RASS: A Real-Time, Accurate, and Scalable System for Tracking Transceiver-Free Objects

Dian Zhang, *Member*, *IEEE*, Yunhuai Liu, *Member*, *IEEE*,
Xiaonan Guo, *Member*, *IEEE*, and Lionel M. Ni, *Fellow*, *IEEE*

Abstract—Transceiver-free object tracking is to trace a moving object that does not carry any communication device in an environment with some monitoring nodes predeployed. Among all the tracking technologies, RF-based technology is an emerging research field facing many challenges. Although we proposed the original idea, until now there is no method achieving scalability without sacrificing latency and accuracy. In this paper, we put forward a real-time tracking system RASS, which can achieve this goal and is promising in the applications like the safeguard system. Our basic idea is to divide the tracking field into different areas, with adjacent areas using different communication channels. So, the interference among different areas can be prevented. For each area, three communicating nodes are deployed on the ceiling as a regular triangle to monitor this area. In each triangle area, we use a Support Vector Regression (SVR) model to locate the object. This model simulates the relationship between the signal dynamics caused by the object and the object position. It not only considers the ideal case of signal dynamics caused by the object, but also utilizes their irregular information. As a result, it can reach the tracking accuracy to around 1 m by just using three nodes in a triangle area with 4 m in each side. The experiments show that the tracking latency of the proposed RASS system is bounded by only about 0.26 m. Our system scales well to a large deployment field without sacrificing the latency and accuracy.

Index Terms—Applications, pervasive computing, localization, wireless sensor networks, transceiver-free, support vector regression

1 Introduction

Real-time tracking of moving objects has been a hot research issue as it is important in many applications, such as vehicle tracking using GPS [1], patient location finding [2] inside a hospital, and animal migration behavior learning [3] in a wild place. Most of these applications require each object to carry a device as the transceiver to help estimate its location. However, such a requirement can hardly be met in some applications. For example, in a safeguard system, thieves are impossible to carry any device to be tracked. Or even in some places, especially at home, where people do not always carry their tracking device (e.g., the cell phones) and they often leave them behind [4]. Although there are some nonradio frequency (RF)-based technologies able to track the object without carry any device, they have many limitations. The traditional infrared [5] and pressure technologies require [6] dense deployment and their cost is high. Video technology [7] cannot work in dark place and keep the privacy of people. Laser range [8] is famous for its high accuracy of distance measurement, but with prohibitively high cost. Therefore, low-cost RF-based technologies to track objects that does not carry any communication device has caught the attention of some researchers. It is called transceiver-free [10], [11] or device-free [9] object tracking.

To the best of our knowledge, we were the first group that proposed the original idea [10], as well as provided some preliminary solutions and initial experimental results. Our first attempt was based on a wireless network composed of a number of communicating nodes (each node can be a simple sensor node without using its sensing ability). Nodes are periodically sending beacon messages. The system first detects the *Radio Signal Strength (RSS) dynamics* for each pair of communicating nodes, which is the difference of signal strength caused by the object. Then, it proposed a model of RSS dynamics to allow tracking transceiver-free objects. Based on this model, it utilized many such pairs of communicating nodes to locate the object.

However, the previous model is only based on one pair of nodes. Its accuracy may be improved by introducing more nodes to eliminate the noise behavior. But introducing more nodes will increase the possibility of interference among nodes. Moreover, each node has to wait for a back off time in order to avoid beacon collision. Therefore, the detection latency will be dramatically increased. This problem becomes more serious if it is applied in a large area deployment.

To solve the above problem, in this paper, we put forward a brand-new real-time tracking system RASS, which utilizes multichannels to monitor different area of the tracking field.

- *D. Zhang is with the College of Computer Science and Software Engineering, Shenzhen University, Nanhai Ave 3688, Shenzhen, Guangdong 518060, P.R. China. E-mail: serena.dian@gmail.com.*
- *Y. Liu is with the Research Center of Internet of Things, The Third Research Institute of the Ministry of Public Security, 76, Yue Yang Road, Shanghai. E-mail: yunhuai.liu@gmail.com.*
- *X. Guo is with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong. E-mail: guoxn@ust.hk.*
- *L.M. Ni is with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, and Shanghai Jiao Tong University. E-mail: ni@ust.hk, ni@cse.ust.hk.*

Manuscript received 6 Dec. 2011; revised 16 Mar. 2012; accepted 18 Mar. 2012; published online 26 Apr. 2012.
Recommended for acceptance by J. Cao.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDS-2011-12-0886.
Digital Object Identifier no. 10.1109/TPDS.2012.134.

Therefore, we may prevent interference among wireless nodes belonging to different channels from happening. This is helpful to improve the tracking accuracy. More importantly, since we only need to consider the wireless communication among nodes in the same channel, the beacon interval of each node can be greatly shortened. As a result, the tracking latency is dramatically decreased. Furthermore, we propose a new model to allow tracking transceiver-free objects. The model is based on just three communicating nodes deployed on the ceiling of the tracking field as a regular triangle. This model utilizes both irregular and regular RSS dynamic information to locate the object. This approach is totally different from our previous model, which only considers the ideal case. Based on our new model, the tracking accuracy can reach 1 m by just using three nodes, while the previous methodology should use 16 nodes to reach a similar accuracy. Furthermore, using three nodes in each channel can further decrease the beacon interval of each node, resulting in a much lower tracking latency.

Our basic approach is to divide the tracking field into different triangle areas. Each triangle area is set up by three communicating nodes deployed on the ceiling of tracking field. The adjacent triangle areas will be assigned with different channels. At first, for each triangle area, we set up a regression model. This model uses *Support Vector Regression* (SVR) to simulate the relationship between the RSS dynamics caused by the transceiver-free object and the object location on the ground. Then, we use this model to locate the object for each triangle area. Moreover, we can easily locate multiple objects, as long as they are in different triangle areas. Our experiments use TelosB [13] as communicating nodes. The experimental results show that the tracking error is around 1 m. More excitingly, the latency of our RASS can reach about 0.26 s with greater scalability and without sacrificing the accuracy.

To sum up, our contributions are as below:

- *Very low latency.* We are the first to introduce multichannel assignment in tracking transceiver-free objects. As a result, the beacon interval of each node is greatly decreased. The tracking latency of our system is bounded by only 0.26 s, which offers excellent response time and is potential for most of the emergency cases. Even in a large area deployment, the tracking latency will not increase.
- *Scale well without sacrificing the latency and accuracy.* We are the first to use SVR model in tracking transceiver-free objects. By using this model, we can reach 1 m accuracy by only using three nodes deployed in a triangle area with 4 meters per side. Even the environment changes, we only need a few recalibrations as our new model can be easily obtained. Moreover, through using multichannels, tracking on different areas of the whole field can be independently performed, resulting in perfect scalability. Theoretically, it can be deployed as large as possible without sacrificing the latency and accuracy, while other systems usually are hard to obtain the two performance matrices simultaneously.
- *High accuracy.* Our methodology is based on multiple channel assignment. Since different triangle areas use different channels, we can avoid interference between adjacent areas to improve the accuracy. Moreover, our SVR tracking model shows about 150 percent better than the previously proposed methods.

The rest of this paper is organized as follows: In the next section, we will discuss relevant related work. Section 3 introduces the SVR tracking model and our channel assignment method, followed by the experimental results and evaluation of the performance. Finally, Section 5 concludes the paper and lists our future work.

2 Related Works

Nowadays, tracking transceiver-free objects can be divided into two categories. One is the non-RF-based technologies and the other is RF-based technologies.

There are some non-RF technologies which may or may partially locate the transceiver-free objects. The infrared technology [5] can count how many people crossing a bounded area by monitoring its entries. Pressure technologies [6] can detect people's footsteps by using acceleration and air pressure sensors. However, these technologies require dense and careful deployment. Laser ranging [8] can achieve high accuracy of its distance measurement. But, its cost is prohibitive nowadays. Video technology [7] cannot work in dark place and protect the privacy of people. Ultrasound [12] only has the ability to count on the number of objects passing by.

RF-based technologies to track transceiver-free objects are still in its infancy. Device-free passive localization [9] points out the challenges lying ahead and observes radio dynamics with just two pairs of 802.11 transceivers. This work, however, is later than us and only provided some raw data from the experiments. Neither real deployment nor real experimental results were reported. Its incremental work [23] focuses on the detection function in a real environment, however, only tracking of single object is considered. Tag-free [24] aims to find the object trajectory without carrying any device, through using data mining methods. Our work is very different from it. At first, their experiments are based on RFID [21] platform. Our work is based on wireless sensor networks. Second, they leverage data mining methods, which require laborious training. Even worse, if environment changes, training should be performed again. ILight [25] tries to track moving object without carrying any device by using light sensors. However, this approach requires dense deployment and cannot work in dark area. Work [26] also does some experiments to track transceiver-free object. However, its experiment is only performed outdoors. Their methodology is not scalable and requires laborious training.

Another two works [10], [11], both proposed by us before, proposed a model for just one pair of communicating nodes, and offered three algorithms based on the signal changing behaviors among a group of nodes on the ceiling. Our new work is dramatically different from them. First, the previous model can only deal with the ideal case. Our new model is based on SVR, which utilizes the irregular information of RSS dynamics to locate the object. As a result, it can accurately estimate the object location by using only three

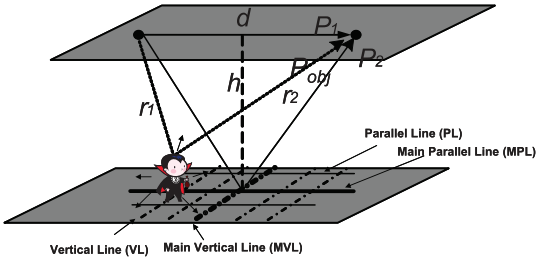


Fig. 1. Test with two communicating nodes. d: the distance between two nodes. h: the height of the nodes from the ground. r_1: the distance from the transmitter to the target. r_2: the distance from the target to the receiver. P_1: the line-of-sight path of the signal. P_2: the ground reflection path of the signal. P_{obj}: the reflection path by the target.

communicating nodes, while previous works require 16 nodes to get a similar accuracy. Second, previous works do not scale well especially if deployed in a very large area. Both latency and tracking error will dramatically increase. The reason is that in a large deployed area, the probability that more nodes transmit at the same time will become higher. Therefore, the interference problem becomes serious. Moreover, in order to avoid beacon collision, each node has to wait a back off time before transmitting. Thus, their latency will be very long. Our new method has a much lower latency at around 0.26 s and can scale very well. The latency is in no relationship with the size of the deployed area and our RASS system is a fast tracking system.

3 Methodology

Our RASS tracking system utilizes TelosB [13] sensor nodes in multichannels to monitor different area of the tracking field. It can reduce the interference among the nodes belonging to different channels. So, the questions are what is the best node deployment for one channel? How many nodes should be in one channel area? And what size should one channel area cover?

Actually, if we introduce many nodes in one channel area, more information may be obtained to improve the tracking accuracy. However, introducing more nodes will increase the interference among nodes. As a result, it may reduce the tracking accuracy to some extent. Moreover, to avoid beacon collisions, the beacon interval of each node should be prolonged, increasing the detection latency. Therefore, how to choose the number of nodes is a tradeoff between latency and accuracy.

Then, if we settle down the node deployment for each channel area, the question comes to what kind of channel assignment method shall we adopt?

In order to answer the questions above, in this section, at first we will discuss how to choose the basic node deployment layout within one channel area. We will begin with some theoretical background, followed by the basic deployment and measurement. We will start analysis from the minimum number of nodes able to cover an area and show different choices. In the following, we design a SVR model based on such deployments, which utilize both irregular and regular RSS dynamic information caused by the object to be located. At last, a novel channel assignment method is proposed.

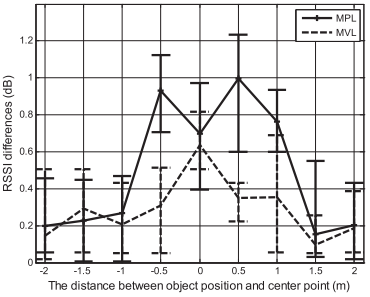


Fig. 2. RSS dynamics with 3 m node distance. MPL is Main Parallel Line and MVL is Main Vertical Line defined in Fig. 1.

3.1 Theoretical Background and Practical Cases

3.1.1 Two Communicating Nodes: Ideal Case

Our previous work [10] showed how the object's position on the ground will cause the RSS dynamics for just one pair of nodes hang on the ceiling of the floor. One of them is the transmitter and the other one is the receiver, as shown in Fig. 1.

At first, in the static environment, the environmental factors are stable and no object moves around. The RSS dynamic is nearly zero (the received radio signal of each node is stable). When an object comes into this area and is close by, the object will cause the radio signal to change. It means the RSS dynamics will grow larger. The difference of power caused by the object $\triangle P$ can be calculated as below:

$$\triangle P = \frac{P_t G_t G_r \lambda^2 \sigma}{(4\pi)^3 r_1^2 r_2^2}. \quad (1)$$

Here, r_1 and r_2 are the distances from the target object to the transmitter and receiver, respectively. σ is the radar cross section of the target object, which is a fixed value for a certain kind of object (for human being, this value is fixed at 1). P_t is the transmitted power in watts, G_t , G_r are the transmitter and the receiver antenna gain, respectively. λ is the radio wavelength in meters.

It was concluded that, when the object is closer to the midpoint of each PL or VL line, the RSS dynamic caused by the object is larger. It can also be proven MPL and MVL are the two typical lines having the largest RSS dynamics (shown in Fig. 2). Here, MPL is the mapping line on the ground of the two communicating nodes. MVL is vertical to MPL crossing the midpoint of MPL. According to this model, if we have detected a RSS dynamic value ΔP, according to the upper equation, we have

$$r_1 r_2 = \sqrt{\frac{P_t G_t G_r \lambda^2 \sigma}{(4\pi)^3 \triangle P}}. \quad (2)$$

We may prove that the possible object area on the ground is like an eclipse on the ground [10].

3.1.2 Two Communicating Nodes: Practical Situation

Actually, the model introduced before only is valid in an ideal situation. In real experiments, the result is much more complicated. Although the basic rules are satisfied, many factors will influence the RSS dynamics, such as interference among nodes, multipath effects, and absorption by

structures and human beings. Moreover, temperature and humidity may affect the signals. Therefore, the RSS dynamics vary in some extend even for the same object position. The following are some measurements based on different deployments.

At first, we test different node distances with two communicating nodes (one communication link in between). Since MPL and MVL are two typical lines having the largest RSS dynamics, we only show RSS dynamics according to different object positions at such places. Fig. 2 shows an example when the node distance is at 3 m. We may see that, in general, the upper rule is satisfied. That is, when the object position is closer to the midpoint of each line, the RSS dynamics are larger. However, the object in the same position may cause different RSS dynamics in different measurements. The error bars in the figures represent such difference based on a number of samples. From those error bar information shown in the figure, even for some object positions not close to the midpoint of MPL or MVL, the RSS dynamics sometimes vary.

3.1.3 Three Nodes Deployed as a Triangle

Let us start thinking a three-node deployment on the ceiling of the floor. Three nodes are the minimum number of nodes to cover an area. We set them as a regular triangle since regular shape is the best topology for the coverage problem [16]. Each node will periodically broadcast beacons, while receiving messages from the other two nodes. Therefore, totally, there are three pairs of links (here, we regard the two nodes transmitting and receiving to each other as one link).

According to the theoretical result based on two commutating nodes, for each link, e.g., Link 1, only if the object is in the defined area (the eclipse area colored in gray in the upper left subfigure of Fig. 4), it will cause a RSS dynamic of Link 1. Similar phenomena hold for Links 2 and 3. Hence, if any RSS dynamic is detected, the theoretical method will assume the object will not appear in the other area beside the three gray eclipses.

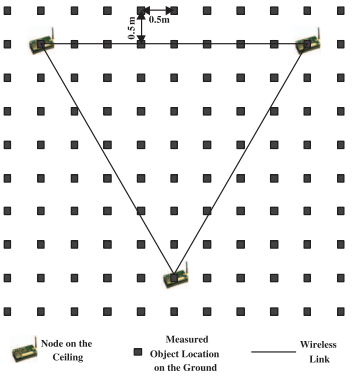


Fig. 3. Tested object locations for $4\ \mathrm{m}$ triangle. This Figure is in a bird view. The three wireless nodes are put on the ceiling of the floor. The tested target locations are on the ground.

However, in real environments, it is not always the case. Fig. 3 shows the object test positions in a 4 m triangle. Here, we arrange a person to act as the target object and test different positions. Then, we observe the RSS dynamics for all the three links. The bottom three subfigures of Fig. 4 show how the object positions above will influence the RSS dynamics of Links 1, 2, and 3, respectively. Dark color in the figure means large RSS dynamics and shallow means small RSS dynamics. For example, the left bottom subfigure of Fig. 4 shows the real RSS dynamic map for Link 1. If at some object position p, we find the color of such position is dark, it means the person standing at position p will cause a large RSS dynamic for Link 1. On the contrary, if at another object

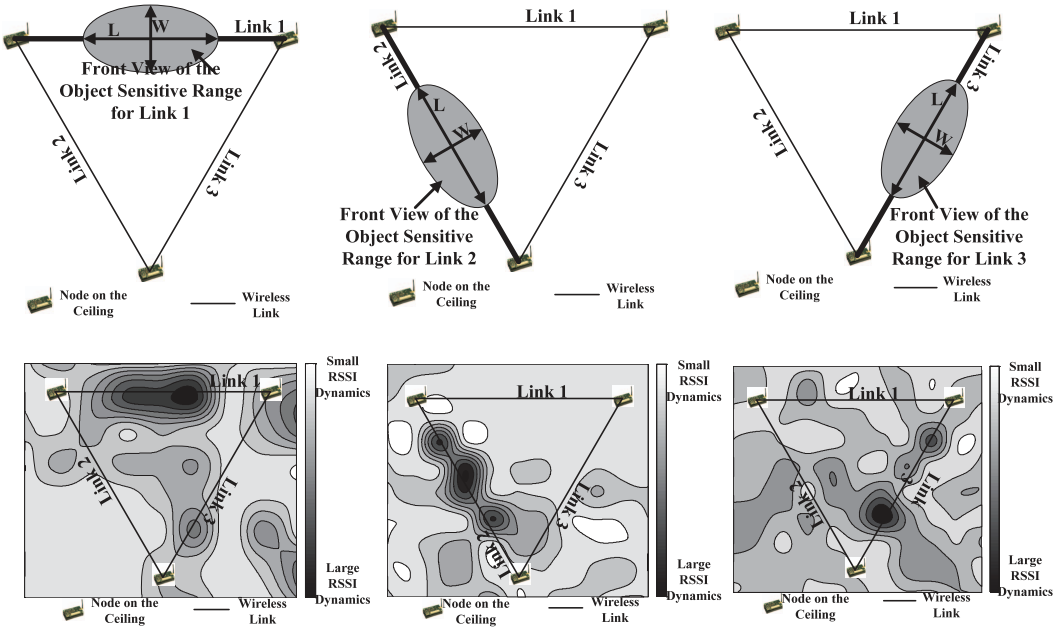


Fig. 4. Object area concluded from model causing RSS dynamic and Real RSS dynamic map. The six subfigures are all in a bird view, the three nodes are on the ceiling. The upper three subfigures are the object area causing RSS dynamic for links 1, 2, and 3. Only the target is in the ellipse area, it will cause RSS dynamic for links 1, 2, and 3. The bottom three subfigures are the real RSS dynamic map for links 1, 2, and 3. Different colors on the ground means different RSS dynamics caused by the target in the same place.

position p', we find the color of such position is shallow, it means the person standing at position p' will cause a small RSS dynamic for Link 1.

From these figures, we may see that, in general, if an object is closer to one link, it will bring larger RSS dynamics for the corresponding link and vice versa. However, for some object positions not in the model area, the object still may cause some dynamics for some of the three links.

3.2 How to Choose the Node Deployment in One Channel

In fact, how to choose the node deployment in one channel depends on user requirements. If they are strict with the response time of the tracking system, the latency is the first issue. Otherwise, tracking error is of importance. As we introduced before, how to choose them to get a good system performance is a tradeoff between latency and accuracy.

Since our RASS system is a real-time system, we would like to shorten the latency while improving the tracking accuracy as much as possible. Therefore, the triangle setting with three nodes to cover an area would be the best for the latency. Because three nodes are the minimum number of setting to cover an area, we only need to consider the wireless communication among three nodes. The beacon interval of each node to avoid transmission collision is the shortest. But, the question comes to what is best tracking accuracy we may achieve based on such deployment?

To answer this question, we have to look back into the former real case of the regular triangle setting. If we use the previous model, many object positions inside the triangle area cannot be correctly located. The reason is that it assumes an object not in the model area (the eclipse area colored in gray) will not cause RSS dynamics. The previous tracking method [10] achieves 1 m accuracy only by introducing many nodes to construct many overlapping model areas.

Thus, can we have a brand-new model just based on the regular triangle setting of only three nodes? Can we find a new model which can utilize the irregular information outside the model area to well locate the object position? Fortunately, SVR is such a method we find which is able to solve this problem and their tracking accuracy can reach around 1 m for the 2, 3, and 4 m triangle settings in our later experiments.

3.3 Model Construction Using SVR

Support Vector Regression [14] is commonly used in forecasting the financial market and reconstruction of chaotic systems. It aims to find a hyperplane which can accurately predict the training data.

Considering our localization problem. We leverage SVR to determine the object location according to the following procedure. At first, we should have some samples before tracking. In each sample, we should collect both the object location and the RSS dynamic caused by this object. Based on all these samples, we would like to train a tracking model to simulate the relationship between the RSS dynamics with the object locations. We then use this SVR model to perform prediction in the tracking state. How to train the SVR model is as follows: Some important notations that will be used to train the model are listed in Table 1.

TABLE 1
Main Notations Used in Section 3.3

Notation	Description
d	Number of links
n	Number of samples (object locations)
k	Dimension of each object location
$X = \{x_i^d\}, i \in (1, n)$	Input data (the RSS dynamics of different links)
$Y = \{y_i^k\}, i \in (1, n)$	Output data (the target object locations)
$f(x)$	The SVR model function
$k(x_i, x)$	Kernel function for similarity measure
Γ	Cost function
ε	Tolerance parameter
c	Penalty to estimate errors

Our setting is based on a regular triangle setting with three nodes. For each triangle with three nodes, there are three links. So, each object position on the ground causes the RSS dynamics of Links 1, 2, and 3. Suppose we have n samples: n object locations and their causing RSS dynamics in the triangle area. The space of the input pattern X is a three dimension data recording the RSS dynamics of Links 1, 2, and 3. These data are denoted by

$$X \in \Re^d, X = \{x_i^d\}, x_i^d = [x_1^d, x_2^d, \ldots, x_n^d]. \tag{3}$$

Here, d is the number of links. In our triangle, setting this value is 3. n is number of test object locations.

The target class Y represents the object location on the ground. It is denoted by

$$Y \in \Re^k, Y = \{y_i^k\}, y_i^k = [y_1^k, y_2^k, \ldots, y_n^k]. \tag{4}$$

Here, k is dimension of object location on the ground. In our triangle setting, this value is 2. n is number of test object locations. This value is the same defined in the last paragraph. Given the training data

$$\{(x_1^d, y_1^k), \ldots, (x_n^d, y_n^k)\}, \tag{5}$$

our goal is to find a function

$$f(x) = w \cdot \Phi(x) + b, \Phi : \Re^d \rightarrow f, w \in \Re^d, b \in \Re \tag{6}$$

that has at most a tolerance parameter ε from the actually obtained targets y_i^k for all the samples and at the same time is as flat as possible [14]. In other words, it does not care about errors as long as they are less than the tolerance parameter ε, but will not accept any deviation larger than this. $f(x)$ outputs the locations as Fig. 5 shows.

This regression estimation function can be rewritten by Lagrange multipliers α_i, α_i^* as

$$\begin{aligned} f(x) &= \sum_{i=1}^{n} (\alpha_i - \alpha_i^*)(\Phi(x_i) \cdot \Phi(x)) + b \\ &= \sum_{i=1}^{n} (\alpha_i - \alpha_i^*) k(x_i, x) + b. \end{aligned} \tag{7}$$

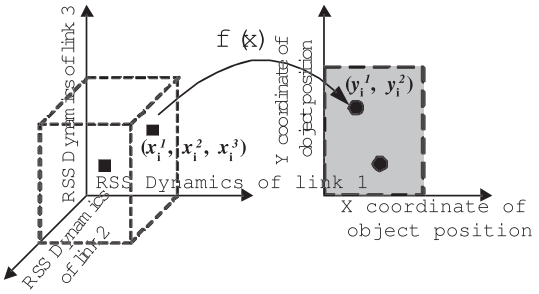


Fig. 5. SVR for localization. $f(x)$ simulate the relation between the RSS and object location.

The function $k(x_i, x)$ is the kernel function for similarity measure. It also should minimize the regression risk,

$$R_{reg}(f) = c\sum_{i=1}^{n}\Gamma(f(x_i) - y_i) + \frac{1}{2}\|w\|^2. \tag{8}$$

Here, Γ define the cost function as ε-intensive loss function

$$\Gamma(f(x) - y) = \begin{cases} |f(x) - y| - \varepsilon, & |f(x) - y| \geq \varepsilon \\ 0, & |f(x) - y| < \varepsilon. \end{cases} \tag{9}$$

And

$$w = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)\Phi(x_i). \tag{10}$$

The constant $c > 0$ determines penalties to estimation errors. It determines the tradeoff between the flatness of f and the amount up to which deviations larger than ε. ε is a radius value, which means that the data inside ε-tube are ignored in regression.

The upper method is included in the standard library LIBSVM [15]. We utilize it to train an SVR model from X to Y under the triangle setting. After we get this model, when a new RSS dynamics vector is received at the tracking status, we may predict the object location by using this SVR tracking model.

3.4 Adaptive Learning with Very Small Samples

In fact, different triangles with different node distances have different RSS dynamic maps. If we choose the same triangle setting in the real application, we can set up an SVR model from one of the triangles as introduced before. This SVR model can be regarded as the general model and applied in predicting the object positions.

But if users would like to get higher tracking accuracy, they would better choose to train different models for different triangles at different places. But, it would be a big effort if we calibrate all the samples on each new triangle at different places. Fortunately, since same triangles have similar properties, their relationship between the RSS dynamic and the object position is also similar. We regard the RSS dynamic vector distance for each pair of object points is similar. As a result, if we know the RSS dynamics of a few samples (noted as *reference locations* in the next section), we may derive the RSS dynamics of other object places according to such relation (similar RSS dynamic vector distance), making the calibration efforts reduced.

In the following, we offer an adaptive learning method. In this method, only three samples are required for each new triangle. The others can be generated based on the old samples for the SVR model. Then, we may use the new generated samples to refine a new SVR model for the new triangle.

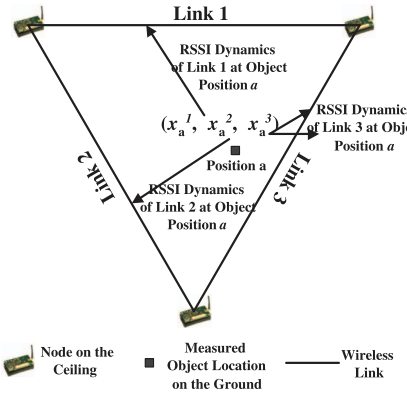


Fig. 6. RSS dynamic vector. It is three dimension vector representing different RSS dynamics for the three links.

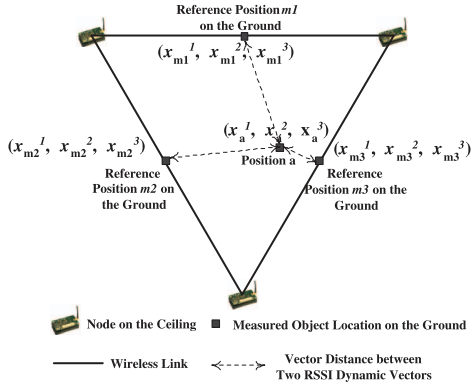


Fig. 7. Adaptive learning on RSS dynamics. $m1$, $m2$, and $m3$ are the only pointed need to be retested in the new triangle.

3.4.1 Vector Distance to Reference Points

Let's look into the old model samples in triangle A. For each object location on the ground under the triangle, e.g., point a in Fig. 6, its dynamic vector is

$$x_a = \left[x_a^1, x_a^2, x_a^3\right]. \tag{11}$$

x_a^1, x_a^2, and x_a^3 are the RSS dynamics for Links 1, 2, and 3, respectively. Then, we introduce three reference locations $m1$, $m2$, and $m3$, as depicted in Fig. 7. Their RSS dynamic vectors are

$$\begin{aligned} x_{m1} &= \left[x_{m1}^1, x_{m1}^2, x_{m1}^3\right] \\ x_{m2} &= \left[x_{m2}^1, x_{m2}^2, x_{m2}^3\right] \\ x_{m3} &= \left[x_{m3}^1, x_{m3}^2, x_{m3}^3\right]. \end{aligned} \tag{12}$$

We choose these points as reference points because these object positions generally can cause large RSS dynamics for Links 1, 2, or 3. For each other position, e.g., point a, we calculate the RSS dynamic vector distance to each reference point as follows:

$$\begin{aligned} D_{a-m1} &= \left(\left(x_a^1 - x_{m1}^1\right)^2 + \left(x_a^2 - x_{m1}^2\right)^2 + \left(x_a^3 - x_{m1}^3\right)^2\right)^{\frac{1}{2}} \\ D_{a-m2} &= \left(\left(x_a^1 - x_{m2}^1\right)^2 + \left(x_a^2 - x_{m2}^2\right)^2 + \left(x_a^3 - x_{m2}^3\right)^2\right)^{\frac{1}{2}} \\ D_{a-m3} &= \left(\left(x_a^1 - x_{m3}^1\right)^2 + \left(x_a^2 - x_{m3}^2\right)^2 + \left(x_a^3 - x_{m3}^3\right)^2\right)^{\frac{1}{2}}. \end{aligned} \tag{13}$$

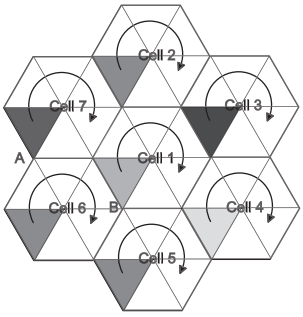


Fig. 8. Topology of Channel Assignment. Each hexagon is a cell which will be assigned a specific channel.

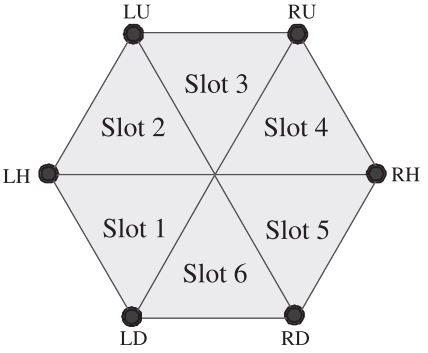


Fig. 9. Selection of the triangles in a cell. Different time slot will choose different triangle.

We repeat this procedure until the vector distance between each other object location and each reference point is calculated.

Based on the upper procedure, we may get the RSS dynamic relation between any other object location and each reference points. According to such relation, for any new triangle with the same size and setting, we may conclude the RSS dynamics of different object positions as long as we get the RSS dynamics of the reference points in the new triangle.

3.4.2 Interpolation by Triangulation

Suppose there is a new triangle A', which is not trained. The shape and size of triangle A' is the same as triangle A. First, we choose the object positions $m1'$, $m2'$, $m3'$ (the same place with $m1$, $m2$, $m3$) in triangle A' as reference points. We collect their RSS dynamic vectors when object (e.g., a person) is in these positions:

$$\begin{aligned} x_{m1'} &= \left[x^1_{m1'}, x^2_{m1'}, x^3_{m1'}\right] \\ x_{m2'} &= \left[x^1_{m2'}, x^2_{m2'}, x^3_{m2'}\right] \\ x_{m3'} &= \left[x^1_{m3'}, x^2_{m3'}, x^3_{m3'}\right]. \end{aligned} \tag{14}$$

Then, for each other object location, e.g., position a', we may deduct its RSS dynamic vector $x_{a'} = [x^1_{a'}, x^2_{a'}, x^3_{a'}]$ by triangulating with D_{a-m1}, D_{a-m2}, and D_{a-m3} obtained in the last section. It is to solve the following equation function:

$$\begin{aligned} D_{a-m1} &= \left(\left(x^1_{a'} - x^1_{m1'}\right)^2 + \left(x^2_{a'} - x^2_{m1'}\right)^2 + \left(x^3_{a'} - x^3_{m1'}\right)^2\right)^{\frac{1}{2}} \\ D_{a-m2} &= \left(\left(x^1_{a'} - x^1_{m2'}\right)^2 + \left(x^2_{a'} - x^2_{m2'}\right)^2 + \left(x^3_{a'} - x^3_{m2'}\right)^2\right)^{\frac{1}{2}} \\ D_{a-m3} &= \left(\left(x^1_{a'} - x^1_{m3'}\right)^2 + \left(x^2_{a'} - x^2_{m3'}\right)^2 + \left(x^3_{a'} - x^3_{m3'}\right)^2\right)^{\frac{1}{2}}. \end{aligned} \tag{15}$$

Since D_{a-m1}, D_{a-m2}, and D_{a-m3} are already known as introduced in the last section, we may deduct $x^1_{a'}$, $x^2_{a'}$ and $x^3_{a'}$ accordingly.

Repeating such procedure, at last we may generate the RSS dynamics vector for each object location for the new triangle A'. After finishing all the interpolation, a new prediction model is constructed by using SVR introduced before. Therefore, we may easily refine a new SVR model, as long as we only measure three samples for the three reference points of the new triangle.

3.5 Multichannel Assignment

Since our RASS tracking system utilizes multiple channels to monitor different areas of the tracking field, it is able to reduce the interference among the nodes belonging to different channels. So, it can improve the tracking accuracy. Moreover, we only need to consider the wireless communication in the same channel. The beacon interval to avoid transmission collision can be greatly shortened.

Because a triangle setting is the basic tracking component in our model, we use triangles to fully cover the whole monitored space. After deployment, we want to obtain a topology like Fig. 8, where only the communication links between each two adjacent nodes exist. In order to ensure the communication topology and to avoid link interference and transmission collisions, we utilize the multichannel ability of the nodes by using a synchronized slot-based channel assignment scheme.

In our scheme, we first define a cell, which is a hexagon composed of six adjacent triangles. As shown in Fig. 8, there are seven cells in total. And we define the center node in a cell as a leader, and the other six surrounding nodes as assistants. It is obvious that one node can be either a leader or an assistant. While a leader always belongs to one cell, an assistant may belong to up to three adjacent cells. For example, assistant A falls into cells 6 and 7, but assistant B falls into cells 5, 6, and 1.

We then assign each cell a specific channel so that all the nodes in the same cell will use the assigned channel to communicate. The nodes are synchronized and time slots are assigned by the following scheme: for each slot for each cell, there are only three adjacent nodes able to transmit, as shown in Fig. 9. We call the triangle formed by the three adjacent nodes in the same channel as a selected triangle. In our scheme, one selected triangle continues for one time slot and then changes clockwise. Thus, after six slots, all the triangles in the cell have been selected once. So, it is like a triangle to sweep clockwise around the cell.

For example, as depicted in Fig. 10, channel 1 is assigned to cell 1. The leader in the cell always stays in the same channel, while assistants may only stay in channel 1 for some slots and then change to other channel to serve for other cells. Since the selection of the triangles is fixed, it is clear that once the assistant knows its relative orientation to the leader, its corresponding slots for the channel are determined. For example, for the assistant which is left down to the leader, it stays in channel 1 at slots 1 and 6; for the assistant which is right up to the leader, it stays in channel 1 at slots 2 and 3.

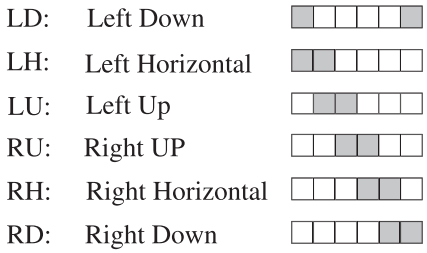


Fig. 10. Orientation-slot occupation mapping table. LD, LH, LU, RU, RH, and RD are different wireless nodes illustrated in Fig. 9.

In a real deployment, the channel assignment procedure consists of three phases:

1. *Initialization.* We first set up the frame of axes before deploying the nodes. After determining the location of each node, we select the cells and assign each cell a unique channel. Each node will remember its location information. If it is a leader node, it also records its leader identity, the number of its assistants, and its corresponding channel for the cell. This part is done offline.

2. *Identity determination.* Each leader broadcasts its location information and its identity status to its one-hop neighbors. Once the assistant nodes get the information from one leader, they calculate their relative orientation to the leader node and assign the leader's channel with slots according to the orientation-slot occupation mapping table. The assistant nodes then acknowledge to the leader. After confirming that the unique acknowledge number equals to the number of assistants, the leader sends a "done" message to the sink.

3. *Synchronization.* When the sink receives all the "done" messages, it sends out the synchronization command, and all the nodes begin to do synchronization by using reference-broadcast method [17]. Right after the synchronization, the nodes enter the slot-based stage and will switch channels according to the results obtained in step 2.

3.6 Number of Channels Used

In order to prevent interference between different hexagon cells from happening, the neighboring cells use different channels. Inside each cell, the same channel is used. Since the six inside triangles will use the channel at different time slots, there is no interference between them. According to four color map theorem [18], at least four channels should be used for arbitrary shapes. In our scenario with hexagon shape, at least three channel should be used in our system. In such case, supposing in a 4 m triangle, at one time slot, the distance between two triangles with the same channel will be at least over 8 m apart. The interference will be dramatically decreased.

Furthermore, if we would like to avoid interference as much as possible, we may use as many number of different channels as possible. In our experiments, TelosB nodes are based on 2.4 GHz band offering up to 16 channels. So, it is sufficient for us to allocate.

4 Performance

In this section, first we show our experimental setup and phases of the tracking system. Second, the investigation of different frequency and different triangles settings are given. Third, we describe the performance of our tracking system by analyzing its latency, tracking accuracy, cost of deployment, and scalability. At last, the tests of one moving object and multiple objects are provided.

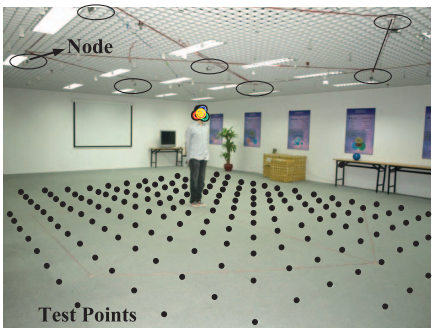


Fig. 11. Test samples in the real environment, each black point represents one tested object location.

4.1 Experiment Setup

Our experiment is conducted in an empty room with 20×20 square meters as shown in Fig. 11. We use 10 popular TelosB sensor nodes with Chipcon $CC2420$ radio chips to set up two adjacent cells on the ceiling of the floor. Each cell is a hexagon containing six regular triangles. In each triangle, the node distance of each triangle is set as 4 m unless otherwise specified. The default transmission power is set as 0 dBm. The radio frequency band we choose from $2{,}400$ to 2483.5 MHz. We program each node to broadcast beacons at an assigned channel in a fixed time slot, as the procedure presented in the last section.

Our RASS system has two phases. The first is the offline training phase. A number of RSS dynamic samples are collected based on different object position. In this phase, the parameters of SVM model is confirmed. The second phase is the online tracking phase. In general, it has two subphases. First, in the pretracking phase, each node will build a static table to store the static RSS values for all its neighbors in the same channel. The initialization phase has to be carried out in the static environment. After all the nodes have built up such tables and the entire triangle areas have been swept at least once, the system enters the tracking subphase. Each node measures the RSS dynamic value from different neighbors in the same channel. If the RSS dynamic value on a link is higher than some link threshold (this value is defined as the RSS dynamics in the static environment [10], as there is still some very small RSS difference even in the static environment), the RSS dynamic value is reported back to the sink node. Otherwise, the dynamic value is used to update the static table. The server connecting to the sink is responsible to calculate the object position.

If the environment changes, users do not have to perform training again. They only need to collect three RSS dynamic samples for $m1$, $m2$, and $m3$ positions (depicted in Fig. 7) of each triangle. The RSS dynamic samples of the other places are all generated by interpolation. The new SVR model can be achieved by using the adaptive learning method based on the new generated data, as introduced in Section 3.4.

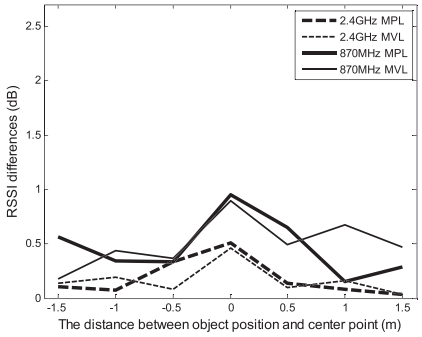


Fig. 12. Impact of different frequency for 2 m node distance.

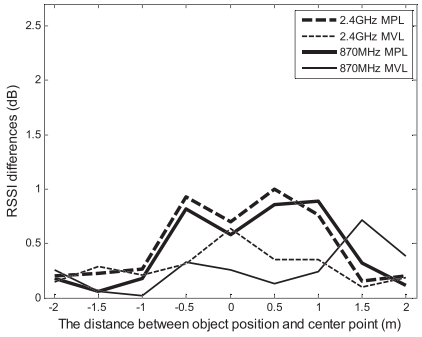


Fig. 13. Impact of different frequency for 3 m node distance.

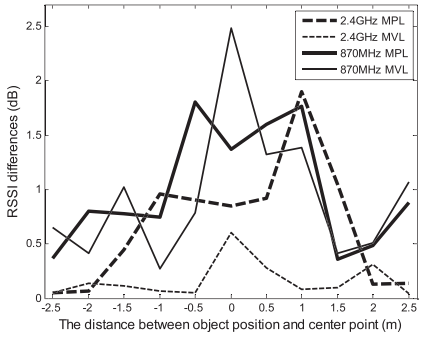


Fig. 14. Impact of different frequency for 4 m node distance.

4.2 Impact of Different Frequencies

Different frequencies may cause different RSS dynamic behavior even for the same object location in the same node deployment. In order to find out its influence, we utilize just two communicating nodes to test different frequency bands. Here, we choose two different frequency bands: 870 MHz and 2.4 GHz. The former frequency band is tested on Mica2 [19] sensor nodes and latter one is tested on TelosB [13] sensors nodes.

Figs. 12, 13, and 14 plot the RSS dynamics caused by different object positions with different node distances. From these figures, we find out no matter what frequency band we use, the RSS dynamics are all bumped up around the midpoint of MPL and MVL.

But, they still have some differences. For example, in Fig. 12, when the frequency is 2.4 GHz, the RSS dynamics reach nearly zero as the object goes further from the midpoint of MPL and MVL. It never happens when the frequency is 870 MHz. Although the RSS dynamics of 870 MHz is a little bit higher, it is not stable as the object is far away.

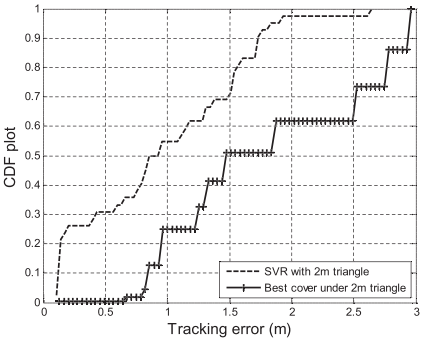


Fig. 15. Algorithm comparison on 2 m triangle.

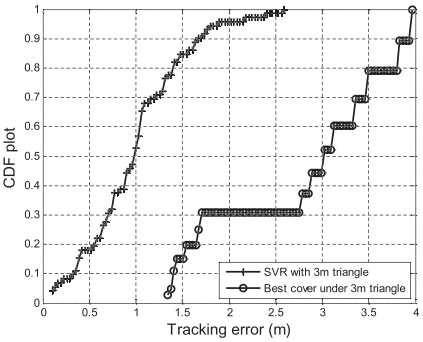


Fig. 16. Algorithm comparison on 3 m triangle.

Our measurements lead us conclude that the RSS dynamics are more sensitive when the frequency is high. More sensitive RSS dynamics bring us benefit for estimating the object location. Therefore, we choose the 2.4 GHz frequency band in our later experiment.

4.3 Impact of the Triangle Size

Since the triangle is the basic component in our node deployment, how to choose a suitable node distance is an important issue. Moreover, the accuracy of position estimation depends on the size of the triangle.

We test from two different aspects. First, we test just one pair of communicating nodes. This work was done before [10] but with 870 MHz band. In 2.4 GHz band, the result is similar. We find that when the node distance is between 2 and 4 m, the RSS dynamics grow larger as the object is closer to the midpoint of MPL and MVL. We call these as valid distance. For other node distances such as smaller than 1 m or greater than 5 m, this trend is not obvious. If the node distance is very short, the received signal strength is very strong on the line-of-sight radio propagation path and it is not easily influenced by the scatted wave caused by the object. On the contrary, if the node distance is very large, the received signal strength is very weak and is susceptible to noise interference.

Second, based on the valid link range measured above, we further test on the 2, 3, and 4 m triangles with three communicating nodes, as shown in Figs. 15, 16, and 17. We find that as the node distance grows larger, there is only a little difference from their tracking errors by using SVR. Based on predicting 42, 72, and 110 object locations, their average tracking errors are 0.98, 1.01, and 1.13 m for the 2, 3, and 4 m triangles, respectively. The results are much better

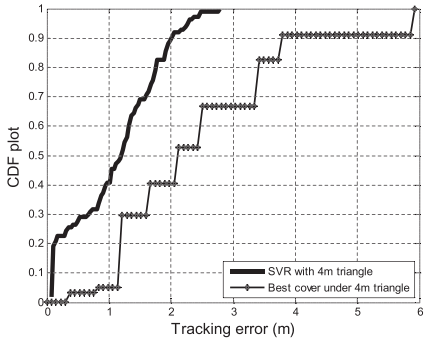


Fig. 17. Algorithm comparison on $4\ \mathrm{m}$ triangle.

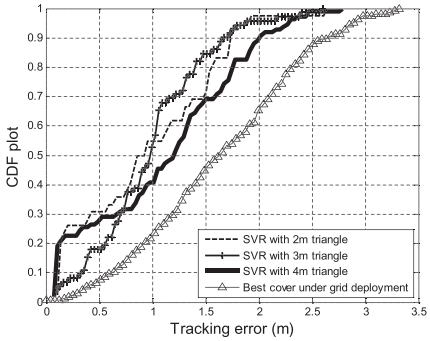


Fig. 18. Tracking error based on different triangle size and comparison with best cover algorithm under grid setting.

than the tracking error if we use the best cover algorithm [10] under such settings.

To sum up, the testing on different node distances gives us different options. Since the tracking errors of different settings are all around 1 m, if the users are not very strict with the slightly difference of the tracking error and want to save the cost, they may choose the 4 m triangle. Hence, fewer nodes will be deployed. On the contrary, if the users would like to get higher accuracy, they may choose the 2 m triangle to deploy. It will also bring more benefit to locate multiple objects, because the covering area of the 2 m triangle is less than that of the 3 and 4 m triangles. Only when different objects are in different triangles, we may easily locate them.

4.4 Latency

The latency of our RASS tracking system depends on how much time for one triangle to finish sweeping its belonging cell, since the tracking policy for each cell is independent of other cells according to our channel assignment scheme.

Within one triangle area to locate an object, there are three links. We want to collect enough information for each of them. For one transmission direction of each link (one link contains two opposite transmission directions for the two nodes), we need to collect one RSS dynamic value once, which requires one packet transmission time. From our study [20], a TelosB sensor node takes $7\ \mathrm{ms}$ on average to transmit a packet with $51\ \mathrm{bytes}$. Also, our previous study revealed that the channel switching costs $0.34\ \mathrm{ms}$ each time. Thus, one slot should be at least $(7 + 0.34) \times 3 \times 2 \approx 44\ \mathrm{ms}$ long. Also, one hexagon cell area contains six triangles. It requires six time slot to sweep all the area. Therefore, in total, it needs $44 \times 6 = 264\ \mathrm{ms} \approx 0.26\ \mathrm{s}$. The total latency T_d can be expressed as follows:

$$T_d = (T_{switch} + T_{trans} \cdot 2 \cdot N_{link} \cdot N_{tri}). \tag{16}$$

Here, T_{switch} is the channel switch time for each node. T_{trans} is the time for one node to transmit a packet. N_{link} is the number of links in one triangle. N_{tri} is the number of triangles in one triangle hexagon area.

In summary, our system can reach the real-time tracking latency to as fast as about $0.26\ \mathrm{s}$, which can satisfy most emergent requirements. On the contrary, our previous best cover algorithm [10] under a node grid requires $3\ \mathrm{s}$ to get response, even the Distributed Dynamic Clustering [11] method needs $2\ \mathrm{s}$ delay for localization. So, the latency of our RASS tracking system significantly outperforms previous tracking systems.

4.5 Comparative Study on Accuracy

Tracking error is an important performance matrix for tracking systems. In our experiments, we test different triangle sizes with 2, 3, and 4 m node distances. Based on 42, 72, and 110 tested object positions for each triangle, the tracking error of using SVR and comparisons with the previous algorithm are given in Fig. 18. We may see that the average accuracy of our RASS algorithm is 0.98, 1.01 and $1.13\ \mathrm{m}$, respectively. It greatly outperforms the previous best cover algorithm. The reason is that the accuracy of latter algorithm is based the total number of wireless links. It has to use a node grid, e.g., 16 nodes, which have 240 wireless links. But in a system with few nodes, its accuracy will decrease dramatically. Our SVR has no such limitation. Therefore, we can get similar accuracy even we use a few nodes.

Therefore, as the size of the triangle grows larger within the node distance limitation from 2 to 4 m, the tracking error of our RASS system is still around 1 m.

The experiment results include the object positions inside and outside one triangle. In order to further investigate their respective influences on the tracking accuracy, we separate them for discussion. For the object positions which are inside one triangle, we find out that 91 percent of the calculated locations by SVR are also inside the triangle, as shown in Fig. 20a. For the other object positions which are outside the triangle, we find out that 82 percent of the calculated locations by SVR are also outside the triangle, as shown in Fig. 20b. It means that for most object locations, we can decide it in the right triangle. We show it in Table 2.

It is concluded that for each object to be tracked, it has high probability to be recognized inside the right triangle. As Fig. 19 depicts, no matter based on 2, 3, or 4 m triangles, 70 percent of the tracking errors is under 1 m for the inside triangle object. As a result, if two or more adjacent triangles all estimate one target object inside them at the same time, we average their coordinates as the output object location. Such case often happens when the object is around the border area connecting many adjacent triangles. If only one triangle senses the object inside it, we just output it.

TABLE 2
Object Inside/Outside One Triangle Decision

	Real object position inside the triangle	Real object position outside the triangle
Calculated position inside the triangle	91%	18%
Calculated position outside the triangle	9%	82%

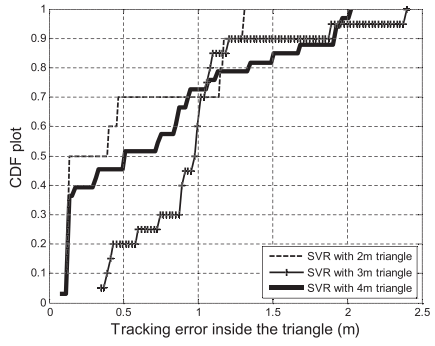


Fig. 19. Tracking error when object is inside triangle.

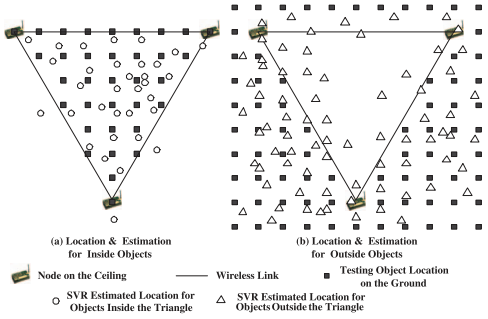


Fig. 20. Object locations and estimation by SVR.

4.6 Cost of Deployment

As introduced before, our RASS tracking system can give users free choices with the deployment. If they do not care the slight difference of the tracking accuracy, they may choose the 4 m triangle setting to save deployment cost. Its tracking accuracy still can reach around 1 m, but very few nodes need to be deployed. Otherwise, they may choose the 2 m triangle setting.

For example, as shown in Fig. 21, suppose we are in an 88 square meter tracking field. Traditional best cover algorithm requires 25 nodes deployed in this area. But our RASS system only needs at least seven (4 m triangle) and at most 23 (2 m triangle) to reach a similar accuracy with significantly improved latency. Therefore, our RASS tracking system can save from 8 up to 72 percent deployment cost in total.

4.7 Tracking Moving Objects

Our RASS tracking system has a good ability to track moving objects because the time for one triangle area to finish sweeping one cell is only 0.26 s. The same is true for the whole tracking field. The time for the central computer to predict the object position by SVR can be almost

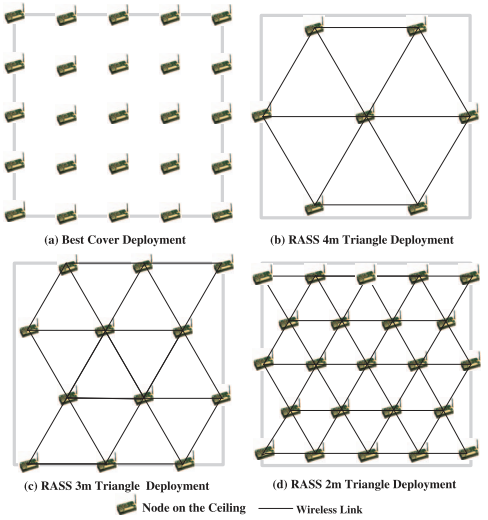


Fig. 21. Cost of deployment.

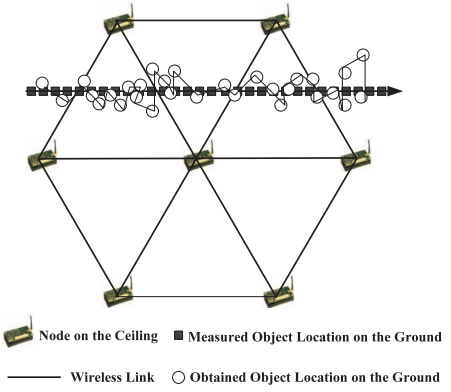


Fig. 22. Moving object tracking.

negligible. So, every 0.26 s, we have a report for the object location. Therefore, it is able to support fast moving objects.

To study the effect of a single moving object, we arrange a person to walk through a fixed trace under the cells. Here, we choose the 4 m triangle setting inside each cell. The person's moving speed is around 1 m/s. One of the testing trace is in Fig. 22. The average tracking error of this example is 0.87 m.

4.8 Multiple Objects

Our RASS system is able to track multiple objects, as long as they are in different triangles. As the size of the rectangle can be chosen by different users, deploying small-size triangles can easily separate different objects. Our experiment shows the smallest node distance of the triangle is 2 m. And we can reach the tracking accuracy of each object to 0.98 m in average.

Therefore, if the deployment cost is not the first issue and we care more about the tracking accuracy of multiple objects, 2 m triangle deployment is recommended. The area of the 2 m triangle is about 1.73 square meters. As a result, only if the locations of the multiple objects are beyond this limit, we may locate them easily, e.g., target objects 2 and 3 in Fig. 23. Otherwise, we just regard them as one big object. As Fig. 23 shows, target objects 1 and 1′ are in the same triangle. We may regard them as one triangle.

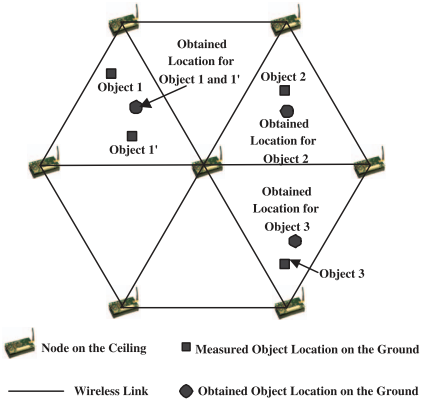


Fig. 23. Multiple objects.

4.9 Scalability

Our RASS system scales well in large area in theory. Since we use multiple channels for different triangle, the tracking procedure for each triangle is relatively independent. As a result, the scalability property is well satisfied. In theory, there is no limitation for the field size of the deployment. In the mean time, the latency will not be sacrificed. Moreover, since the distance between different triangles with the same channel could be very long (it depends on how many used channels and the size of the triangle), the interference of the remote overlapping channels will be very small. Thus, the interference effect to the accuracy has the high probability to be small.

5 Conclusion and Future Work

In this paper, we have presented a novel tracking system RASS, which has the ability to track transceiver-free objects in real time. We are the first to introduce multiple channels into this topic and bring many benefits to the system. Our basic idea is to divide the tracking field into different triangle areas. Each triangle will use different channels at different time slots. Thus, we can avoid the interference among nodes belonging to different channels. Moreover, we only require considering the wireless communication of nodes in the same channel. As a result, the beacon interval of each node to avoid transmission collision can be very short. For each triangle area, we utilize an SVR model to estimate the object position, which can get good tracking accuracy with only three communicating nodes. The model is to find a regression function to simulate the relationship between the RSS dynamics caused by the object and the object location. Our model is adaptive and easy to be retrained by only using very few samples. The other data can be obtained from the old samples. The RASS tracking accuracy is around 1 m with the latency only about 0.26 s. Even for multiple objects, as long as they are in different triangles, we may locate each of them. If they are very close to each other, we just regard them as one object. At last, our system is well scaled in a large deployment without sacrificing the latency and accuracy.

As future work, we would like to try a larger area or with different topologies to deploy nodes. This may help improve the accuracy of location estimation. Our solution to multiple moving objects is limited by the size of the rectangles. Also, if the multiple objects are all in the border area, the tracking error will increase. Better model still need to be found for the objects which are very close together. Furthermore, we would also like to try some probabilistic models to analyze the moving trace of multiple objects in the future.

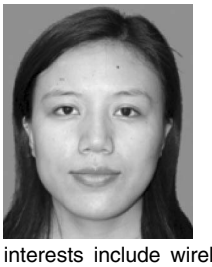
Acknowledgments

This research was supported in part by Hong Kong RGC Grant HKUST617710, China NSFC Grants 60933011, 60933012, 60673122, 61170077, 61170076, 61033009, 61103001, 61103272 the Science and Technology Planning Project of Guangdong Province, China under Grant 2009A080207002, the SZ-HK Innovation Circle Project under Grant ZYB200907060012A, China NSFGD Grant 10351806001000000, and the Science and Technology Project of Shenzhen JC200903120046A.

References

[1] G. Xu, *GPS: Theory, Algorithms and Applications.* Springer-Verlag, 2003.
[2] T. Gao, D. Greenspan, M. Welsh, R.R. Juang, and A. Alm, "Vital Signs Monitoring and Patient Tracking over a Wireless Network," *Proc. 27th IEEE Eng. in Medicine and Biology Soc. (EMBS '05),* 2005.
[3] P. Zhang, C.M. Sadler, S.A. Lyon, and M. Martonosi, "Hardware Design Experiences in ZebraNet," *Proc. ACM Conf. Embedded Networked Sensor Systems (SenSys '04),* 2004.
[4] S.N. Patel, J.A. Kientz, G.R. Hayes, S. Bhat, and G.D. Abowd, "Farther than You May Think: An Empirical Investigation of the Proximity of Users to Their Mobile Phones," *Proc. ACM Int'l Conf. Ubiquitous Computing (UbiComp '06),* 2006.
[5] "ACOREL Corporation,"People Counting Technology Using in Infrared, http://www.acorel.com, 1989.
[6] J.O. Robert and D.A. Gregory, "The Smart Floor: A Mechanism for Natural User Identification and Tracking," *Proc. ACM Conf. Human Factors in Computing Systems (CHI '00),* 2000.
[7] Q. Cai and J.K. Aggarwa, "Automatic Tracking of Human Motion in Indoor Scenes across Multiple Synchronized Video Streams," *Proc. Sixth Conf. IEEE Computer Vision and Pattern Recognition (CVPR '98),* 1998.
[8] R. Dorsch, G. Hausler, and J. Herrmann, "Laser Triangulation: Fundamental Uncertainty in Distance Measurement," *Applied OPTICS,* vol. 33, 1994.
[9] M. Youssef, M. Mah, and A. Agrawala, "Challenges: Device-Free Passive Localization for Wireless Environments," *Proc. ACM MobiCom '07,* 2007.
[10] D. Zhang, J. Ma, Q. Chen, and L.M. Ni, "An RF-Based System for Tracking Transceiver-Free Objects," *Proc. Fifth Ann. IEEE Int'l Conf. Pervasive Computing and Comm. (PerCom '07),* 2007.
[11] D. Zhang and L.M. Ni, "Dynamic Clustering for Tracking Multiple Transceiver-Free Objects," *Proc. Seventh Ann. IEEE Int'l Conf. Pervasive Computing and Comm. (PerCom '09),* 2009.
[12] Q. Chen, M. Gao, J. Ma, D. Zhang, L.M. Ni, and Y. Liu, "MOCUS: Moving Object Counting Using Ultrasonic Sensor Networks," *Int'l J. Sensor Networks,* vol. 3, pp. 55-65, 2008.
[13] "XBOW Corporation,"TelosB Mote Specifications, http://www.xbow.com/Products/productdetails.aspx?sid=252, 2005.
[14] A.J. Smola and B. Scholkopf, "A Tutorial on Support Vector Regression," *Statistics and Computing,* vol. 14, no. 3, pp. 199-222, 2004.
[15] "LIBSVM,"Library to Using SVM, http://www.csie.ntu.edu.tw/cjlin/libsvm/, 2012.
[16] S.M. Nazrul Alam and Z.J. Haas, "Coverage and Connectivity in Three-Dimensional Networks," *Proc. 12th ACM MobiCom '06,* 2006.
[17] J. Elson, L. Girod, and D. Estrin, "Fine-Grained Network Time Synchronization Using Reference Broadcasts," *Proc. Fifth Symp. Operating Systems Design and Implementation (OSDI '02),* 2002.

[18] "FourColorMap,"Math World, http://mathworld.wolfram.com/Four-ColorTheorem.html, 2012.
[19] "XBOW Corporation,"XBOW MICA2 Mote Specifications, http://www.xbow.com/Products/productdetails.aspx?sid=174, 2001.
[20] Y. Yang, Y. Liu, and L.M. Ni, "Level the Buffer Wall: Fair Channel Assignment in Wireless Sensor Networks," technical report, HKUST, 2008.
[21] L.M. Ni, Y. Liu, Y.C. Lau, and A.P. Patil, "LANDMARC: Indoor Location Sensing Using Active RFID," *Proc. First IEEE Int'l Conf. Pervasive Computing and Comm. (PerCom '03),* 2003.
[22] P. Bahl and V.N. Padmanabhan, "RADAR: An In-Building RF Based User Location and Tracking System," *Proc. 19th IEEE INFOCOM '00,* pp. 8-16, 2000.
[23] M. Moussa and M. Youssef, "Smart Devices for Smart Environments: Device-Free Passive Detection in Real Environments," *Proc. Seventh Ann. IEEE Int'l Conf. Pervasive Computing and Comm. (PerCom '09),* 2009.
[24] Y. Liu, L. Chen, J. Pei, Q. Chen, and Y. Zhao, "Mining Frequent Trajectory Patterns for Activity Monitoring Using Radio Frequency Tag Arrays," *Proc. Fifth Ann. IEEE Int'l Conf. Pervasive Computing and Comm. (PerCom '07),* 2007.
[25] X. Mao, S. Tang, X. Xu, X. Li, and H. Ma, "iLight: Indoor Device-Free Passive Tracking Using Wireless Sensor Networks," *Proc. IEEE INFOCOM '11,* 2011.
[26] F. Viani, L. Lizzi, P. Rocca, M. Benedetti, M. Donelli, and A. Massa, "Object Tracking through RSSI Measurements in Wireless Sensor Networks," *Electronics Letters,* vol. 44, no. 10, pp. 653-654, 2008.



Dian Zhang received the PhD degree in computer science and engineering from the Hong Kong university of Science and Technology, Hong Kong, in 2010. She was a postdoctoral fellow and later a research assistant professor of Fok Ying Tung Graduate School, the Hong Kong University of Science and Technology, from 2010 to 2011. She is now a lecturer in the College of Computer Science and Software Engineering, Shenzhen University. Her research interests include wireless sensor networks, pervasive computing and Networking and Distributed Systems. She is a member of the IEEE.

Yunhuai Liu received the BS degree from the Computer Science and Technology Department of Tsinghua University in July 2000. He received the PhD degree from the Computer Science Department, the Hong Kong University of Science and Technology in 2008. He is now a professor of the Third Research Institute of the Ministry of Public Security. He is a member of the IEEE.

Xiaonan Guo is currently working toward the PhD degree in the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology. His research interests include wireless sensor network and mobile computing. He is a member of the IEEE.

Lionel M. Ni received the PhD degree in electrical and computer engineering from Purdue University, West Lafayette, IN, in 1980. He is a chair professor and was the head of Computer Science and Engineering Department at the Hong Kong University of Science and Technology. He is also a visiting chair professor at Shanghai Jiaotong University. His research interests include parallel architectures, distributed systems, wireless sensor networks, high-speed networks and pervasive computing. He has chaired many professional conferences and has received a number of awards for authoring outstanding papers. He is a fellow of the IEEE.