中央研究院
資訊科學研究所
Institute of Information Science, Academia Sinica • Taipei, Taiwan, ROC

TR-IIS-20-001

Methods for Determining Indoor Positions of Tracked Objects

C. -Y. Lai, S.-Q. Zhou and J. W. S. Liu

May 19, 2020 || Technical Report No. TR-IIS-20-001
http://www.iis.sinica.edu.tw/page/library/TechReport/tr2020/tr20.html

Institute of Information Science, Academia Sinica

Technical Report TR-IIS-20-001

Methods for Determining Indoor Positions of Tracked Objects

C. -Y. Lai, S.-Q. Zhou and J. W. S. Liu
Institute of Information Science, Academia Sinica
Taipei, Taiwan
{jimmylai, janeliu}@iis.sinica.edu.tw

Institute of Information Science, Academia Sinica

Technical Report TR-IIS-20-001

Methods for Determining Indoor Positions of Tracked Objects

C. -Y. Lai, S.-Q. Zhou and J. W. S. Liu
Institute of Information Science, Academia Sinica
Taipei, Taiwan
{jimmylai, janeliu}@iis.sinica.edu.tw

Abstract

Fingerprint lookup and proximity detection are positioning technologies commonly used by indoor positioning and indoor object tracking systems (IOTS) to locate objects and track their movements. Ideally, every fingerprint captured by every device is mapped to the cell containing its current location, and the UUID of every tag is detected by the proximity detector closest to its location. This enables the IOTS to return the *correct location* of the device/tag. As long as the device/tag does not move, a graphical user interface (GUI) of the IOTS shows it at the location. When the device/tag moves, the GUI shows its new correct location after a short delay. In real-life operating environments, many factors, including the presence of objects in signal paths, interferences from emitters nearby, natural fluctuations in transmitted power, and so on, can cause unpredictable fluctuations in received signal strengths. Consequently, the GUI may show the object moving sporadically even when the object is standing still or moving haphazardly when it actually moves along a well defined trajectory. Such apparent changes in object positions returned by an IOTS due to unpredictable and sporadic changes in signal strengths is referred to as noisy movements or false movements. Users are likely to find noisy/false movements annoying and the location information confusing. The techniques described in this report aim to eliminate or reduce noisy/false movements and improve the accuracy of locations returned by IOTS.

Keywords: Indoor object tracking; Location beacons; Indoor positioning; Proximity detection; fingerprint-based systems

1. Background and Motivation

Real-time indoor location-based services (RT-ILBS) in general, and *indoor positioning and indoor navigation* (*IPIN*) and indoor object tracking applications and services (IOTS) in particular, need to determine sufficiently accurately within buildings of all sizes and characteristics and under various operating conditions locations of mobile devices, including smart phones, tablets, and wireless tags. Systems that provide this function are called *indoor positioning systems* (*IPS*).

(A) Indoor Positioning Techniques

Fingerprint-based and proximity-based techniques are often used by IPS for this purpose:

- *Fingerprint-based schemes*: A *fingerprint* is a set of location-specific values of signal strength (i.e. a signal pattern). Types of fingerprints include patterns of WiFi, FM and Bluetooth signals, acoustic echo patterns and background spectrum, and magnetic signatures of the building [1-6]. A fingerprint-based IPS has a database of fingerprints captured at different locations in the building during setup and maintenance times and a location/fingerprint server. Figure 1(a) illustrates a practical embodiment of such an IPS that uses magnetic signatures as fingerprints. The indoor space is partitioned into cells. The database contains fingerprint-to-cell mappings. To determine its own location, a mobile device sends the fingerprint captured by it at its location to the server and relies on the server to find in the database the cell(s)/location(s) with matching fingerprint(s). The location accuracy provided by such a system is the size of the cells.
- Proximity-based schemes: Proximity detection is also a common approach to indoor positioning. Short-range transmitters such as RFID tags, ibeacons from Apple and Eddystones from Googles may be used to provide mobile devices data (e.g., UUID of the beacons or tags or a URL associated with each beacon) using which the devices may query for their locations. Figure 1(b) illustrates an embodiment that uses *location beacons* (*Lbeacons*) [7, 8] for this purpose. Lbeacons are low-cost Bluetooth devices with directional antennas. They are installed to provide coverage throughout the building and networked together with a server. Each Lbeacon stores its own 3-D coordinates and broadcasts its coordinates to devices under its coverage. Location accuracy is determined by the angles of their conical radiation beams and thus the size of their coverage areas. Indoor navigation applications running on mobile devices can generate navigation instructions based on location data from Lbeacons.

A system of Lbeacons can also support object tracking applications. Figure 1(b) illustrates how a Bluetooth indoor object tracking system (IOTS) works. Each object to be tracked carries a Bluetooth low-energy device, called *tag*. In addition to broadcasting, Lbeacons can capture the MAC addresses of tags contained in the advertizing data packets broadcast by the tags. Each Lbeacon timestamps every address its captures and forwards the time stamped address to the server together with its own coordinates. Since the MAC address of every Bluetooth device is unique, a system of Lbeacons can locate every tag, and thus the object (e.g., a person, or a wheel chair, or a medical device) carrying the tag, and track the movement of the object over time as long as the tag is under the coverage of some Lbeacon in the building.

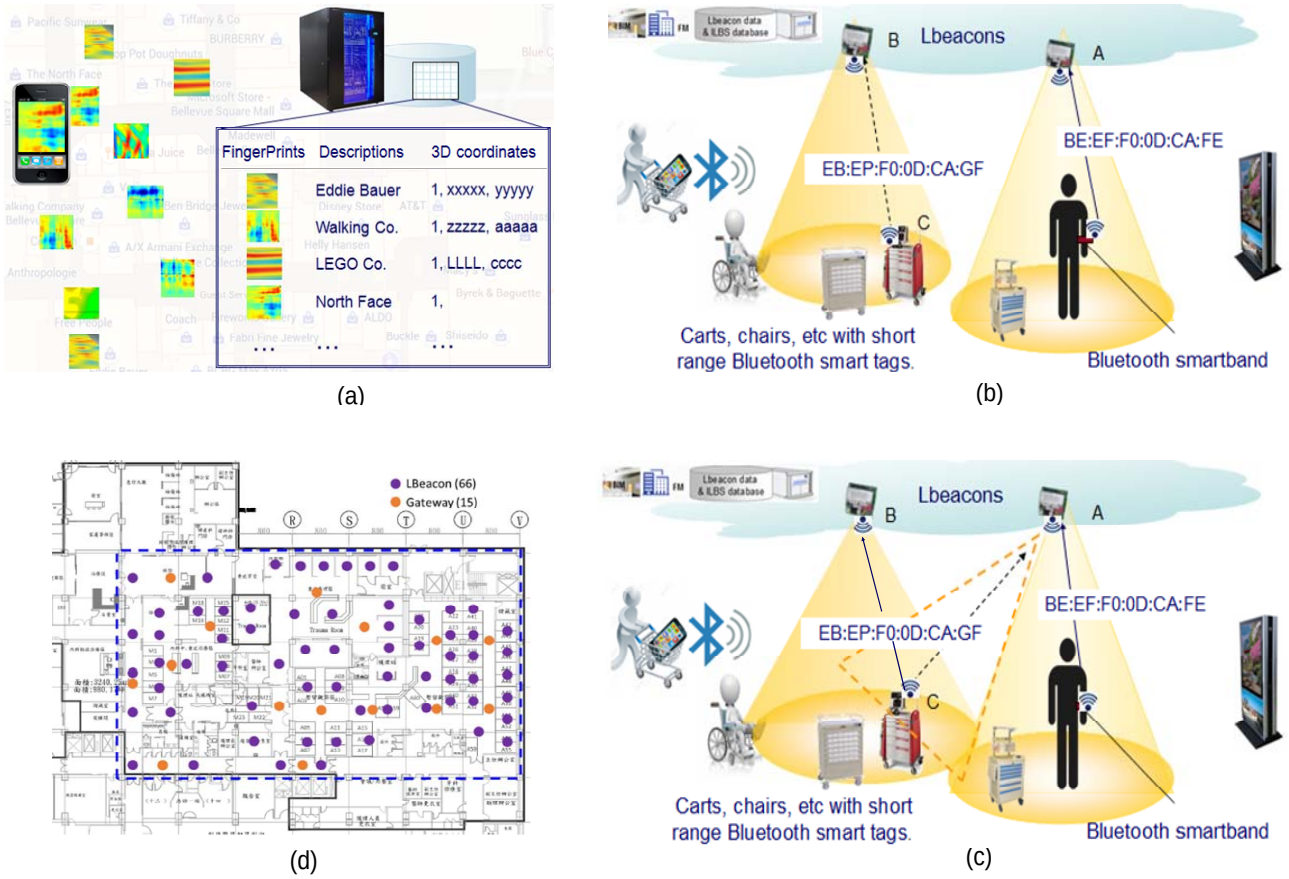

Figure 1 Examples of embodiments of IPS

(B) The Noisy/False Movement Problem and Objectives of Its Solutions

Figure 1(a) and (b) show the ideal condition: Every fingerprint captured by every device is mapped to the cell containing the current location of the device, and the advertizing packet containing the MAC address of every tag is received by only one Lbeacon among all Lbeacons and the current location of the tag is in the coverage area of the Lbeacon. This enables the IPS to return the *correct location* of the device/tag: that is, the location of the matching cell center or a location within the cell in case of a fingerprint-based system, or in case of a system of Lbeacons, a location within the coverage area of the Lbeacon. As long as the tag does not move, a graphical user interface (GUI) of the IOTS continues to show it at the location. When the tag moves, the GUI shows its new correct location after a short delay.

Under real-life operating conditions, the strength of a received signal depends not only on the distance from the signal source. Many factors, including the presence of objects in the signal path, interferences from emitters nearby, natural fluctuations in transmitted power, variations in time of transmissions, and so on, can cause unpredictable fluctuations in received signal strength. Figure 1(c) illustrates the possibility that one or more nearby Lbeacons may also receive advertising packets, and hence the MAC addresses, broadcast by tags located in the coverage areas of a Lbeacon. Despite a longer signal path, their received signal strengths may be stronger. This is especially likely to happen when Lbeacons are densely deployed as illustrated by Figure 1(d) where

Lbeacons are placed closely to achieve bed-level location accuracy (< 3m) in an emergency room. – Similarly, the cell returned by a fingerprint-based system in response to a device's query may change when there are unpredictable variations in the fingerprints captured by the device. Based on such data, the GUI of an IPS or an object tracking system may show the object moving around sporadically every few seconds or a few tens of seconds even when the object is standing still or show the object moving haphazardly when it actually moves along a well defined trajectory.

Hereafter, the apparent changes in object positions returned by an indoor object tracking system (IOTS) due to unpredictable and sporadic changes in signal strengths will be referred to as *noisy movements* or *false movements*. Users are likely to find noisy/false movements displayed by a GUI annoying. Worst yet, they may present the user with confusing and incorrect information. The methods described here aims to address this problem. Specifically they aim to

1. Improves the accuracy of locations returned by the IPS and IOTS, and
2. Eliminate or reduce noisy/false movements of objects presented by GUI.

Following this introduction, Section 2 presents as a concrete example of IOTS implemented with Lbeacons and uses it to illustrate the object location problem addressed here. Sections 3 and 4 describe alternative methods to determine the locations of objects tracked by such a system and methods for elimination of false movements, respectively. Section 5 discusses how the solutions for Lbeacon-based systems can be generalized and applied to fingerprint-based systems. Section 6 summarizes the report.

2. A Motivating Example Based on an Embodiment of IOTS

To illustrate the problem and motivate the methods to be presented in subsequent sections, this section first describes the characteristics of advertising data packets in a system of Lbeacons, using data collected from the IOTS shown Figure 1(d) for illustration purpose. The system makes decisions on the locations of objects based on these characteristics. This section describes how decision support data used to determine locations of tracked object presented to users by the GUI are generated from these characteristics.

(A) Illustrative examples of observed data

The upper half of Figure 2 shows plots of *observed RSSI* (received signal strength indices) of advertizing packets (or simply packets) broadcast by a tag at locations p11 and p12. The map is that of the upper right hand corner of the room shown in Figure 1(d). A01, A02, … A10 and A11 are names of Lbeacons around p11 and p12. Each dot in the plots for a location above the name of an Lbeacon indicates the observed RSSI value of a packet captured by the Lbeacon when the tag is at that location. The scattered plot was obtained by keeping a tag at each location and recording the RSSI's of its data packets received by each of the Lbeacons within a given *observation window, i.e.,* the time interval during which packets from a tag was received and their RSSI's was measured by each Lbeacon. In other words, each vertical cluster of dots is a *summary statistics* [9] that describes the RSSI's of packets from a tag at a location as observed by a Lbeacon.

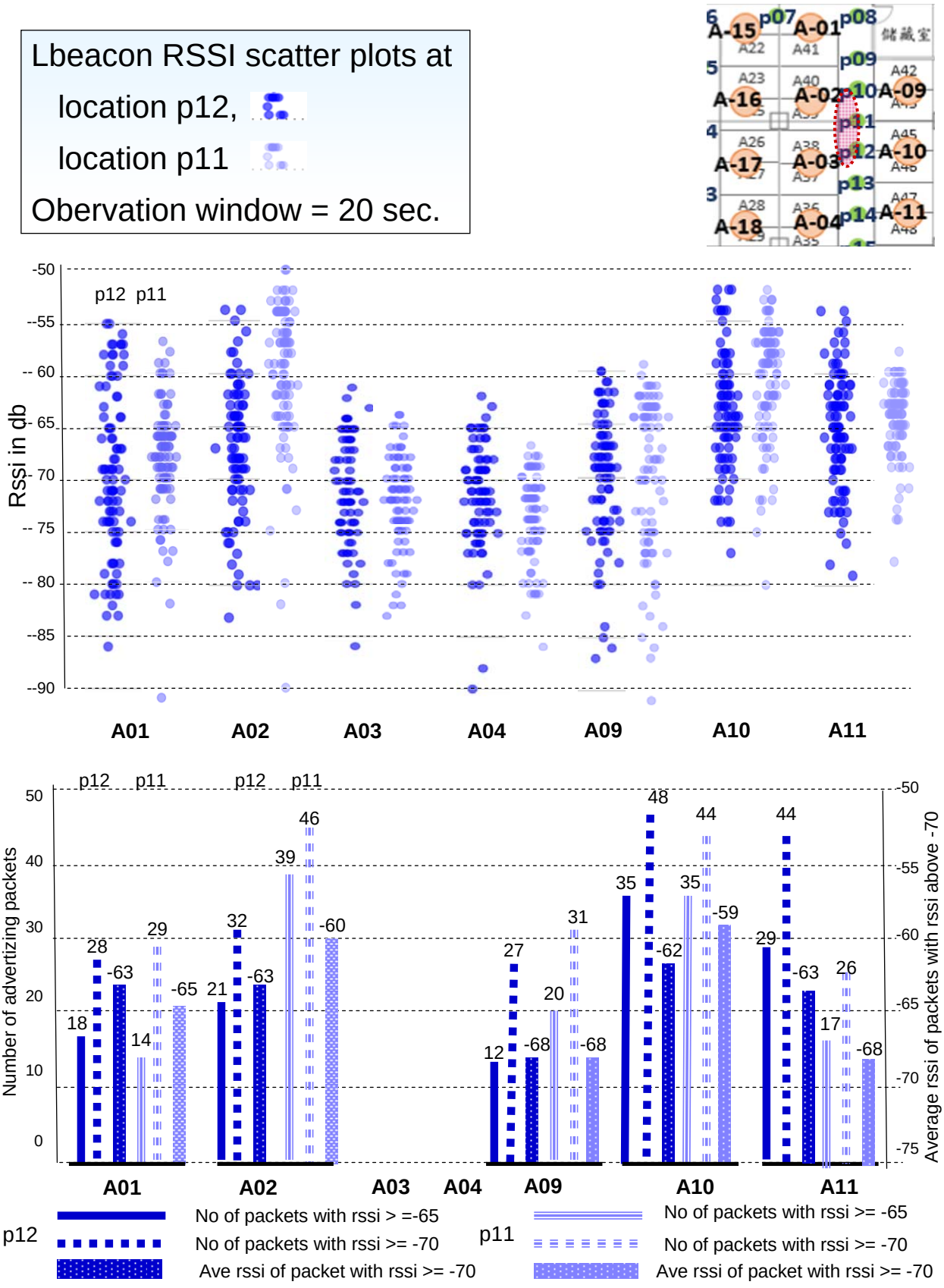

Figure 2 RSSI scatter plots and sample values - in densely deployed area

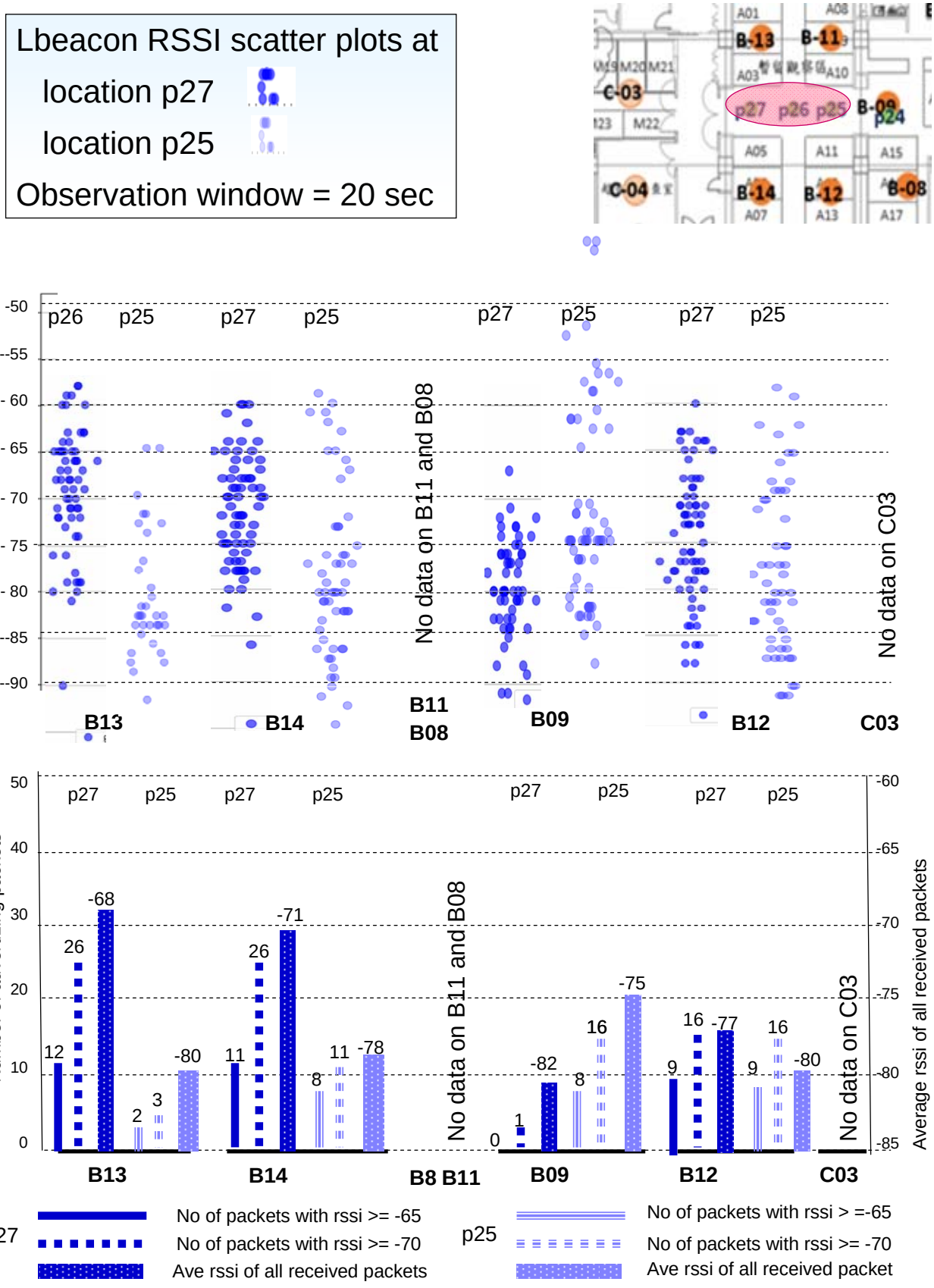

Figure 3 RSSI scatter plots and sample values - in sparsely deployed area

Similarly, the upper half of Figure 3 shows the summary statistics of RSSI's of packets received by Lbeacons B13, B14, B09, and B12 that are around locations p25 and p27 when the tag is at these locations. For both plots, the size of observation window is 120 seconds. The longer the window, the more reliable is the observation, but the slower the response of the system.

(B) Test Statistics

The bottom halves of Figures 2 and 3 show examples of *test statistics* [10] that are computed from the corresponding summary statistics. Decisions on locations of the tag are made on the basis of these and other test statistics. Examples shown here include numbers of packets with RSSI's higher than specified thresholds (e.g., -65 db and -70db) received by a Lbeacon within an observation window and average RSSI of all received data packets (in Figure 3) or of received packets with RSSI higher than a threshold (e.g., -70 db in Figure 2). They are natural choices of test statistics for this type of observation data.

Formally, these test statistics are denoted by

- N (T, W; L): Number N of packets with RSSI's higher than the threshold T captured by Lbeacon L within an observation window of length W
- R (T, W; L): Average value of RSSI's of packets with RSSI higher than the threshold T captured by Lbeacon L within an observation window of length W

In the special case where no threshold is chosen (i.e., $T = -\infty$), these notations become N (W; L) and R (W; L), respectively, denoting the total number of packets and the average RSSI's of all packets captured by Lbeacon L within window W, or simply N and R when there is no need to be specific. Table 1 lists sample values of these test statistics for the data shown in Figures 2 and 3.

Table 1 Sample values of test statistics

Li	*N* (-70, 120; *Li*)		*R* (-70, 120; *Li*)	
	p12	p11	p12	p11
A01	28	29	-63	-65
A02	32	46	-63	-60
A09	27	31	-68	-68
A10	48	44	-62	-59
A11	44	26	-63	-68

Li	*N* (-70, 120; *Li*)		*R* (120; *Li*)	
	p27	p25	p27	p25
B13	26	3	-68	-80
B14	26	11	-71	-78
B09	1	16	-82	-75
B12	16	16	-77	-80
C03	No data			

3. Decision and Estimation Rules of Determining Object Locations

At any time, the IOTS makes a decision on the location, or computes an estimate of the location, of a tag (i.e., the object affixed with the tag) based on the sample values of a test statistics according to a *decision/estimation rule.* There are two types of rules: Location-ignorant rules and location-aware rules. A *location-ignorant rule* decides on the location of an object based on

available sample values of one or more test statistics without taking into account the locations of Lbeacons that collected the samples. In contrast, a *location-aware* rule estimates the location of the object based not only on sample values but also on locations of the Lbeacons.

Location-ignorant decision rules Below are rules based on hypothesis testing:

The tracked object is in the coverage area of L_i if

(1) $\max_{all\ k} \{N(T, L_k; W)\} = N(T, L_i; W)$ or

(2) $\max_{all\ k} \{R(T, L_k; W)\} = R(T, L_i; W)$

According to rule (1), the decision is made on the basis of numbers of packets received by individual Lbeacons: The system concludes that the tag is in the coverage area of Lbeacon L_i if L_i received in the current observation window the largest number of packets with RSSI higher than T. According to rule (2), the decision is based on the average RSSI values of all packets with RSSI higher than T received by individual Lbeacons: The system concludes that the tag is the coverage area of Lbeacon L_i if the average RSSI of all packets received by L_i is at least equal to that of all other Lbeacons.

To illustrate, one can see from the summary statistics on Lbeacons around location p11 and p12 in Figure 2, A03, A04 and A09 perform less well compared with other nearby Lbeacons. Sample values from them are ignored. Based on sample values of N(-70, 120*,* L_i), the system may decide that the tag at p12 is located in the coverage area of A10 since N(-70, 120*, A10*) is the largest among all samples. In this case, the system would make the same decision if it uses rule (2) since R(-70, 120*, A10*) = -62 is the highest average RSSI among that of all Lbeacons. Similarly, the system decides that the tag at p11 is under the coverage of A02 and A10 according to rules (1) and (2), respectively, leading to location errors of the same size.

In the example illustrated by Figure 3 and the right half of Table 1, the system would decide that the tag at location p27 is in the coverage area of B13 according to both rules (1) and (2). It would decide that the tag at location p25 is in the coverage area of B12 (or B09) according to rule (1) and in coverage area of B09 according to rule (2).

These location-ignorant decisions are illustrated by the small maps in the left half of Figure 4.As it turns out, the decisions are excellent ones: As one can see from the small map at the upper left corner, A10 is indeed the Lbeacon closest to p12. The choices of A02 and A10 for tag at p11 are also the best for the given locations of Lbeacons relative to the location of p11. Similarly, rules (1) and (2) work as well as they can for tags at p27 and p25. The relatively large location errors are results of sparse development of Lbeacons around these locations.

Location-aware estimation rules A parameterized family of location aware rules compute an estimate (X_t, Y_t) of the tag location based on the locations of Lbeacons that contributed the best K samples values of the chosen test statistics among samples from all Lbeacons. Let $\boldsymbol{S}(N, K)$ and $\boldsymbol{S}(R, K)$ denote the sets of Lbeacons that have the K best sample values in the sets $\{n(T, L_i; W)\}$ and $\{r(T, L_i; W)\}$ of sample values of test statistics $\{N(T, L_i; W)\}$ and $\{R(T, L_i; W)\}$, respectively, in the

current observation window W. Let x_i and y_i denote the horizontal coordinates of lbeacon L_i. Below are examples of location-aware rules:

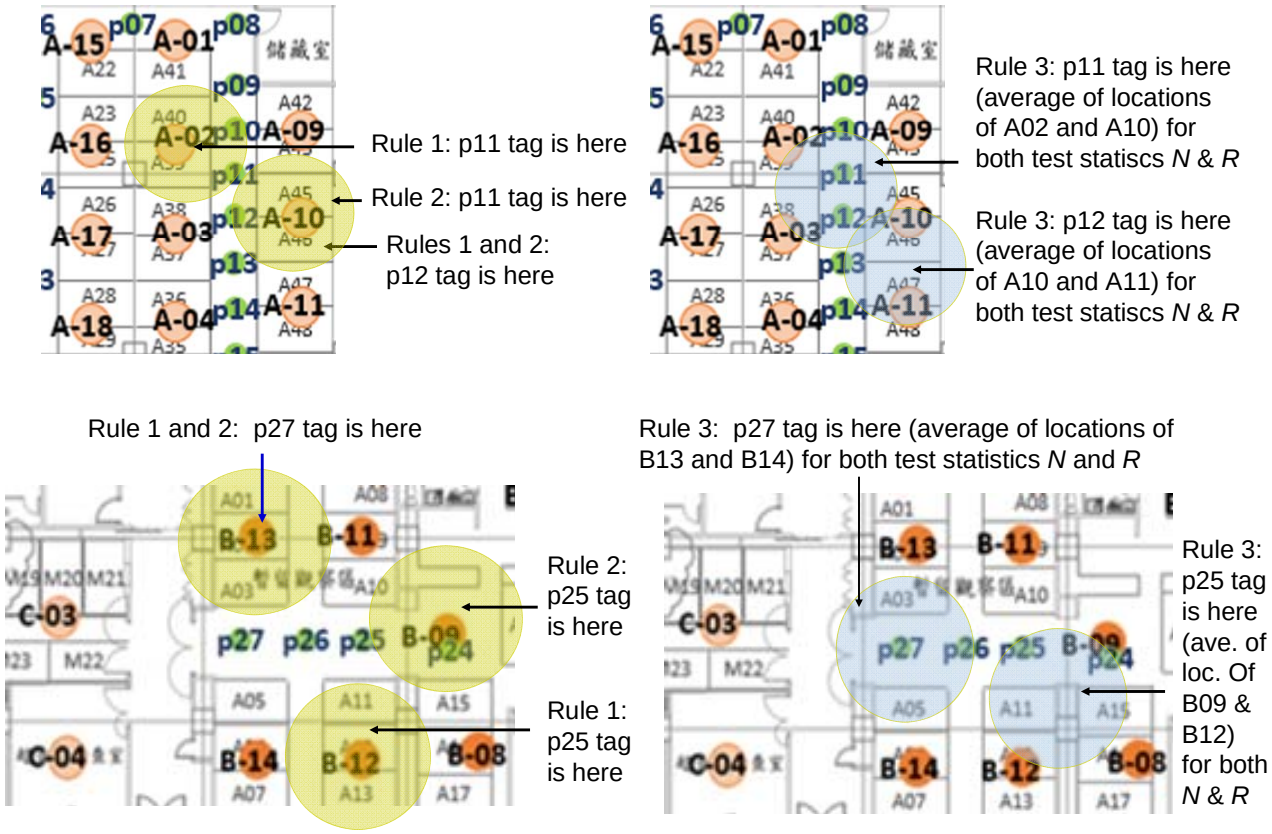

Figure 4 Example illustrating the results of decision/estimation rules

(3) *Geographical centriod rule*: The estimated coordinates (X, Y) of the tag is the average of the coordinates of the K points in $\{(x_i, y_i) \mid L_i \subseteq \boldsymbol{S}(N, K)\}$ when the test statistics is $\{N(T, L_i; W)\}$, or the K points in $\{(x_i, y_i) \mid L_i \subseteq \boldsymbol{S}(R, K)\}$ when $\{R(T, L_i; W)\}$ is the test statistics.

(4) *Weighted centriod rule*: The estimated coordinates (X, Y) of the tag is the weighted average of coordinates of the K points in $\{(x_i, y_i) \mid L_i \subseteq \boldsymbol{S}(N, K)\}$ (or $\{(x_i, y_i) \mid L_i \subseteq \boldsymbol{S}(R, K)\}$) when the test statistics is $\{N(T, L_i; W)\}$ (or $\{R(T, L_i; W)\}$) where the weight w_i of (x_i, y_i) is given by

$w_i = n(T, L_i; W) / \sum_{L_j \subseteq \boldsymbol{S}(N,K)} n(T, L_j; W))$ or

$w_i = r(T, L_i; W) / (\sum_{L_j \subseteq \boldsymbol{S}(R,K)} r(T, L_j; W))$

for test statistics $\{N(T, L_i; W)\}$ or $\{R(T, L_i; W)\}$, respectively.

The right half of Figure 4 illustrates rule (3): One can see that rule (3) provides the best location estimate of the tag at p11 with near zero error, but a location estimate of the tag at p12 has a slightly larger error than the choices of rules (1) and (2).

Both examples support the conjecture stated below:

- *Simple rules (1) and (2) works well enough in areas where Lbeacons are densely deployed, especially when the distances between each Lbeacon and its nearest neighbors are within the required location error for the area.*

- *Rules such as (3) and (4) work better than rules (1) and (2) where Lbeacons are relative sparsely deployed (e.g., when the coverage areas of individual Lbeacons no longer overlap and the required location error is equal to or smaller than the size of their coverage areas.*

4. False Movement Elimination Methods

Often, the best samples for a tag from nearby Lbeacons are close in values. Table 1 illustrates this behavior: Some of the best sample values are either equal or differ by a small percentage. One can easily see that in the presence of natural fluctuations in RSSI's of data packets and hence variations in sample values, the locations of a tag computed by the system according to the rules presented above and displayed by the GUI may exhibit false movements. For example, the locations of the tag displayed by GUI at consecutive update instants may differ, showing the tag moving even when the tag is stationary. A *false movement elimination* (*FaME*) method aims to minimize or eliminate false movements when it is applied after a decision/estimation rule or in combination with the rule to determine the displayed location of each tag at each GUI update instant.

There are two types of FaME methods: *Quantized sample value* method and *lowpass filtering* method. They all consider data and decisions in multiple observation windows and at multiple update instants. To present their common approach, let t_0 denotes the current update time instant, and… t_{-1}, t_0, t_1, ... denote the past and future instants. According to the rules described above, at t_0, the current location of each tag is computed based on the RSSI values of data packets collected from the tag by all nearby Lbeacons in the current window ($t_0 - W, t_0$]. The GUI might exhibit false movements if it would display the location thus computed directly, replacing the location displayed starting at t_{-1}. Rather than doing so, FaME extensions take into account also sample data collected and or decisions made during one (or more) previous observation window(s) ($t_{-1} - W, t_{-1}$], ($t_{-2} - W, t_{-2}$] and so on. (In practice, consecutive update instants are one or two seconds apart, while observation windows are 5, 10 or 20 seconds in size.)

Quantized Sample Approach: Simply put, the quantized sample method considers two sample values different only when their differences exceed the quantum *Q*. *Quantum size Q* is specified in terms of either number of data packets or number of decibels (such as 2 packets or 2 decibels) for test statistics *N* and *R*, respectively, or in terms of a percentage of a given sample value. Rules (1) and (2) can be extended as follows: The extended rules are stated in terms of the notations below in addition to quantum size Q:

- L_{-1} *and* L_{-0} denote the Lbeacon selected at the previous update instant t_{-1} and current update instant t_0 to be the location of the tracked tag whose current location is to be computed.
- W_0, W_{-1} and W_1 denote the current observation window ($t_0 - W, t_0$], the previous observation window ($t_{-1} - W, t_{-1}$], and the next observation window ($t_1 - W, t_1$], respectively.
- N_{-1} and N_0 (or R_{-1} *and* R_0) denotes the sample values based on RSSI's of data packets from the tag collected by L_{-1} in W_{-1} and L_0 in W_0, respectively

In essence, at the current update time, the FaME extended rules (1) and (2) continue to use the

decision L_{-1} made at the previous update time t_{-1} if the best sample value N_0 (or R_0) at current time collected by L_0 differ from sample value from L_{-1} by no more than the quantization size Q. If the difference exceeds Q, L_0 is the decision. Below are formal definitions of the extended rules

(1E) *FaME extended rule* (1): *The tracked tag is in the coverage area of* L_i *if*
$\max_{all\ k} \{N(T, L_k; W_0)\} = N(T, L_i; W_0) >= N(T, L_{-1}; W_0) + Q$
Else the track object is in the coverage area of L_{-1}

(2E) *FaME extended rule* (2): *The tracked tag is in the coverage area of* L_i *if*
$\max_{all\ k} \{R(T, L_k; W_0)\} = R(T, L_i; W_0) >= R(T, L_{-1}; W_0) + Q$
Else the track object is in the coverage area of L_{-1}

To illustrate using the data from Table 1, note that both rules (1) and (2) decide that the tag at p12 is in the coverage area of A10 at t_0 because the sample values of N = 48 (and R = -62) from the Lbeacon are the best at the time. Suppose that Q is 2 packets (or 2 db). Moreover, suppose that at the next update instant t_1, A11 has the best sample values and they are N = 44 (and R = -63 db). According to the FaME extended rules, the decision is that the tag at p12 is under the coverage of A11 at t_1, if at t_1, the sample value N (or R) from A10 is less than 44-2 = 42 (or -63-2 = - 65db); otherwise, the tag at p12 remains to be under the coverage of A10.

Lowpass Filter FaME Method As its name implies, the *lowpass filter FaME method* aims to filter out fast changes in location estimates that violate physical laws governing the movements of tracked objects. It is applied after a location decision/estimation has been computed. To illustrate, we note from the right half of Table 1 that rule (1) may decide that the tag at p27 is covered by B13 at the previous update instant and by B14 at the current update instant because these Lbeacons have the best sample values for both test statistics. However, the separation between Lbeacons is about 10 meters, update instants are one second apart and people and equipment/devices pushed by people move at 1 to1.5 meters per second. It is impossible for the tag to leap to B14 at current update instant if it was at B13 at the previous update instant. If the speed limit were set at 1.5 m/s, then the position of the tag at current update instant is at most 1.5 meter from B13 on the line connecting locations of B13 and B14. This is the underlying approach of the lowpass filter method.

In case of this example and in general, the lowpass filter method is not needed when the tags are stationary. The leap mentioned above can be easily eliminated by using FaME extended rules (1) and (2) with Q set larger than zero (e.g., 2). Similarly, centriod rules (3) and (4) should conclude the tag at p27 is in between B13 and B14.

The tag may be moving, however. A reasonable approach works as follows;

1. Compute coordinate estimate of the tag at each update instant using a centriod rule (3) or (4).
2. Plot the trajectory of the tag at the current update instant t_0 by extending the trajectory generated from past M (> 1) coordinate estimates (X_{-M}, Y_{-M}), ... (X_{-2}, Y_{-2}), (X_{-1}, Y_{-1}) of the tag to include the current coordinate estimate (X_0, Y_0). Any of the well known curve-fitting methods with a sufficiently large M can be used to produce a smooth trajectory.

3. At each instant, the maximum speed of the tag is taking into account by constraining the distance traveled between consecutive update instants. If necessary, re-plot the trajectory generated at one or more previous update instants to satisfy the constraint.

5. Application to Fingerprint-based Systems

Despite the vast differences between the systems, the indoor location decision/estimation and false movement elimination techniques presented in previous sections for Lbeacon-based systems can be easily modified and applied to fingerprint-based systems. As stated earlier, a fingerprint, also frequently called a *signature,* is a set of location-specific observed sensor values. A fingerprint based indoor positioning system typically partitions the indoor space into cells of a size roughly equal to its required location accuracy. It maintains a database of signatures/fingerprints captured for each and every cell during setup and maintenance times and uses a location/fingerprint server to respond to each query for the cell with signature matching the signature provided by the user.

This is where the similarity among the fingerprint-based systems ends. As example, the systems described in papers cited above differ significantly in the types and characteristics of the signatures used by them. A signature at a location and stored for the cell containing the location may be

- A vector of RSSI values of signals received at the location from σ WiFi sources [2, 3], or
- A vector of vectors, each of which contains RSSI value, SNR (signal-to-noise ratio), and multipath indicator of the signal from each of the σ FM stations [3, 4], or
- A geomagnetic signal or σ lines in the FFT (Fast Fourier Transform) of the signal [1, 5], etc.

In the terms introduced in the previous sections, they are summary statistics.

To determine its own location, a mobile device sends the signature S_u captured by it at its location to the server and relies on the server to find the cell with a matching fingerprint. The decision on which cell k has the stored signature S_k that best matches the input signature S_u from the user device is often made on the basis of Manhattan distance or Euclidean distance between the stored vector S_k and input vector S_u. The chosen distance is the test statistics. For an input signature S_u, and the values of the distances D_M (S_k, S_u) or D_E(S_k, S_u) for all cells k are the sample values. A decision/estimation of the location with the input signature S_u can be made using any of the rules (1) – (4) based on the sample values.

6. Summary

This report describes several methods that can be used by a indoor object tracking system (IOTS) to compute the locations of tracked objects at each update instant based on values of sensor data available at that instant and previous update instants. The methods aim to minimize location errors on the one hand, and to reduce or eliminate false movement on the other hand. The term noisy/false movement refers to sporadic changes in object locations displayed by the GUI caused by unpredictable fluctuations in sensor data. Users are likely to find such movements annoying and location information confusing.

The methods are defined in terms of statistical decision and estimation rules. Once the test statistics are chosen, it is straightforward to apply them. Specific details about the IOTS are not important. For this reason, the methods can be applied by IOTS based on proximity detection as well as systems based on fingerprints.

References

[1] "Indoor Atlas uses magnetic sensors to find places in the great indoors," https://venturebeat.com/2013/11/15/indoor-atlas-uses-magnetic-sensors-to-the-great-indoors/, (Last retrieved: May 10, 2020) and "Magnetic Positioning, The Arrival of 'Indoor GPS'," https://www.indooratlas.com/2014/06/01/magnetic-positioning-the-arrival-of-indoor-gps/, (Last retrieved: May 2020), Opus Research, June 2014,

[2] A. Farshad, J. Li, M. K. Marina and F. J. Garcia "A microscopic look at WiFi fingerprinting for indoor mobile phone localization in diverse environments," *Proceedings of International Conference on Indoor Positioning and Indoor Navigation,* October 2013.

[3] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha, "FM-based indoor localization," *Proceedings of ACM MobiSys*, 2012, pp. 169-182.

[4] A. Popleteev, "Indoor Positioning Using FM Radio Signals," PhD Dissertation. University of Trento, Italy, 2011.

[5] Jaewoo Chung, Matt Donahoe, Chris Schmandt, Ig-Jae Kim, Pedram Razavai, and Micaela Wiseman, "Indoor location sensing using geo-magnetism," *Proceedings of ACM MobiSys'11*, 2011.

[6] S.-H. Fang and T.-N. Lin., "Cooperative multi-radio localization in heterogeneous wireless networks," *IEEE Transactions on Wireless Communications*, 9(5), 2010, pp. 1536-1276.

[7] C. C. Li, J. Su, E. T.-H. Chu, and J. W. S. Liu, "Building/environment Data/information Enabled Location Specificity and Indoor Positioning," *IEEE Internet of Things Journal,* Volume: 4, Issue: 6, pp. 2116 – 2128, Dec. 2017

[8] J. W. S. Liu, L. J. Chen, J. Su, C. C. Li and E. T.H. Chu, "A Building/environment Data Based Indoor Positioning Service," 2015 *IEEE International Conference on Data Science and Data Intensive System*s, December 2015.

[9] Summary Statistics, https://en.wikipedia.org/wiki/Summary_statistics, (Last retrieved: May 2020).

[10] Test Statistics, https://en.wikipedia.org/wiki/Test_statistic, (Last retrieved: May 2020).