = \{t_1, \ldots, t_{|\mathcal{T}|}\}$ as:

$$I_{min} = \min_{t \in \mathcal{T}} [\hat{\mathbf{x}}_G]_t. \tag{13}$$

We then low-pass filter I_{min}:

$$I_f(k) = \alpha_f I_f(k-1) + (1-\alpha_f) I_{min}(k), \tag{14}$$

where $\alpha_f = 0.9$. Thus, the threshold T_t is set as follows:

$$T_t(k) = \begin{cases} \beta I_f(k) & \text{if } |T(k)| > 0 \\ 2\bar{I}_e & \text{otherwise,} \end{cases} \tag{15}$$

where $|T(k)|$ is the number of targets found in the monitored area at time instant k and $\beta < 1$. In the experiments, we could not observe any significant performance change for values of β in the range $[0.75, 0.9]$. The on-line updating of T_t ensures that the voxels surrounding the tracks are not filtered and will go through the clustering process.

A vector mask $\mathbf{M_t}$ of size $N \times 1$ is created as follows:

$$[\mathbf{M_t}]_n = \begin{cases} 1 & \text{if } [\hat{\mathbf{x}}_G]_n > T_t \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

The denoised and filtered RTI image $\hat{\mathbf{x}}_f$ is finally calculated as the element-wise product of the denoised image $\hat{\mathbf{x}}_G$ and $\mathbf{M_t}$:

$$\hat{\mathbf{x}}_f = \hat{\mathbf{x}}_G \wedge \mathbf{M_t}. \tag{17}$$

3.2 Clustering

In the clustering phase, voxels are assigned to each blob found in the image. Since we do not make any a priori assumption on the number of targets to be tracked (new targets can enter the monitored area at any time as tracked targets can leave it, and spurious blobs can also appear), clustering algorithms, such as *k-means* [29], for which the number of clusters must be known a priori can not be applied. For this reason, we use an hierarchical agglomerative clustering (HAC) algorithm [30].

We define $\mathcal{V}$ as the set of voxels that are not filtered in the thresholding phase (see Section 3.1), *i.e.*, $\mathcal{V} = \{j : [\hat{\mathbf{x}}_f]_j > T_t\}$. Each voxel $j \in \mathcal{V}$ has coordinates $\mathbf{v}_j = [x_j, y_j]$ in the XY plane. In the HAC algorithm, each voxel is initially considered as an independent cluster. At each iteration, the two closest clusters are merged. The distance between two clusters, $S_a \subset \mathcal{V}$ and $S_b \subset \mathcal{V}$, is measured with the average linkage distance $\bar{d}$, *i.e.*, the average of the Euclidean distances between all the voxels assigned to the two clusters:

$$\bar{d}(S_a, S_b) = \frac{\sum_{j_a \in S_a} \sum_{j_b \in S_b} \|\mathbf{v}_{j_a} - \mathbf{v}_{j_b}\|}{|S_a||S_b|}. \tag{18}$$

The iterations terminate when the minimum of the average linkage distances among the clusters is larger than a threshold T_c, which determines the average size of the formed clusters, and ultimately their number, (*i.e.*, several small clusters for low values of T_c, few larger clusters for high values of T_c). The value of T_c used in the experiments (see Table 2) represents the average diameter of a blob corresponding to an actual person, and was derived experimentally.

We normalize the intensity of the image in the range $[0, 1]$, so that the normalized intensity $\tilde{I}_j$ of each voxel $j \in \mathcal{V}$:

$$\tilde{I}_j = \frac{[\hat{\mathbf{x}}_f]_j}{\max_{j \in \mathcal{V}} [\hat{\mathbf{x}}_f]_j}. \tag{19}$$

For each formed cluster S_i, the voxel $h_i \in S_i$ having the maximum normalized intensity is selected as the cluster head:

$$h_i = \arg\max_{j \in S_i} \tilde{I}_j. \quad (20)$$

We define the set of *original cluster heads* $\mathcal{H}$ as the set of voxels h_i for all i.

3.3 Cluster Heads Selection

Due to the 3D shape of the blobs found in the RTI image (typically round in open environments, having more distorted shapes in obstructed areas and when targets trajectories intersect), the HAC algorithm can form several cluster heads. In this case, we want to decrease the number of elements in $\mathcal{H}$ in order to reduce the complexity of the observations-targets association problem (see Section 3.4.2), and simultaneously keep, for every occupied region, only the cluster heads with the highest intensities, which are more likely to correspond to the real targets found in the monitored area. As a result of the selection process, a new set $\mathcal{H}_I \subseteq \mathcal{H}$ of cluster heads having higher intensities is formed.

In each indoor environment where the sensors are deployed, we define $\mathcal{R}_e \subset \mathcal{V}$ as the set of voxels included in the entrance/exit region, *i.e.*, the region the targets must go through in order to enter and exit the monitored area. This region can be limited to a specific part of the monitored area (as in the apartment described in Section 4.3), or can cover the entire region along the perimeter of the monitored area (as in the open and office environments in Section 4.2 and 4.4). All the cluster heads that are located within $\mathcal{R}_e$ are included in $\mathcal{H}_I$, regardless of their intensity.

The remaining cluster heads are selected based on their proximity to the targets being tracked and their intensity. As first step, gating is applied on the cluster heads in $\mathcal{H}$. At time k, for each cluster head h at position $\mathbf{v}_h$, we define χ_h as:

$$\chi_h = \{t : \|\mathbf{v}_h - \hat{\mathbf{v}}_t\| < r_t\}, \quad (21)$$

where $\hat{\mathbf{v}}_t$ is the position of the cluster head associated to target t at time $k-1$. The parameter r_t represents the radius of the gating area centered at $\hat{\mathbf{v}}_t$. Initially, $r_t = r$ (see Table 2). The value of r_t is modified whenever the trajectory of target t intersects the trajectory of another track (see Section 3.6). The gating area has to accommodate for the motion variance of the targets, indicating how fast the targets can move, and the typical noise of RTI images, (*i.e.*, spurious and disappearing blobs, and blobs merging and splitting in case of intersecting trajectories).

Based on the results of the gating process, a cluster head h is included in $\mathcal{H}_I$ if *a*) $\|\chi_h\| \geq 1$, *i.e.*, if there is at least one target t whose gating area includes h, and *b*) the normalized intensity of h is larger than a threshold T_h, defined as:

$$T_h = \rho \min_{t \in \chi_h} \tilde{I}_t, \quad (22)$$

where $\rho = 0.8$.

3.4 Target Tracking

3.4.1 Tracks Confirmation and Deletion

The selected cluster heads in $\mathcal{H}_I$ are considered for updating the existing tracks and for potentially initiating new tracks. In machine vision, track confirmation and deletion is usually determined by rules [31]. In our case, the rules have to deal with RTI images that show new blobs when people enter the monitored area and stop showing blobs when people exit the monitored area. RTI images can show spurious blobs not corresponding to people, while blobs corresponding to real targets can temporarily disappear, as it is shown in the videos in [15]. Due to this, a trade-off exists between the reactivity of the system to the entrance and exit of targets and the sensitivity to the noise in the images.

At each frame, the data association methods presented in Section 3.4.2 assign some of the cluster heads in $\mathcal{H}_I$ to the already existing tracks. Of the cluster heads which are not assigned, only those that are located in $\mathcal{R}_e$ are considered as new *candidate tracks*, *i.e.*, tracks potentially corresponding to new targets, whereas the ones located outside of $\mathcal{R}_e$ are considered as noise and are discarded. A candidate track becomes a *confirmed track* only if it has been assigned a cluster head (see Section 3.4.2) at least n_{app} times in the last m frames ($n_{app} \leq m$). The value of n_{app} makes the system more or less reactive to the entrance of new targets. By using this rule, the system confirms the entrance of a new target only after having multiple confirmations of its presence. This introduces a small latency between when a new target enters the monitored area and the moment the system acknowledges it, as shown in Fig. 3. However, the rule makes the system more robust to the appearances of spurious blobs in $\mathcal{R}_e$. Since the DFL system used in the experiments produces approximately 10 RTI images (or frames) per second, we set $m = 10$ so to consider the appearances of the tracks in a one second time interval before confirming their presence.

On the other hand, a track (whether confirmed or candidate) is deleted after it has not been assigned a cluster head in the last n_{del} consecutive frames. Also in this case, the DFL system deletes the existence of the tracks with a small delay compared to reality. However, the rule prevents the system from incorrectly deleting tracks that have not been associated a cluster head for a few consecutive frames due to noise in the RTI image.

3.4.2 Target Tracking

The problem of tracking multiple targets can be formulated as a data assignment problem (DAP), in which at each frame a set of new observations has to be assigned to the set of existing targets, as shown in Fig. 1. In our DFL system, at each new formed RTI image the set of selected cluster heads $\mathcal{H}_I = \{h_1, \ldots, h_{|\mathcal{H}_I|}\}$ has to be assigned to the set of estimated targets $\mathcal{T} = \{t_1, \ldots, t_{|\mathcal{T}|}\}$. In general, the solution of the DAP consists in finding the optimal permutation ϕ of the set $\mathcal{H}_I$, where the permutation matrix Θ is defined as:

$$\Theta_{h,t} = \begin{cases} 1 & \text{if } t = \pi(h) \\ 0 & \text{otherwise.} \end{cases} \quad (23)$$

In our case, based on the outcome of the gating process (see Section 3.3), an association matrix Ω is defined as follows:

$$[\Omega]_{h,t} = \begin{cases} \|\mathbf{v}_h - \hat{\mathbf{v}}_t\| & \text{if } t \in \chi_h \\ \infty & \text{otherwise.} \end{cases} \quad (24)$$

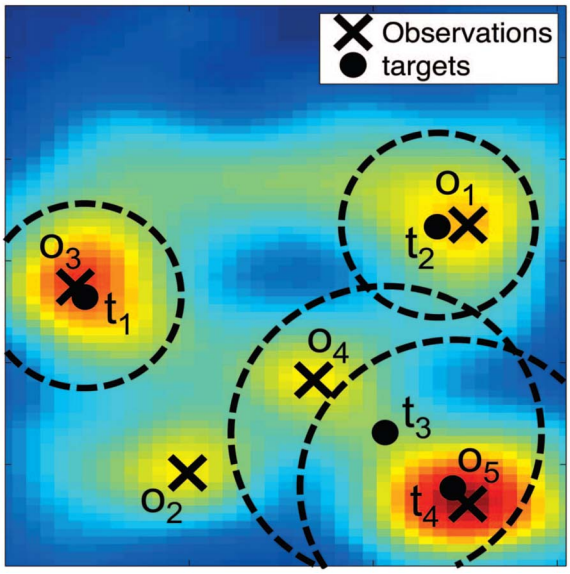


Fig. 1. DAP in RTI. The dashed circles represent the gating areas centered at the tracks. In this case, the observation-tracks assignments are: $o_1 - t_2$, $o_3 - t_1$, $o_4 - t_3$ and $o_5 - t_4$. Observation o_2 is not assigned and is considered for starting a new track. The radius of the gating areas centered at t_3 and t_4 is larger as these two targets are intersecting.

The elements set to ∞ represent unfeasible observation-target assignments. When Ω is interpreted as a cost matrix, the DAP becomes the problem of selecting observation-target assignments that minimize the total cost.

3.4.3 Prior Work

The DAP in multiple target tracking has received significant attention from the research community, and several methods, including particle filters (PF, [32]), probabilistic and joint probabilistic data association methods (PDA and JPDA, [33]) and multiple hypothesis tracking (MHT, [34]), have been proposed and analyzed [35]. However, these methods present some limitations that make them not suitable to the requirements of our DFL system, which are *a*) no a priori assumptions on the number of targets and *b*) real-timeliness.

The PDA and JPDA approaches assume that the number of targets is known and constant. Moreover, the PF and MHT approaches are methods that struggle to meet strict real-time requirements: the PF needs a high number of particles to obtain accurate tracking, at the expense of a high computational complexity and long processing time [14], [18]. Moreover, whenever the number of targets to be tracked increases, the PF needs a higher number of particles to maintain the same accuracy, making the processing time even longer. In MHT, each feasible target-observation association is considered as an hypothesis with a specific probability. At each new RTI image, each hypothesis is expanded into a set of new hypothesis having specific probabilities, so that a tree of hypothesis is incrementally generated. Each hypothesis is expressed as a permutation matrix, and the method keeps a probability distribution over the space of all permutation matrices. After several frames have been analyzed, the tracks having the highest probabilities are selected. However, the number of hypothesis grows exponentially with the number of targets in $\mathcal{T}$, making this method computationally intensive [36]. Furthermore, the MHT approach is a *batch* method, which postpones the DAP solution until more clear information is available. On the contrary, we aim at using a recursive method that estimates the targets positions at time k based on the observations available at time k and the estimates of the targets positions at time $k-1$.

3.4.4 Nearest Neighbor Methods

In this work, two computationally efficient versions of the nearest neighbor (NN) approach, namely the global nearest neighbor (GNN) and the greedy (or suboptimal) nearest neighbor (SNN), are tailored for the characteristics of RTI. Despite its simplicity, the NN approach has demonstrated high tracking performance in scenarios characterized by noisy measurements, such as radar and sonar applications [37], [38]. In GNN and SNN, the tracks are updated at each frame. A track can be associated to only one observation, and in turn one observation can be associated only to one track.

The GNN method applies the Hungarian algorithm [39], [40], which is capable of finding the optimal solution in a polynomial time $O(n^3)$, in which $n = \min(|\mathcal{T}|, |\mathcal{H}_I|)$. The SNN method applies a greedy approach to the selection of the observation-target assignments, which in $O(n)$ time is not guaranteed to find the optimal solution but requires fewer computations than the GNN method (especially when the number of feasible assignments is large). For the greedy SNN method, the upper bound of the total cost associated to the selection of the observation-target assignments is equal to twice the optimal cost guaranteed by the GNN method. The observations that are not assigned to any target are considered for starting new tracks (see Section 3.4.1). On the other hand, the tracks that are not assigned an observation are still predicted by using the Kalman filter (KF).

3.5 Kalman Filter Tracking

Whenever a candidate track t is confirmed, a track-specific KF [41], [42] is initialized and recursively applied to track its movements. The system runs the KFs in parallel. Each KF estimates the new state, *i.e.*, position, of a track by taking into consideration its previous state and the new observation, *i.e.*, cluster head, associated to it (see Section 3.4.2). We assume that the targets to be tracked move as a Brownian process and that the measurement noise is Gaussian. The KF smoothes the trajectories of the targets and reduces their sudden changes of direction. At this purpose, it is particularly useful in the case of noisy RTI images.

3.6 Handling Intersecting Trajectories

In this work, we tackle the problem of tracking targets having intersecting trajectories. In RTI, this situation manifests itself in two (or more) blobs slowly merging into a single one and then splitting again after some frames. In cluttered indoor environments, when two targets approach each other, the formed RTI images become very noisy due to the unpredictable overlap of the multiple propagation paths modified by each target. For this reason, whenever

TABLE 1
Image Reconstruction Parameters

Parameter	Value	Description
p	0.1524	Pixel width [m]
λ	0.02	Ellipse excess path length[m]
σ_x	0.2236	voxels standard deviation [dB]
σ_N	1	noise standard deviation [dB]
δ_c	3	correlation coefficient

TABLE 2
Multiple Target Tracking Parameters (Default Values)

Parameter	Value	Description
β	0.80	Parameter used to set T_t in (15)
T_e	0.01	Empty area intensity threshold
T_c	1.25	Voxels clustering threshold [m]
T_i	2	Intersecting trajectories threshold [m]
r	2	Radius of the gating area [m]
r_G	0.75	Radius of the Gaussian kernel [m]
σ_G	1	Std. deviation of the Gaussian kernel [m]

the distance between a target t and any other target drops below T_i, r_t in (21) is doubled. With this adjustment, the motion variance of the targets is increased. Since in intersecting situations the merging-splitting blobs can disappear for several consecutive frames, increasing the radius of the gating area avoids losing track of targets.

4 Experimental Setup

In this section, we describe the hardware and communication protocol used in the experiments and the environments in which we carried out the tests. The values of the image reconstruction parameters and of the multiple target tracking methods are listed in Table 1 and 2, respectively.

4.1 Hardware and Communication Protocol

The sensors used in all the experiments are TI CC2531 USB dongle nodes, transmitting at their maximum nominal power, *i.e.* 4.5 dBm [43]. For multi-channel communication, the sensors run a multi-channel token passing protocol, *multi-Spin* [44]. In it, the sensors transmit in TDMA fashion based on their ID number. Each transmitted packet contains the ID number of the transmitting node and the most recent RSS measurements of the packets received from the other sensors. At the end of each communication cycle, the sensors switch synchronously to the next frequency channel found in a list pre-defined by the user. On average, the time interval between two consecutive transmissions is 2.9 ms. A sink node that overhears all the packets transmitted by the nodes stores the RSS measurements for processing.

For a network composed of approximately 30 nodes (like the ones used in our experiments), the time required to complete a communication cycle on a single frequency channel is approximately 100 ms. In order to be able to sample the RSS on each link and frequency channel two times per second, our system can use up to five frequency channels. In our tests, the initial calibration of the system, performed when the monitored area is empty, lasts for approximately 15 seconds *i.e.*, the time required to collect 30 RSS measurements on each link for each used frequency channel.

The CC2531 nodes transmit in the 2.4 GHz ISM band in one of 16 selectable frequency channels, which are 5 MHz apart, as specified by the IEEE 802.15.4 standard [45]. The carrier frequency (in MHz) of channel c is:

$$f_c = 2405 + 5 \cdot (c - 11), \quad c \in [11, 26]. \tag{25}$$

In the following, we describe the three different indoor environments where the experiments were carried out. In all the deployment environments, multiple 802.11 b/g networks create interference in the 2.4 GHz band, increasing the floor noise level of the RSS measurements [46]. People being tracked in the tests are walking at a constant speed. We do not consider the situation in which people stop *e.g.* to sit down at desks or talk to each other.

4.2 Open Environment

In an open indoor environment, *i.e.* where no obstructions or objects are present, 30 sensors are deployed along the perimeter of a $70m^2$ area, as shown in Fig. 2(a). The sensors are placed on podiums at a height of one meter from the floor. They transmit on 5 different channels, *i.e.* $c \in \{11, 15, 18, 22, 26\}$. We select these channels since they are approximately equidistant in the 80 MHz frequency range. During the tests, two people are instructed to walk at constant speed along a pre-defined rectangular path. In the first test, one person enters the monitored area and walks along the path, followed by the second person after few seconds. The two people walk along the path in the same direction, *i.e.*, one behind the other. In the second test, the two people enter the monitored area few seconds one after the other, but this time the second person walks along the path in the opposite direction to the first person, so that the trajectories of the two people intersect. In this environment, we assume that people could enter and exit the monitored area at any point along the perimeter.

4.3 Apartment

We deploy 33 sensors in a one bedroom, $58m^2$ (7x8.25 m) apartment shown in Fig. 2(b) and (d). The sensors are attached to the walls and furniture, at approximately one meter from the floor. They transmit on 4 different channels, *i.e.* $c \in \{15, 20, 25, 26\}$. The apartment is located in a residential building in an area where more than ten Wi-Fi networks belonging to residents of the neighbouring apartments overlap. To partially mitigate the effect of Wi-Fi interference on the RSS measurements [46], we select four interference-free channels. Moreover, to further reduce the noise of the RSS measurements [6], the nodes are attached so to keep the antenna at least 5 centimeters away from the wall. In the tests, two people walk along pre-defined paths. The trajectories intersect multiple times. We assume that the apartment has two specific entrance regions, located at the main door and at the sliding glass-door separating the balcony from the living room.

4.4 Office Environment

In a typical office environment, shown in Fig. 2(c) and (e), containing several metallic objects, such as desks, chairs, computer towers and monitors, we deploy 32 sensors to

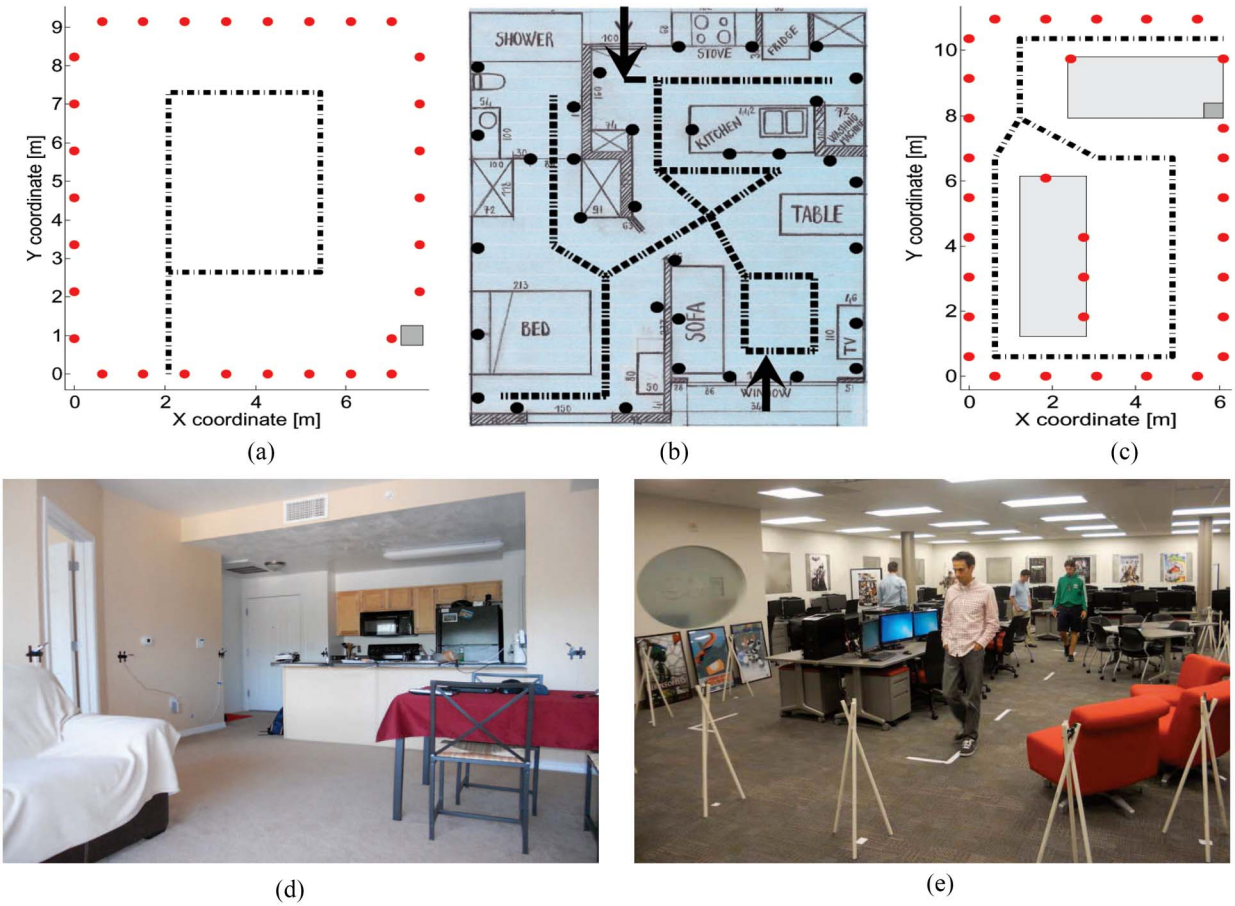


Fig. 2. In (a)–(c), the setups in the three indoor environments used for the experiments. The circles represent the sensors, while the dashed lines represent the paths followed by the people moving in the monitored area. The light gray rectangles in (c) are the rows of desks found in the office environment. The dark gray squares in (a) and (c) represent a concrete pillar. In (d), an image of the apartment. In (e), an image of the office environment.

cover a $67m^2$ area (27 sensors along the perimeter, 5 in internal positions). The nodes along the perimeter are placed on podiums at a height of one meter from the floor, while the other five are positioned on desks. They transmit on 5 different channels, *i.e.* $c \in \{11, 15, 18, 22, 26\}$. Also in this case, we select these channels since they are approximately equidistant in the 80 MHz frequency range. In this environment, we conduct several tests with two, three, and four people simultaneously walking at constant speed along a pre-defined path. At first, people enter the area one after the other and walk along the path in the same direction, exiting the area again one after the other. Later, people enter the area again one after the other, but this time they walk in opposite direction, so that their trajectories intersect multiple times.

5 Evaluation Metrics

5.1 Cardinality Error

The cardinality error ϵ_c is calculated as the fraction of frames during the test in which the number of people moving in the monitored area and the estimated number of confirmed tracks differ.

5.2 Tracking Accuracy

To evaluate the tracking accuracy, we use the optimal mass transfer (OMAT) metric, introduced in [47] and defined as:

$$\epsilon_O(\mathcal{T}, \mathcal{Z}) = \frac{1}{|\mathcal{T}|} \min_{\upsilon \in \Upsilon} \left(\sum_{i=1}^{|\mathcal{T}|} \sum_{j=1}^{|\mathcal{Z}|} \upsilon_{i,j} \|\mathbf{t}_i - \mathbf{z}_j\|^q \right)^{\frac{1}{q}}, \quad (26)$$

in which Υ is the set of all the possible permutations υ between the set of estimated targets $\mathcal{T}$ and the set of real targets $\mathcal{Z}$, and q is the order of the metric (we set $q = 2$). Thus, the OMAT error ϵ_O is equal to the root mean square error (RMSE) of the best possible association between estimated and real targets.

5.3 OSPA Metric

Differently than the OMAT metric defined in (26), the optimal subpattern assignment (OSPA) metric [48] includes an additional term, *i.e.*, a constant g measured in meters, which penalizes the cardinality error. When $\mathcal{T}$ is the set of estimated targets, $\mathcal{Z}$ the set of real targets, and $|\mathcal{T}| \leq |\mathcal{Z}|$, the OSPA metric $\epsilon_p^{(g)}(\mathcal{T}, \mathcal{Z})$ can be calculated as:

$$\epsilon_p^{(g)}(\mathcal{T}, \mathcal{Z}) = \left(\frac{1}{|\mathcal{Z}|} \min_{\upsilon \in \Upsilon} \sum_{i=1}^{|\mathcal{T}|} d^{(g)}\left(t_i, z_{\upsilon(i)}\right)^q + g^q \left(|\mathcal{Z}| - |\mathcal{T}|\right) \right)^{\frac{1}{q}}, \quad (27)$$

where $d^{(g)}(t, z) = \min\{d(t, z), g\}$ and d is the Euclidean distance between the estimated and real position of a target.

TABLE 3
Processing Time for the GNN, SNN, and MHT Methods

			GNN method		SNN method		MHT method	
Targets	Environment	Intersections (#)	$\max[T_p]$ [ms]	$E[T_p]$ [ms]	$\max[T_p]$ [ms]	$E[T_p]$ [ms]	$\max[T_p]$ [ms]	$E[T_p]$ [ms]
2	open	NO	15.4	7.3	14.5	7.0	16.8	7.3
2	open	YES (9)	13.5	7.2	12.3	7.0	622.5	134.6
2	apartment	YES (4)	19.4	6.4	18.2	5.9	712.9	121.1
2	apartment	YES (5)	19.3	6.5	11.8	5.8	537.7	109.2
2	office	NO	18.6	6.8	12.2	6.2	12.9	5.9
2	office	YES (5)	25.3	7.4	14.2	6.9	419.4	53.1
3	office	NO	26.1	9.3	21.8	8.8	231.6	28.0
3	office	YES (6)	30.3	10.4	19.2	9.2	1844.2	194.7
4	office	NO	35.6	11.6	34.9	11.4	321.3	34.6
4	office	YES (9)	43.4	13.3	33.5	11.5	4632.1	244.7

We set $q = 2$. When $|\mathcal{T}| > |\mathcal{Z}|$, the OSPA metric is calculated as $\epsilon_p^{(g)}(\mathcal{Z}, \mathcal{T})$, *i.e.*, as in (27) but inverting $\mathcal{T}$ and $\mathcal{Z}$ in it.

5.4 Tracking Consistency

Since in some of the tests the trajectories of the targets intersect, we define a metric to evaluate the capability of the DFL system to avoid losing track of targets during and after their intersection. To this purpose, we use the 95%-percentile Q_{95} of the OMAT errors of the estimated tracks. A low Q_{95} indicates that even with intersecting trajectories and noisy RTI images the system does not lose track of the targets.

6 Experimental Results

6.1 Processing Time

The maximum, $\max[T_p]$, and average, $E[T_p]$, processing time of the multiple target tracking algorithms are measured on a laptop having a 2.50 GHz Intel® Core™ i5-2450P processor and 8.0 GB of RAM memory. Table 3 lists $\max[T_p]$ and $E[T_p]$ for the two considered nearest neighbor approaches, GNN and SNN. For comparison, the table includes also $\max[T_p]$ and $E[T_p]$ for the multiple hypothesis tracking (MHT) method [34], [36]. Both the methods are well below the limit set by the length of an entire TDMA communication cycle on one channel (see Section 4.1), which is approximately 100 ms for a network composed of about 30 nodes. The results show that the SNN method is faster than the GNN method. On the other hand, the MHT method does not fulfill the real-time requirement, especially when the trajectories of the targets intersect. In this situation, the MHT method starts creating a tree of hypothesis for the possible paths followed by the targets based on the available observations, postponing the decision till following RTI images which are more easy to be interpreted. Moreover, the processing time of the MHT method increases considerably with the number of targets to be tracked due to the increased size of the hypothesis tree. Both $\max[T_p]$ and $E[T_p]$ increase with the number of targets to be tracked also for the GNN and SNN methods, due to the higher number of voxels going through the clustering process and the higher number of observations that have to be associated to the existing tracks. However, both methods cope much better than MHT with the increased complexity of the tracking problem.

6.2 Cardinality Error

The cardinality error (ϵ_c) and the accuracy (ϵ_O) and consistency (Q_{95}) of the tracking obtained by applying the methods presented in this paper are listed in Table 4. The cardinality error measured in all the experiments is limited only to a few frames before the actual entrance of the targets in the monitored area and a few frames after their actual exit, as shown in Fig. 3. This depends on the confirmation and deletion rules discussed in Section 3.4.1.

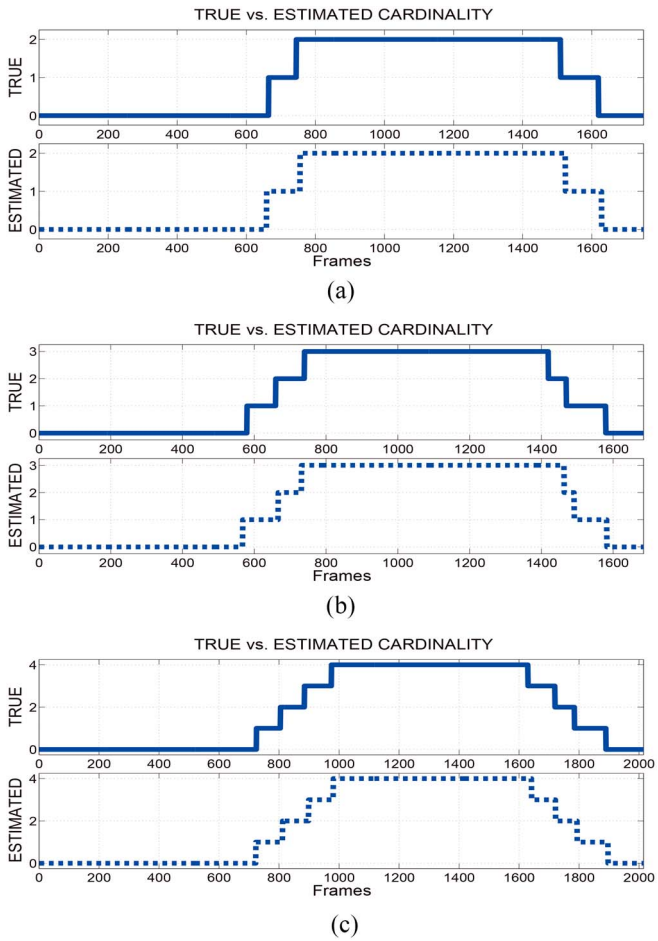


Fig. 3. In (a)–(c), the comparison between the true numer of targets in the monitored area and the number of targets estimated by the DFL system during the tests carried out in the office environment with targets having separated trajectories. (a) 2 targets - open environment. (b) 3 targets - office environment. (c) 4 targets - office environment.

TABLE 4
ϵ_C, ϵ_O and Q_{95} for the GNN and SNN Methods

				GNN method		SNN method	
Targets	Environment	Intersections (#)	ϵ_C	ϵ_O [m]	Q_{95} [m]	ϵ_O [m]	Q_{95} [m]
2	open	NO	0.001	0.32	0.71	0.33	0.72
2	open	YES (9)	0.001	0.32	0.74	0.33	0.77
2	apartment	YES (4)	0.001	0.30	0.71	0.30	0.75
2	apartment	YES (5)	0.002	0.26	0.68	0.27	0.71
2	office	NO	0.029	0.38	0.75	0.38	0.76
2	office	YES (5)	0.023	0.44	0.96	0.45	1.01
3	office	NO	0.016	0.37	0.66	0.37	0.67
3	office	YES (6)	0.032	0.44	1.05	0.46	1.07
4	office	NO	0.027	0.44	0.94	0.45	0.96
4	office	YES (9)	0.029	0.54	1.16	0.55	1.26

During the experiments we observed that the delay in detecting the entrance of a target in the monitored area due to the confirmation rule can sometimes be compensated by the use of multiple frequency channels. Deep-fade links show variation of the RSS even when the person is at some position far away from the link line. When a new target approaches the entrance region, the RTI images start showing a new blob. If this blob receives multiple successive confirmations, the presence of a new target is detected with a small anticipation with respect to its actual entrance in the monitored area. This effect can be observed in Fig. 3(a) and (b) at the entrance of the first target.

6.3 Tracking Accuracy and Consistency

For what concerns the tracking accuracy, the largest average OMAT error, $\epsilon_O = 0.55$ m, is measured when four targets with intersecting trajectories are tracked in the office environment, where multipath propagation of the radio signals is predominant. On the other hand, the largest OMAT errors measured in the open environment are $\epsilon_O = 0.33$ m and $\epsilon_O = 0.30$ m, respectively.

In the apartment environment, the difference in the ϵ_O and Q_{95} values measured in the two tests carried out is due to the paths covered by the two people, which were more separated in the second experiment than in the first one. In fact, the spots where the two tracks intersect have a huge impact on the consistency of the tracking accuracy. Due to the particular positioning of the sensors in an environment with walls, various objects, and furniture, certain areas are covered by a higher number of anti-fade links (which favor the localization effort), whereas in some others the number of deep-fade links (less reliable for localization) is larger [6]. For this reason, the amount of noise found in the RTI images formed when the two trajectories intersect is different in each test, and this explains the slower convergence of the estimated tracks to the real ones after the intersections. Despite this, the average OMAT error measured during the two tests is approximately 0.30 m, both with the GNN and SNN methods.

In the office environment, the average OMAT error measured with two targets is slightly higher than in the other two environments, in both the situations of separated ($\epsilon_O = 0.38$ m) and intersecting ($\epsilon_O = 0.45$ m) trajectories. The difference is due to the fact that this environment is the most challenging for DFL due to the presence of multiple desks, chairs, computer towers and monitors. Nevertheless, the largest ϵ_O and Q_{95} measured with 4 targets are 0.45 m and 0.96 m with separated trajectories and 0.55 m and 1.26 m with intersecting trajectories. These results show that our DFL system is capable of accurately and consistently tracking up to 4 targets in a challenging indoor environment where multipath propagation is predominant. Fig. 4 shows the real and estimated trajectories of the test conducted in the office environment with three targets having intersecting trajectories. The cumulative distribution functions (CDFs) of ϵ_O obtained with the GNN and SNN methods in the tests carried out in the office environment with intersecting trajectories are shown in Fig. 5. Since the GNN method finds the optimal solution of the DAP at each frame, the improvement in Q_{95} measured with this method becomes more consistent with more noisy images, *i.e.* when four intersecting targets are tracked.

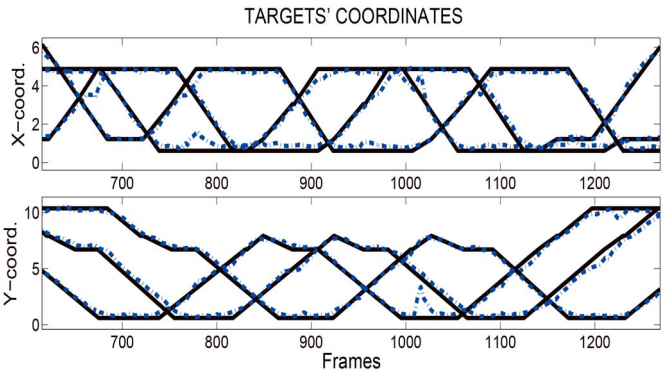


Fig. 4. Real (solid lines) and estimated (dashed lines) coordinates of the targets during a test carried out in the office environment. Despite multiple intersections, $\epsilon_O = 0.44$ m and $Q_{95} = 0.99$ m.

Table 5 lists the average OSPA errors for the tests carried out in the office environment for different values of the cardinality penalty g. Despite the fact that no assumption is made on the number of targets found in the monitored area, due to the very small fraction of frames in which $|\mathcal{T}| \neq |\mathcal{Z}|$, the average OSPA errors increase by a very small amount even when the cardinality penalty is set to a very high value ($g = 5$).

6.4 Sensitivity Analysis

Fig. 6 shows ϵ_O, Q_{95}, $\max[T_p]$, and $E[T_p]$ measured with the GNN method for different values of the clustering threshold T_c (see Section 3.2) for the test carried out in the office environment with 4 targets having intersecting trajectories. T_c determines the average size of the formed clusters, and ultimately their number. The results show that

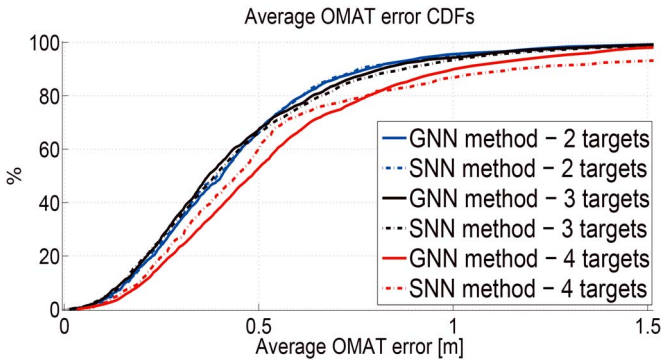


Fig. 5. Cumulative distribution functions of the average OMAT error obtained with the GNN (solid lines) and SNN (dashed lines) methods in the tests carried out in the office environment with targets having intersecting trajectories.

TABLE 5
Average OSPA Errors for the Tests in the Office Environment

		Average OSPA error [m]		
Targets	Intersections	$g = 1$	$g = 2.5$	$g = 5$
2	no	0.41	0.45	0.51
2	yes	0.46	0.51	0.56
3	no	0.39	0.41	0.43
3	yes	0.48	0.59	0.75
4	no	0.47	0.52	0.59
4	yes	0.57	0.69	0.83

the tracking accuracy of the system is consistent for different values of T_c, with ϵ_O taking values from 0.54 m to 0.74 m. Related to the processing time, when T_c takes small values the system is not capable of processing the RTI images in real-time. This is due to the large number of formed clusters, which increases the time required for solving the data association problem (see Section 3.4.2). Both $\max[T_p]$ and $E[T_p]$ decrease when T_c takes larger values. Similar results were obtained while considering all the other tests.

6.5 Accuracy vs. Nodes' Density

One of the key aspects for the use of RTI systems in real-life scenarios is the number of sensors required to enable accurate localization and tracking of multiple targets. In this section, we present results of simulations performed by progressively removing sensors from the system, *i.e.*, progressively reducing the nodes' density. We consider the most challenging environment, *i.e.*, the office environment, and the tests with three and four targets having intersecting trajectories. In the simulations we use the same data collected during the experiments described in Section 4. For each number of removed nodes, we perform 15 simulations, each time removing a different subset of sensors. The nodes' density is reduced from the original 0.48 nodes/m^2 (*i.e.*, 32 nodes in $67m^2$) to 0.30 nodes/m^2 (*i.e.*, 20 nodes in $67m^2$). The results are shown in Fig. 7.

The results indicate an approximately linear relationship between the average ϵ_O (calculated over 15 different permutations of removed nodes) and the nodes's density. In both the tests, ϵ_O remains constantly below 1 m. Moreover, the variance in ϵ_O for the same number of removed nodes shows that some nodes provide RSS measurements that enhance the tracking accuracy, while other nodes' measurements decrease it. This fact indicates that an optimal positioning of the nodes in the deployment environment could potentially achieve the same (or better) tracking accuracy with a considerably lower number of sensors. We observed similar results in the simulations performed for the other experiments and deployment environments.

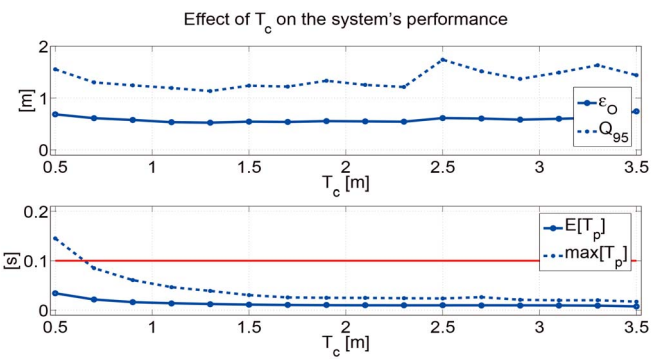


Fig. 6. ϵ_O, Q_{95}, $\max[T_p]$, and $E[T_p]$ measured with the GNN method for different values of T_c for the test carried out in the office environment with 4 targets having intersecting trajectories. The solid line in the image at the bottom represents the real-time limit for the processing time.

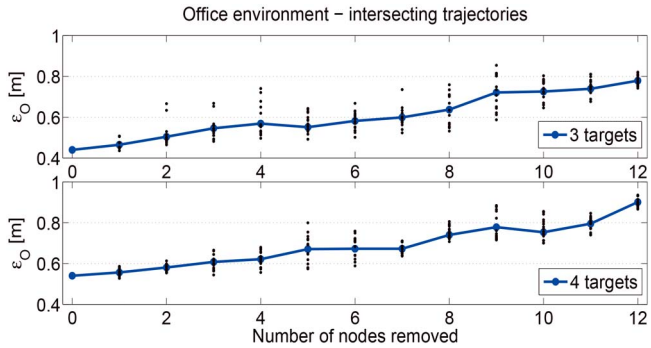


Fig. 7. ϵ_O as a function of nodes' density for the tests carried out in the office environment with three and four targets having intersecting trajectories. The black dots represent ϵ_O for each different simulation.

7 Related Work

In this section, we review previous works related to multiple target tracking with RF sensor networks. In [16], the authors present a clustering algorithm to form clusters of those links whose RSS is affected by the same object. The nodes are positioned on the ceiling of an open indoor environment. The RSS measurements of each identified cluster are then processed separately to localize the targets. This approach achieves a 1.08 m RMSE with two targets. However, to achieve these results, the probabilistic cover algorithm introduced in the paper requires that the two targets are separated by at least 5 m. The latency of the system is approximately 2 s. In [17], the monitored area is divided into different sections, each of which is covered by three nodes communicating on a different frequency channel and positioned on the ceiling as to form a triangle. A support vector regression model is applied to locate the targets. The system can detect and locate two targets when these are positioned in different triangles, *i.e.* at least 2 m apart, with a 0.98 m RMSE. The latency is reduced to 0.26 s.

In [14], tests are carried out in a cluttered book store and in a through-wall scenario with 2 targets having separated trajectories. A particle filter [32] is used for tracking, achieving an RMSE of 0.84 m in the book store and of 1.1 m in the through-wall scenario. The number of targets is assumed to be known a priori. Due to the computational complexity of the particle filter method, tracking is not performed in real-time.

The work in [21] introduces a novel measurement model that assumes that the attenuation in RSS due to the simultaneous presence of multiple targets on the link line is approximately equal to the sum of the attenuations due to the single targets. This additive model is applied in [18], [20]. The tests in [20] are conducted in an outdoor uncluttered environment with up to four targets having separated trajectories. The number of targets is assumed to be fixed

TABLE 6
Comparison Table

Work	$\lvert\mathcal{T}\rvert$	Intersections	Environment	Nodes' Density	ϵ_O [m]	Latency [s]	Assumptions
Zhang *et al.* 2009 [16]	2	no	Uncluttered indoor	0.44 ($16/36m^2$)	1.08	2.00	Targets at least 5 m apart.
Zhang *et al.* 2011 [17]	2	no	Uncluttered indoor	0.36 ($23/64m^2$)	0.98	0.26	Targets at least 2 m apart.
Wilson *et al.* 2011 [14]	2	no	Bookstore Through wall	0.62 ($34/55m^2$) 0.47 ($34/72m^2$)	0.84 1.10	Not in real-time	$\lvert\mathcal{T}\rvert$ fixed and known a priori.
Thouin *et al.* 2011 [21], [20]	4	no	Uncluttered outdoor	0.48 ($24/50m^2$)	0.63 0.77 0.96	7.6 1.7 0.048 (Real-time)	$\lvert\mathcal{T}\rvert$ fixed and known a priori.
Nannuru *et al.* 2012 [18]	3	no	Uncluttered indoor Office	0.38 ($24/64m^2$) 0.30 ($24/81m^2$)	0.80	Real-time if $N_p \leq 100$	$\lvert\mathcal{T}\rvert$ fixed and known a priori. Frequent ϵ_c with varying N_t.
Xu *et al.* 2012 [19]	3	no	Apartment	0.40 ($16/40m^2$)	0.89	Real-time	$\lvert\mathcal{T}\rvert$ fixed and known a priori.
Xu *et al.* 2013 [22]	4	partial overlap	Office Uncluttered indoor	0.15 ($22/150m^2$) 0.05 ($20/400m^2$)	1.08 1.49	0.88	Frequent ϵ_c with overlapping trajectories.
Bocca *et al.* 2013	4	yes	Uncluttered indoor Apartment Office	0.43 ($30/70m^2$) 0.57 ($33/58m^2$) 0.48 ($32/67m^2$)	0.55	Real-time	

and known a priori. For tracking, three types of PF are used. The initial set of particles for each target is drawn from a Gaussian distribution centered around the real target location. The results show the existing trade-off between the tracking accuracy (which depends on the number of particles used per target) and the processing time (which depends on the computational complexity of the tracking algorithms). With four targets, the most accurate tracking algorithm achieves 0.63 m average error with 500 particles per target, requiring in average 7.6 s per time step. When 50 particles per target are used, the most accurate PF achieves 0.77 m error, requiring 1.7 s per time step. On the other hand, the fastest algorithm achieves 0.96 m error in 48 ms per time step. In comparison, our system achieves lower average error (0.55 m) requiring on average 13.3 ms per time step.

In [18], the tests are carried out in an open indoor environment and in an office environment with up to three targets having separated trajectories. The authors use different PFs to track the targets. The filters perform better in average when the particles are initialized according to a Gaussian distribution centered at the true targets locations, which in this case are assumed to be known a priori. The performance of the filters decays when the particles are initialized according to a uniform distribution within the observation region. When the number of targets is assumed to be fixed and known a priori, the most accurate tracking algorithm achieves a RMSE of 0.30 m, 0.72 m, and 0.80 m with one, two, and three targets, respectively. However, when the number of targets is varying, the algorithms are prone to mis-estimate the number of targets. In the tests with two targets, this increases the average OSPA error to 1.35 m when the cardinality penalty $g = 5$. In comparison, our system achieves an average error of 0.45 m, 0.46 m, and 0.55 m with two, three, and four targets, and a maximum OSPA error of 0.83 with four targets when $g = 5$.

The work in [19] proposes a fingerprinting method that uses probabilistic approaches based on discriminant analysis to estimate in which *cells* people are located. Tests are carried out in a one-bedroom apartment with up to three targets having separated trajectories. The number of targets is assumed to be known a priori. The cell estimation accuracy is 89.5% with two targets and 83.5% with three targets. The corresponding average localization errors are 0.82 m and 0.89 m, respectively. In [22], the authors address the problems of counting and localizing multiple targets in the monitored area by relying on the calibration data collected with a single target. Tests are carried out in a large ($150m^2$) office space and in a large ($400m^2$) uncluttered area with up to four targets having partially overlapping trajectories (only two of the four targets walk side by side). The localization error is 1.08 m in the office space and 1.49 m in the uncluttered area. In the case of a fixed number of targets having partially overlapping trajectories, the system estimates correctly the number of targets 80% of times. In comparison, our system estimates correctly the number of targets at least 97% of times, even when the targets have intersecting trajectories and their number is varying. The fingerprinting methods used in [19], [22] require a long calibration period, *i.e.*, 15 minutes for the apartment used in [19] and 30 minutes for the uncluttered area used in [22]. In comparison, the calibration period of our system is much shorter, *i.e.*, 15 seconds. Table 6 summarizes the characteristics of all the systems described in this section.

8 Conclusion

This paper presents an RF sensor network capable of tracking in real-time multiple targets simultaneously moving in an area where low-power wireless transceivers are deployed. The system does not require people to be tracked to participate in the localization task by carrying any radio device or RFID tag. Instead, the RSS measurements collected on the links forming the mesh network on multiple frequency channels are processed to form RTI images, *i.e.*,

images of the change in the propagation field due to the presence of people in the area. Using machine vision methods adapted to RTI, we process the RTI images to detect and track the blobs corresponding to real targets. In this work, we address the situation in which the targets have intersecting trajectories. We apply computationally light-weight methods that can be executed in real-time. Furthermore, the number of targets is varying and is not assumed to be known a priori.

We conduct experiments in three different indoor environments, *i.e.* an open environment with no obstructions nor objects, a one-bedroom apartment with internal walls, furniture and various objects, and a heavily cluttered office environment. In the tests, up to four targets with intersecting trajectories are tracked. The results show that the system correctly estimates the number of targets found in the monitored area and accurately tracks them in real-time in all three environments, also when their trajectories intersect. The measured tracking RMSE ranges from 0.27 m (in the apartment environment with two targets having intersecting trajectories) to 0.55 m (in the office environment with four targets having intersecting trajectories).

In future work, we will address other challenging tracking situations, such as people entering the monitored area side-by-side and then splitting and moving in different directions. In addition, whenever two (or more) targets converge and then separate again, the system has a high probability of not keeping the correct track-to-blob association. To overcome this limitation, we will equip the targets with an active badge and use the AGAPE method presented in [49] to maintain the correct track-to-blob association.

Acknowledgments

This work is supported in part by the U.S. National Science Foundation Grants 0748206 and 1035565 and by the Finnish Funding Agency for Technology and Innovation (TEKES). The authors thank B. Mager for his support in setting up the experiments.

References

[1] J. Wilson and N. Patwari, "Radio tomographic imaging with wireless networks," *IEEE Trans. Mobile Comput.*, vol. 9, no. 5, pp. 621–632, May 2010.

[2] M. A. Kanso and M. G. Rabbat, "Compressed RF tomography for wireless sensor networks: Centralized and decentralized approaches," in *Proc. 5th IEEE Int. Conf. DCOSS*, Marina Del Rey, CA, USA, Jun. 2009.

[3] X. Chen *et al.*, "Sequential Monte Carlo for simultaneous passive device-free tracking and sensor localization using received signal strength measurements," in *Proc. ACM/IEEE IPSN*, Chicago, IL, USA, Apr. 2011.

[4] O. Kaltiokallio and M. Bocca, "Real-time intrusion detection and tracking in indoor environment through distributed RSSI processing," in *Proc. IEEE 17th Int. Conf. RTCSA*, Toyama, Japan, 2011, pp. 61–70.

[5] N. Patwari and J. Wilson, "RF sensor networks for device-free localization and tracking," *Proc. IEEE*, vol. 98, no. 11, pp. 1961–1973, Nov. 2010.

[6] O. Kaltiokallio, M. Bocca, and N. Patwari, "Follow @grandma: Long-term device-free localization for residential monitoring," in *Proc. 7th IEEE Int. Workshop SenseApp*, Clearwater, FL, USA, Oct. 2012.

[7] N. Patwari, J. Wilson, S. Ananthanarayanan, S. K. Kasera, and D. R. Westenskow, "Monitoring breathing via signal strength in wireless networks," ArXiv.org, vol. abs/1109.3898, 2011.

[8] M. McCracken, M. Bocca, and N. Patwari, "Joint ultra-wideband and signal strength-based through-building tracking for tactical operations," in *Proc. 10th Int. Conf. Sensing, Commun. Netw.*, 2013.

[9] T. Hnat, E. Griffiths, R. Dawson, and K. Whitehouse, "Doorjamb: Unobtrusive room-level tracking of people in homes using doorway sensors," in *Proc. 10th ACM Conf. SenSys*, New York, NY, USA, Nov. 2012.

[10] H. Hashemi, "A study of temporal and spatial variations of the indoor radio propagation channel," in *Proc. PIMRC*, Hague, Netherlands, Sept. 1994, pp. 127–134.

[11] M. Ghaddar, L. Talbi, and T. Denidni, "Human body modelling for prediction of effect of people on indoor propagation channel," *Electron. Lett.*, vol. 40, no. 25, pp. 1592–1594, 2004.

[12] J. Wilson and N. Patwari, "See through walls: Motion tracking using variance-based radio tomography networks," *IEEE Trans. Mobile Comput.*, vol. 10, no. 5, pp. 612–621, May 2011.

[13] O. Kaltiokallio, M. Bocca, and N. Patwari, "Enhancing the accuracy of radio tomographic imaging using channel diversity," in *Proc. 9th IEEE Int. Conf. MASS*, Las Vegas, NV, USA, Oct. 2012.

[14] J. Wilson and N. Patwari, "A fade level skew-Laplace signal strength model for device-free localization with wireless networks," *IEEE Trans. Mobile Comput.*, vol. 11, no. 6, pp. 947–958, Jun. 2012.

[15] M. Bocca [Online]. Available: https://sites.google.com/site/boccamaurizio/research

[16] D. Zhang and L. M. Ni, "Dynamic clustering for tracking multiple transceiver-free objects," in *Proc. IEEE Int. Conf. Pervasive Computing Communications.* Washington, DC, USA: IEEE Computer Society, 2009, pp. 1–8.

[17] D. Zhang, Y. Liu, and L. Ni, "Rass: A real-time, accurate and scalable system for tracking transceiver-free objects," in *Proc. IEEE Int. Conf. PerCom*, Washington, DC, USA, Mar. 2011, pp. 197–204.

[18] S. Nannuru, Y. Li, Y. Zeng, M. Coates, and B. Yang, "Radio frequency tomography for passive indoor multi-target tracking," *IEEE Trans. Mobile Comput.*, vol. 12, no. 12, pp. 2322–2333, Dec. 2012.

[19] C. Xu *et al.*, "Improving RF-based device-free passive localization in cluttered indoor environments through probabilistic classification methods," in *Proc. 11th Int. Conf. IPSN*, New York, NY, USA, 2012, pp. 209–220.

[20] S. Nannuru, Y. Li, M. Coates, and B. Yang, "Multi-target device-free tracking using radio frequency tomography," in *Proc. 7th ISSNIP*, Adelaide, SA, Australia, Dec. 2011, pp. 508–513.

[21] F. Thouin, S. Nannuru, and M. Coates, "Multi-target tracking for measurement models with additive contributions," in *Proc. 14th Int. Conf. FUSION*, Chicago, IL, USA, Jul. 2011, pp. 1–8.

[22] C. Xu *et al.*, "SCPL: Indoor device-free multi-subject counting and localization using radio signal strength," in *Proc. 12th Int. Conf. IPSN*, New York, NY, USA, 2013.

[23] T. S. Rappaport, *Wireless Communications: Principles and Practice*. Upper Saddle River, NJ, USA: Prentice-Hall Inc., 1996.

[24] Texas Instruments. *Small Size 2.4 GHz PCB Antenna* [Online]. Available: www.ti.com/lit/an/swra117d/swra117d.pdf

[25] N. Patwari and P. Agrawal, "Effects of correlated shadowing: Connectivity, localization, and RF tomography," in *Proc. IEEE/ACM Int. Conf. IPSN*, St. Louis, MO, USA, Apr. 2008, pp. 82–93.

[26] P. Agrawal and N. Patwari, "Correlated link shadow fading in multi-hop wireless networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 8, pp. 4024–4036, Aug. 2009.

[27] Y. Zhao and N. Patwari, "Noise reduction for variance-based device-free localization and tracking," in *Proc. 8th IEEE Conf. SECON*, Salt Lake City, UT, USA, Jun. 2011

[28] E. Davies, *Computer and Machine Vision: Theory, Algorithms, Practicalities*. London, U.K.: Academic Press, 2012 [Online]. Available: http://books.google.com/books?id=AhVjXf2yKtkC

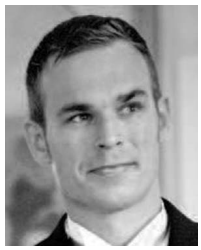
[29] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Mathematical Statistics Probability*, vol. 1. Berkeley, CA, USA, 1967, pp. 281–297.

[30] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, NY, USA: Springer, 2009.

[31] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Boston, MA, USA: Artech House Publishers, 1999.

[32] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Boston, MA, USA: Artech House, Jan. 2004.
[33] Y. Bar-Shalom, F. Daum, and J. Huang, "The probabilistic data association filter," *IEEE Control Syst. Mag.*, vol. 29, no. 6, pp. 82–100, Dec. 2009.
[34] D. B. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Autom. Control*, vol. 24, no. 6, pp. 843–854, Dec. 1979.
[35] G. W. Pulford, "Taxonomy of multiple target tracking methods," *IEE Proc. Radar Sonar Navig.*, vol. 152, no. 5, pp. 291–304, Oct. 2005.
[36] S. S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 19, no. 1, pp. 5–18, Jan. 2004.
[37] H. Leung, Z. Hu, and M. Blanchette, "Evaluation of multiple radar target trackers in stressful environments," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 35, no. 2, pp. 663–674, Apr. 1999.
[38] P. Konstantinova, A. Udvarev, and T. Semerdjiev, "A study of a target tracking algorithm using global nearest neighbor approach," in *Proc. 4th Int. Conf. Computer Systems Technologies: E-Learning*. New York, NY, USA: ACM, 2003, pp. 290–295.
[39] H. W. Kuhn, "The Hungarian method for the assignment problem," *Nav. Res. Logist. Q*, vol. 2, no. 1–2, pp. 83–97, 1955.
[40] F. Bourgeois and J.-C. Lassalle, "An extension of the Munkres algorithm for the assignment problem to rectangular matrices," *Commun. ACM*, vol. 14, no. 12, pp. 802–804, Dec. 1971
[41] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME J. Basic Eng.*, vol. 82, pp. 35–45, 1960.
[42] G. Welch and G. Bishop, "An introduction to the Kalman filter," Dept. Comput. Sci., Univ. North Carolina, Chapel Hill, NC, USA, Tech. Rep. 95-041, 1995.
[43] Texas Instruments. *A USB-Enabled System-on-Chip Solution for 2.4 GHz IEEE 802.15.4 and ZigBee Applications* [Online]. Available: http://www.ti.com/lit/ds/symlink/cc2531.pdf
[44] M. Bocca, O. Kaltiokallio, and N. Patwari, "Radio tomographic imaging for ambient assisted living," in *Proc. EvAAL*, Berlin, Germany, 2013.
[45] *IEEE 802.15.4 Standard Technical Specs* [Online]. Available: http://www.ieee802.org/15/pub/TG4Expert.html
[46] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis, "Understanding the causes of packet delivery success and failure in dense wireless sensor networks," in *Proc. 4th Int. Conf. SenSys*, New York, NY, USA, 2006, pp. 419–420.
[47] J. Hoffman and R. Mahler, "Multitarget miss distance via optimal assignment," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 34, no. 3, pp. 327–336, May 2004.
[48] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A consistent metric for performance evaluation of multi-object filters," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3447–3457, Aug. 2008.
[49] Y. Zhao, N. Patwari, P. Agrawal, and M. Rabbat, "Directed by directionality: Benefiting from the gain pattern of active RFID badges," *IEEE Trans. Mobile Comput.*, vol. 11, no. 5, pp. 865–877, May 2012.

Maurizio Bocca received the B.Sc. (2003) and M.Sc. (2006) degrees in Computer Science Engineering from the Politecnico di Milano (Milan, Italy), and the Ph.D. (2011) in Electrical Engineering from Aalto University (Helsinki, Finland). In 2012, he joined as a post doc the Sensing and Processing Across Networks (SPAN) Lab at the University of Utah (Salt Lake City, Utah, USA), where he is conducting research in the area of RF sensor networks for indoor device-free localization, context awareness and elder care.



Ossi Kaltiokallio received the B.Sc and M.Sc degrees in electrical engineering from Aalto University, School of Electrical Engineering, Helsinki, Finland, both in 2011. He is currently a Ph.D. student in the Department of Automation and Systems Technology, Aalto University School of Electrical Engineering. He is a member of the Wireless Sensor Systems Group at Aalto University. His current research interests include RSS-based localization, signal processing, and design and implementation of embedded wireless systems.

Neal Patwari received the B.S. (1997) and M.S. (1999) degrees from Virginia Tech, and the Ph.D. from the University of Michigan, Ann Arbor (2005), all in Electrical Engineering. He was a research engineer at Motorola Labs, Florida, between 1999 and 2001. Since 2006, he has been at the University of Utah, where he is an Associate Professor in the Department of Electrical and Computer Engineering, with an adjunct appointment in the School of Computing. He directs the Sensing and Processing Across Networks (SPAN) Lab. His research interests are in radio channel signal processing, in which radio channel measurements are used to benefit security, networking, and localization applications. He is a member of the IEEE.

Suresh Venkatasubramanian received his Ph.D degree from Stanford University in 1999. After spending seven years at AT&T Labs, he came to the University of Utah in 2006, where he is now the Warnock Assistant Professor in the School of Computing. His interests include algorithms, computational geometry, data mining and the challenges of large data processing. He received an US NSF CAREER award in 2009.