CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE
Concurrency Computat.: Pract. Exper. 2015; **27**:2839–2850
Published online 5 June 2015 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/cpe.3443

SPECIAL ISSUE PAPER

Robust wireless signal indoor localization

Liang Kong, Gavin Bauer and John Hale*,†

Institute for Information Security, The University of Tulsa, 800 Tucker Driver, Tulsa,OK 74104, USA

SUMMARY

Localization is a key enabler of context awareness in computing environments. This paper presents a technique for indoor localization using wireless signal strength from mobile devices. The method described treats locations as fuzzy sets and fuzzifies signal strength-related features to define membership. Membership values are then fused from multiple sources using a rule engine to deduce objective location values. The principal benefits of this technique are that it requires little or no calibration and that it can be used with widely available commercial devices. Simulation shows that this technique is robust to errors and provides reasonable accuracy. Applications to collaborative workflow and human computer interaction are discussed. Copyright © 2015 John Wiley & Sons, Ltd.

Received 20 February 2014; Revised 13 October 2014; Accepted 5 November 2014

KEY WORDS: indoor localization; fuzzy logic; wireless signal; context awareness

1. INTRODUCTION

Tracking user locations greatly benefits applications in security, context awareness, and social networks. Localization is a key enabler of context awareness in computing environments. Outdoor localization has been widely used based on satellite localization systems, for example, the Global Positioning System [1]. On the other hand, indoor localization has not yet been studied extensively.

This paper introduces a technique to localize widely available commercial devices via wireless signals. In this paper, traditional coordination-based location is replaced with fuzzy location which is a formalized point of interest. This approach localizes devices to fuzzy locations which is accurate enough for most applications while requiring less calibration and can be implemented with consumer grade device.

Bluetooth technology is used where received signal strength indication (RSSI), link quality, and transmission power level are sampled by stationary scanners with pre-configured locations. They are then fuzzified to membership values with respect to the scanners. Each scanner represents a fuzzy set and provides its own subjective location observations to a host. The host collects observations and resolves conflicts to form an objective location result.

This approach only requires a simple configuration of each scanner for its room location. Once launched, each scanner automatically joins devices and localizes incoming devices. Device location can be used in policies applying function restriction and other security measures.

In Section 2, background is presented including indoor localization and fuzzy logic. We present a way to calculate three different signal and link quality measures into a membership value in Section 3. Section 4 formalizes location inference, defines subjective location and objective

*Correspondence to: John Hale, Institute for Information Security, The University of Tulsa, 800 Tucker Driver, Tulsa, OK 74104, USA.
†E-mail: hale-john@utulsa.edu

location, and presents an inference approach. In Section 5, we present a reference implementation along with validation and comparison. Conslusions and future work are presented in Section 6.

2. BACKGROUND

This section presents background information on indoor localization techniques and introduces fuzzy logic and domain specific languages (DSLs) as supporting elements of our approach and implementation.

2.1. Indoor localization

Localization of an indoor mobile device is an active field of study. Research has explored various technologies including Active Badge System [2], wide area cellular systems [3], dead reckoning [4], Wifi [5], and Bluetooth [6–8].

An indoor localization system is typically composed of static scanners/beacons with known location features. It estimates device locations by measuring communications. RSSI, time-of-arrival (TOA), and angle-of-arrival can be used in estimation. In this study, we target smartphones on the consumer market, which have limited access to accurate metadata due to system/hardware constraints.

For Bluetooth indoor positioning, there are two types of approaches: connection-based and inquiry-base [9]. In Figure 1, the Bluetooth devices' (dot) location can only be determined under a coordination system setup based on scanner (square) locations. Most localization solutions require pre-calibrated sensors with known coordinates. Some may record signal distribution in certain area as a 'fingerprint' to match sensors' recording [10]. But in real world environment, obstructions including walls and furnitures bring in a great amount of distortion into the triangulation process. All

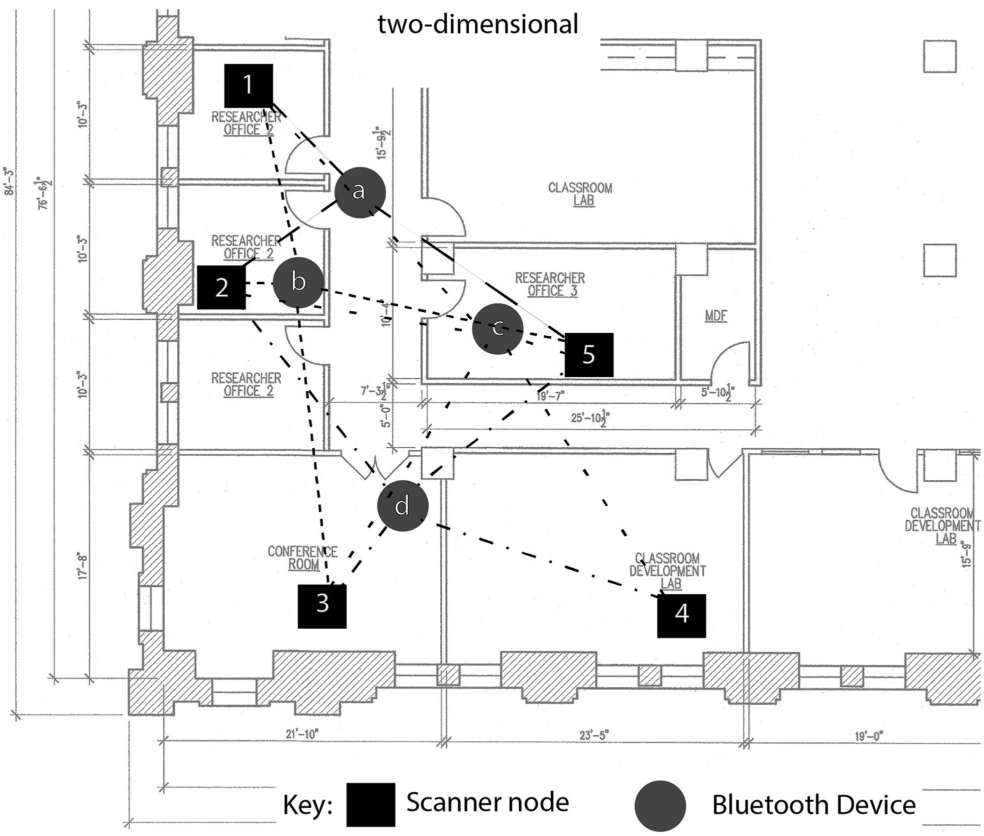


Figure 1. Conventional localization approach.

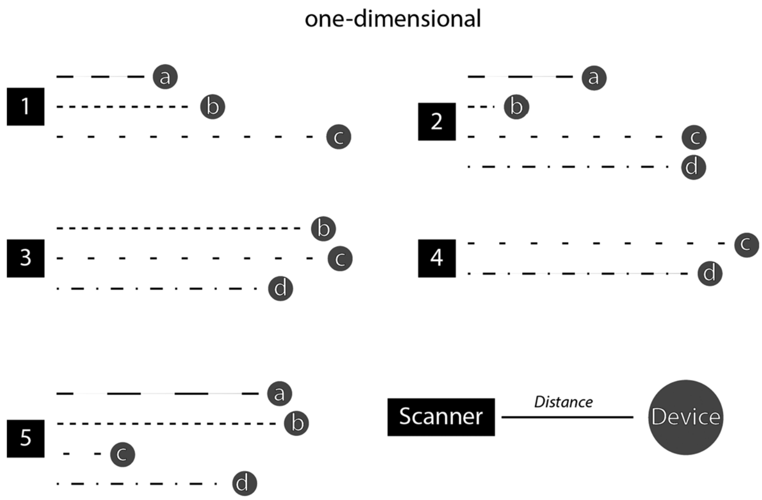


Figure 2. Scanners' reception perspective.

of these require deploying and calibrating equipment, some even during construction. More importantly, common deployment strategies treat an entire floor of a building uniformly, whereas in most instances, individual rooms are a more appropriate level of granularity. Other solutions may use difference time of arrival, which relies on nanosecond-scale precision in specialized hardware [11].

In this study, fine-grained device locations (x, y, and z-coordinates) are not required. Instead, information that indicates which room (usually at least a 3×3-m area) is good enough to provide contextual support for collaboration. Our system ingests the information scanners receive, without knowing their coordinates, as illustrated in Figure 2. These are distance values without known directions. For simple deployment and configuration, the presented system is constituted with one Bluetooth scanner in each room and Bluetooth-enabled smartphones. Each scanner continually interrogates Bluetooth-enabled devices in range with RSSI.

2.2. *Fuzzy inference system*

Fuzzy inference systems [12] are computing frameworks based on fuzzy logic, if-then rules, and fuzzy reasoning. Normally, they are comprised of three components:

1. A rule base, which contains a set of rules.
2. A database, which contains membership functions for rules.
3. A reasoning mechanism, which acts as a rule engine [13].

In contrast with finite-valued logic (e.g., 'true', 'false', and 'unknown'), fuzzy logic is an infinite-valued logic. A classic (crisp) set is normally defined as a collection of elements or objects x $\in$ X that can be finite, countable, or over countable. Each single element can either belong to or not belong to a set A (finite valued logic), A $\subseteq$ X [14].

For a fuzzy set, the characteristic function allows various degrees of membership for the elements of a given set. A fuzzy set A in X can be represented as $A = \{(x, \mu_A(x)) \| x \in X\}$, $\mu_A(x)$ is called the membership function or grade of membership (also degree of truth). For example, a fuzzy set representing a user's preference on laptop brands is

$A = \{(Lenovo, 0.6), (Dell, 0.3), (HP, 0.3), (Apple, 0)\}$ (note the sum of grade of membership is not necessarily less than or equal to 1).

A fuzzy rule system [15] usually takes crisp inputs and fuzzifies them into terms with truth values by membership functions. For example, A temperature of 99 is 0.5 of high, which is expressed as $\mu_{temperature} = high(99) = 0.5$. It could also be 0.1 of low, which is expressed as $\mu_{temperature} = low(99) = 0.1$. Then the system fires rules accordingly to defuzzify for crisp output. Continuing the last example, the fuzzy rule system was designed to turn the power to medium when the temperature is high and the humidity is high. Then it is expressed as follows: $\mu_{power=medium} = min[\mu_{temperature=high}, \mu_{humid=high}]$. So the truth value of the rule is decided by the minimum of truth values of temperature and humidity. In a defuzzification process, fuzzy sets and corresponding membership are used to produce a quantifiable result, in this case, a distance value from 0 to 100, for example, power level medium is 90. There are several defuzzification methods, for example, adaptive integration, basic defuzzification distribution, constraint decision defuzzification, and center of gravity (COG).

In this work, the COG is chosen for speed and accuracy. This method calculates the COG for the area under the curve: $COG = \frac{\Sigma_{x=a}^{b}\mu_A(\chi)x}{\Sigma_{x=a}^{b}\mu_A(\chi)}$.

A DSL [16] is a programming language or specification language dedicated to a particular problem domain, a particular problem representation technique, and/or a particular solution technique [17]. Hypertext Markup Language [18] is a well known and widely used DSL.

Our system uses a python-based DSL [16, 17] and fuzzy rule engine, pyfuzzy, to define and execute localization rules. A fuzzified rule set is used to process numerical values into fuzzy assertions. These assertions are then used to generate fuzzified results. Finally, fuzzified results are defuzzified into numerical values.

3. BLUETOOTH APPROXIMATION WITH FUZZY CLASSIFIER

In this section, we present an approximation method using Bluetooth signals in conjunction with a fuzzy classifier. This method considers multiple aspects of Bluetooth transmission and generates more accurate membership values than using any single feature. In the presented system, multiple scanners are used to sense the locations of devices. Each scanner is running the same software and reports to a host for coordination.

3.1. Data collection and pre-process

As a testing platform, we use pyBlue and hcitool as querying tools. pyBlue is a python module used for scanning active Bluetooth devices and inquiry for RSSI. hcitool is a Linux command used to connect to Bluetooth devices and get transmission power level and link quality.

In the first stage of scanning, each scanner uses pyBlue to obtain the MAC addresses of surrounding devices as well as their RSSI. RSSI is reported as decibel-milliwatts. Although decibel-milliwatts is not a linear measurement of signal strength, it is treated as linear here. In this case, Bluetooth RSSI is usually in the range of -99 to -35 dBm. During the scanning stage, one device often gets scanned multiple times as pyBlue is trying to discover every device in range. At the end of the first stage, a map of MAC addresses and average RSSI is reported.

In the second stage of scanning, hcitool is used to connect to every device in the map. Once a connection is established, transmission power level and link quality are acquired and attached to devices according to MAC addresses.

After the second stage of scanning, we have a map of device MAC addresses, RSSI, transmission power level, and link quality. Each scanner uses a fuzzified function to calculate a membership value for devices in its returned map.

3.2. Fuzzy classifier and membership approximation

In this section, we present a fuzzy system which converts a set of numerical signal strengths to a membership value of a fuzzy set. A fuzzy classifier works in tandem with a membership approximation scheme to enable our indoor localization technique.

pyfuzzy is a framework to work with fuzzy sets and processes them with operations in fuzzy logic. In this system, it is used as a fuzzifier and language base. pyfuzzy is controlled by a Fuzzy Control Language file, wherein input values are fuzzified and then checked against rules. Rules are used to generate results, for example, IF (Temperature IS Cold) THEN (Output IS High).

DOI: 10.1002/cpe

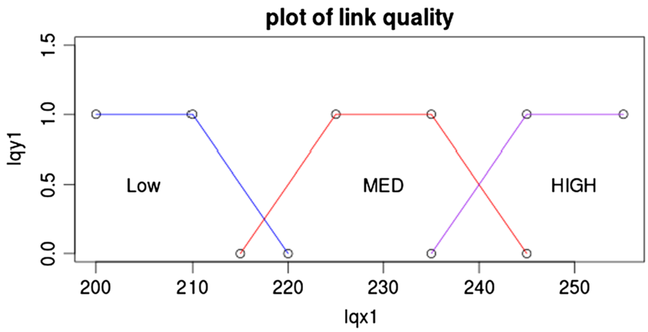


Figure 3. Fuzzifier for link quality.

In the fuzzy control language, the basic concept is the 'control block'. Each control block has a set of input variables and one output variable. First, we define the input and output of control block as follows:

```
VAR_INPUT
    rssi: REAL;
    lq: REAL;
    tpl: REAL;
END_VAR
VAR_OUTPUT
    membership: REAL;
END_VAR
```

We define a list of linguistic terms for each range of values (e.g., link quality)

```
FUZZIFY lq (*Link quality*)
    TERM LOW := (200,1)(210,1)(220,0) ;
    TERM MED := (215,0)(225,1)(225,1)(245,0) ;
    TERM HIGH := (235,0)(245,1)(255,1) ;
END_FUZZIFY
```

Each given input is fuzzified into a term or terms with membership value from 0 to 1 (Figure 3). In this particular control block, there are three input variables with fuzzifiers for each of them. Similarly, a defuzzifier is defined for the output

```
DEFUZZIFY membership
    TERM VERY_CLOSE :=100;
    TERM CLOSE :=80;
    TERM HiMEDIUM :=60;
    TERM LoMEDIUM :=40;
    TERM FAR :=20;
    ACCU:MAX;
    METHOD: COGS;
    DEFAULT :=30;
END_DEFUZZIFY
```

In a defuzzification process, multiple rules are allowed and expected to fire at the same time. An RSSI value of -55 is both HIGH and HiMED in after fuzzification. More specifically, it has truth values of 0.3 of HIGH and 1 of HiMED. Similarly assuming, link quality is 1 of HIGH, and transmission power level is 0.5 of LOW and 0.5 of MED. As a result, rules 1 and 4 are triggered. Then truth values are calculated for each rule that is triggered. With a rule specifying an AND relationship among its inputs, the minimum of them is used, for example, $truth_{rule1} = min(truth_{rssi}, truth_{tpl}, truth_{lq})$. In this particular case, rule 1 has a truth value of 0.3 and rule 4 has

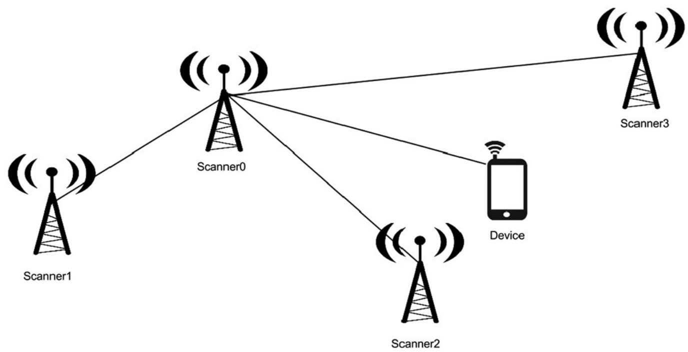


Figure 4. Scanner and device relationships.

a truth value of 0.5. As a result, the output should be $\frac{0.3\times CLOSE+0.5\times HiMEDIUM}{0.3+0.5}$ which is 67.5. This crisp output value is called membership value denoted as M.

4. LOCATION ASSUMPTION AND INFERENCE

This section presents a device localization framework based on the approximation method using membership values and a scanner network.

4.1. Scanner network

This presented system is constituted of multiple physical scanners with ethernet connections. Each scanner receives an updated scanner list right after submitting its own MAC address. Thus, every scanner has a list of other scanners' MAC addresses and does not scan them repeatedly because scanners are relatively static.

A scanner scans other scanners less often, keeping a map of their MAC address and membership with respect to the scanner itself. Once a new device enters the range of the scanner, it is reported to the host by the scanner. The host then utilizes the information to localize the newly entered device.

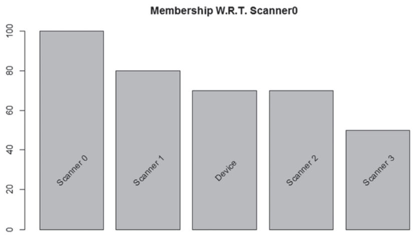
4.2. Location assumption

The host generates an assumption based on the signal strength of the device and other scanners with respect to one scanner. For example, given a case illustrated as Figure 4, longer lines stand for greater distance between units. With the information sent by scanner 0, we could draw a chart such as Figure 5(a).

To formalize, we denote a subjective location of a device i as L_{ki} with respect to k and the objective location of device i as $\bar{L_i}$. We assign the device an id 'd' and the set of scanners 'S'. The whole localization process is deducing $\bar{L_d}$ from $L_{kd}\ :\ k\ \in\ S$. Here. we define $\approx$ as the 'closest to'. A device can be closest to one or multiple scanners. The result of the process should be $\bar{L_d} \approx \{\bar{L_{k1}}, \bar{L_{k2}}, \bar{L_{k3}}, \ldots, \}, k_i \in S$.

First, we need to derive L_{ki} from membership values. We denote the membership of device or scanner i with respect to scanner k as $M_{k,i}$. Then we have $D^k_{i,j}$ as differences of locations between device or scanners j and i with respect to scanner k. Let $D^k_{i,j} =\mid M_{kj} - M_{ki} \mid$. Naively, $argmin_i D^k_{i,d}$ should give the location of device 'd': $L_{kd} \approx \bar{L_i}$.

This assumption might not always be correct because membership is a one-dimensional value. Luckily, we have more information coming from other scanners. The assumption made based on data sent from scanner 1 would look like 5(b), similar for scanners 2 and 3.



(a) Membership Chart With Respect To Scanner 0

(b) Membership Chart With Respect To Scanner 1

(c) Membership Chart With Respect To Scanner 2

(d) Membership Chart With Respect To Scanner 3

Figure 5. Membership chart.

Table I. Example LogicGrid.

	Scanner 0	Scanner 1	Scanner 2	Scanner 3
Scanner 0	30	30	40	60
Scanner 1	10	50	60	40
Scanner 2	0	0	10	20
Scanner 3	20	10	20	10

4.3. Localization inference

Because assumptions generated from scanners can be self-contradictory, a mechanism to solve conflicting assumptions and yield the result closest to reality is needed. The problem here is much like a logic puzzle. It is a logic problem constituted by multiple assertions asking what can be deduced from them. An extended form of logic puzzle is Sudoku. There are three rules used to solve these kind of problems: contrapositive rule, implication rule, and contradiction rule [19].

One approach is the use of a logic grid, which uses assertions as rows and the origin of assertions as columns. Then deduction is made row by row to fill unknown blocks in the grid. If there were conflicts in assertions, these would be reflected as one column that does not fit in the grid. Removing this column eliminates conflicts in the grid. Similarly, the logic grid could be used in the localization problem with small tweaks. A logic grid for device 'd' can be generated as $G_{i,j} = D^{i}_{jd}$. For the situation shown in Figure 4, there is at least one conflict as shown in Table I.

With the entire picture presented, it is easy to choose n rows with the lowest sum as the location of device 'd'. In the given example $\bar{L_d} = \bar{L_2}$. In the case that it is preferable to know the closest two locations of device 'd', two rows are chosen: $\bar{L_d} \in \{\bar{L_2}, \bar{L_3}\}$

4.4. The whole simulation section

A simulation is done to verify the design. About 3 to 10 points are randomly generated in a 100 $\times$ 100 space as stationary receivers. Each receiver defines a fuzzy location. Then a new point is inserted into the space to be localized. The coordinates of this point are known and this is called the true location.

Fuzzy location membership values are generated as distance with errors up to 50%. Errors are fashioned into account for signal attenuation and other system imperfections. For a given error range

DOI: 10.1002/cpe

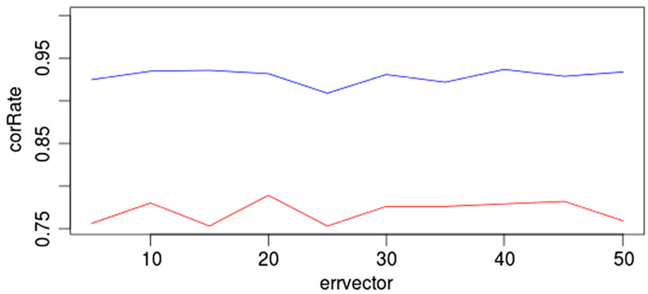


Figure 6. Simulation location accuracy.

Table II. Device subjective locations.

Reference scanner	Device	Membership value
C4:46:19:E4:XX:XX	22:22:54:C7:XX:XX	37.41935483870968
58:91:CF:3B:XX:XX	22:22:54:C7:XX:XX	49.99999999999999

X, we have *Membership* $=$ *Distance* $*(1+x)$, where x is randomly generated and $-X \leqslant x \leqslant X$. Those values are then fed into a logic grid to calculate the closest location. Figure 6 is the plot of the simulation where y-axis is the rate of successful localization and x-axis is the error range. The lower line around 75% is the correct rate when the best singleton match was chosen. This is because the target device sits in the overlapping zone between two or more scanners. When we extend the result to the best two matches, the error rate is shown as the higher line around 94%.

5. IMPLEMENTATION AND VALIDATION

In this section, we present a reference implementation of our localization approach. Our implementation is then compared with other existing approaches.

5.1. Implementation

This subsection presents our reference implementation of the localization technique with minor adjustments. The reference implementation engages two software systems—one for scanners and one for hosts. The host is a program running as a network server to which all scanners report. The host also generates logic grids and reports the localization results. Scanners and hosts communicate via the network following the SOAP standard. When a scanner starts, it reports to the host to check in with its MAC address and pre-configured room ID. This scanner then scans for surrounding scanners communicating that information to the host. The scanner then begins to scan for devices that are not known scanners. Every time a scanner discovers a device, it reports this information to the host, wherein a logic grid is computed in order to localize this device.

The host uses two tables for storing data, one for devices, for example, Table II and one for scanners, for example, Table III. The logic grid is generated from those two tables. Table IV is created from Tables II and III. In this example, two scanners are placed to localize one device. 58:91:CF:3B:XX:XX is with room id: U315 and C4:46:19:E4:XX:XX with room id: U314. The location is given by the column with minimum sum value, which in this example indicates that the device labeled as 22:22:54:C7:XX:XX is in U315.

In a real-world scenario, wireless scanning is unstable which makes implementation challenging. Three principal challenges confront our approach: (i) acquisition of link quality and power transmission level data; (ii) the range limitations of scanners; and (iii) the latency of scan rates, which may create temporal gaps in device localization knowledge. The first two challenges are addressed directly in our implementation, whereas the third remains an open area of research.

DOI: 10.1002/cpe

Table III. Scanner subjective locations

Reference scanner	Target scanner	Membership value
C4:46:19:E4:XX:XX	58:91:CF:3B:XX:XX	40.0
C4:46:19:E4:XX:XX	C4:46:19:E4:XX:XX	100.0
58:91:CF:3B:XX:XX	C4:46:19:E4:XX:XX	34.99999999999999
58:91:CF:3B:XX:XX	58:91:CF:3B:XX:XX	100.0

Table IV. Computed logic grid.

Reference scanner	Distance to 58:91:CF:3B:XX:XX	Distance to C4:46:19:E4:XX:XX
C4:46:19:E4:XX:XX	2.5806451612903203	62.58064516129032
58:91:CF:3B:XX:XX	50.00000000000001	15.0

In a real world scenario, wireless scanning is unstable which makes implementation challenging. Three principal challenges confront our approach: (i) acquisition of link quality and power transmission level data; (ii) the range limitations of scanners; and (iii) the latency of scan rates, which may create temporal gaps in device localization knowledge. We address the first two challenges directly in our implementation, while the third remains an open area of research.

Link quality and power transmission data can only be acquired by establishing a connection to the device, which is sometimes difficult to accomplish. Thus, our scanners are configured to inquire multiple times for link quality and transmission power level, whereas a guardian thread is used to prevent scanners from dead looping. This guardian thread keeps a timer to make sure each inquiry does not take more than 15 s. It is also implemented to avoid race conditions between scanners (e.g., two or more scanners trying to connect to the same device at the same time). To achieve this, the guardian thread makes the scanner sleep after every three failed connections.

In the extreme cases, the scanners are able to get RSSI from one device but failed to acquire link quality and power transmission level multiple times. Link quality and power transmission level of such device are then assigned as 0 and 8.

Furthermore, scanners are not always in range of each other. Scanners are allowed to and may report a partial map of surrounding scanners. In generating the logic grid, the membership values of missing scanners are filled with zeroes.

5.2. Validation

Experimental validation of our reference implementation was conducted on a single floor of an office building using three rooms—U315, U314, and U311. Symbols such as star, dot, and dash in Figure 7 represent devices belonging to a particular scanner group. This experiment takes place in a building with normal obstructions including furniture and doors.

Three laptops running debian wheezy are placed, one for each room, as the larger symbols. Each laptop has a boardcom BCM43XX family wireless chip for Bluetooth communication. Each scanner runs under python 2.7 with libraries including pyBluez, pyfuzzy, simplesoappy, and numpy. All scanners are interconnected over ethernet.

A desktop computer runs as the host and starts before all scanners. Scanners are preconfigured with room IDs and launched (in no particular order). The system starts to yield results after at least two scanners begin running and are checked in. A smartphone (HTC One m7) is used as the device to be localized.

Every time an update is received from any scanners, this system calculates locations for all devices in range. On average, three scanner systems update location of devices every 10 s. Responding time of this system can be improved by adding more scanners to the system.

Every location this device has been placed is labeled as one of the three aforementioned symbols. As shown in Figure 7, the system does a good job when a device is clearly in range of a given scanner and gives more ambiguous results when the device enters the overlapping zones.

DOI: 10.1002/cpe

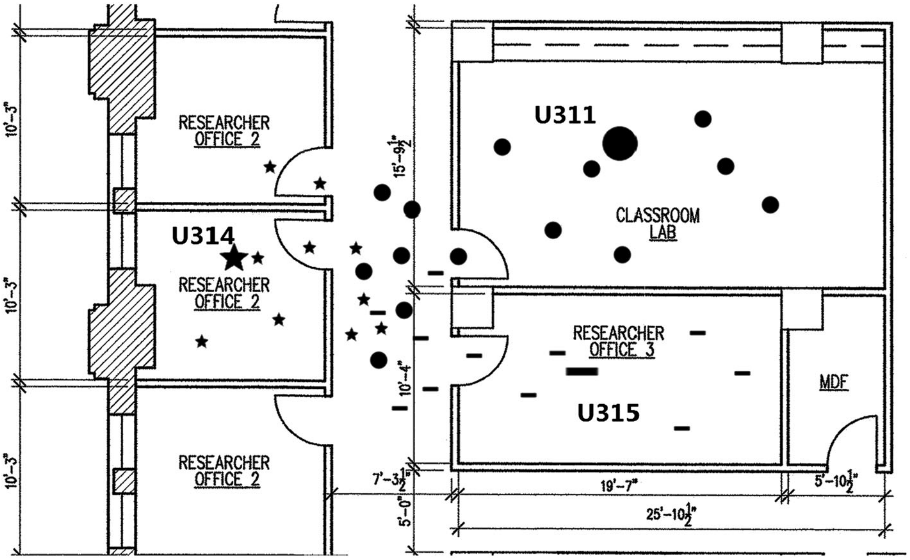


Figure 7. Experimental results.

In the validation, our system achieves 100% accuracy for any devices in one of the labeled rooms. The system fails to localize devices outside preconfigured locations. In this validation, only three scanners are used which cover three rooms of a building floor. The area left undefined should be considered outside of the system. Devices in those areas should not be localized at all.

5.3. Comparison with other approaches

In this section, we compare our approach with existing technologies. Here, we isolate a few approaches mentioned in Section 2. Kothari *et al.* [4], Bargh and de Groote [10], and Patwari *et al.* [20] presented different approaches of indoor localization based on Bluetooth technology.

In [20], a detailed approach is presented using Cramer–Rao bounds to regulate unreliable wireless signal strength. Relative location is calculated based on RSSI and TOA from multiple devices. An experiment in this work is set up with lab grade equipment (DS-SS and Sigtek model ST-515 for RSSI and Datum ExacTime GPS and rubidium time standard for TOA). In this experiment, reference devices are put in corners of two 9 m $\times$ 9 m areas. One is an empty parking lot, another is a residential home with obstructions. In the empty environment, this approach has RMS location error is 1.46 m. In the environment with obstructions, RMS error increases to 2.1 m.

Bargh and de Groote [10] presented an approach utilizing fingerprint-based localization. Every location is fingerprinted by inquiry response rates. The Bluetooth inquiry response rate is defined as 'the percentage of inquiry response to total inquiries'. In the experiment, 6 h of learning time is spent in each room with consumer grade devices. Fourteen rooms are tested in total. This approach claims a 98% accuracy to localize a device to six positions with full Bluetooth coverage. It takes 3 min to localize a device to one location.

Kothari *et al.* [4] combined the multiple complimentary localization systems including dead reckoning, Wifi, and GSM using a particle filter for robust localization. A signal strength map is generated for each floor for Wifi localization. This approach archives a 3 $\pm$ 3 m accuracy.

In Table V, we compare these approaches to our own. Wen a common numerical basis from the literature for comparison is absent , subjective measures are used to characterize each system. In comparison, we focus on calibration time, localization time, expandability, and accuracy of approaches. An approach that takes more than half an hour per location to calibrate is considered high, otherwise, it is medium or low.

For the localization time, more than 1 min is considered high, 30–10 s is considered medium, and lower than 10 s is considered low. Expandability represents the ability of a system to deploy and expand. There is no hard measures on expandability. Generally, if a system needs to be largely

DOI: 10.1002/cpe

Table V. Localization comparison.

	[20]	[10]	[4]	Our approach
Pre-calibration	Low	High	Medium	Low
Localization time	Medium	High	Low	Medium
Expandability	High	Low	Medium	High
Accuracy	High	Medium	Medium	Low

reconfigured to expand, it is considered low in expandability. A system that can expand on the fly or accept new node with little configuration is considered high in expandability. Otherwise, it is considered medium. Accuracy is related to a system's RMS location error. It is considered high when the RMS error is under 2 m, medium when it is 3–5 m, and otherwise low.

6. CONCLUSIONS AND FUTURE WORK

In this paper, a Bluetooth indoor localization technique is presented and tested. Conventional localization techniques require pre-calibrated coordinates of scanners. These approaches rely heavily on the accurate measurement of coordinates and are challenging to apply to consumer mobile device technologies.

This technique takes a new approach where exact coordinates of scanners are not necessary. Instead, devices are grouped under scanners. Every scanner is assigned to a room, region, or zone. The location of devices are determined by the scanner in the same group.

This technique requires little calibration and places more attention on locations that matter (e.g., rooms in a floor). Although not providing coordinates, this approach supplies information needed to control security related features in a unit of a device group. Additional research may improve the performance of the system by considering heuristics to address the temporal limitations of the scheme induced by scan rates and by leveraging previously localized devices to further inform the localization process.

ACKNOWLEDGEMENTS

This material is based on the research sponsored by DARPA under agreement number FA8750-09-1-0208. The US Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, or DARPA or the US Government.

REFERENCES

1. Hofmann-Wellenhof B, Lichtenegger H, Collins J. Global positioning system, 2000.
2. Want R, Falcao V, Gibbons J. The active badge location system. *ACM Transactions on Information Systems* 1992; **10**:91–102.
3. Tekinay S. Wireless geolocation systems and services. *IEEE Communications Magazine* 1998; **36**(4):28.
4. Kothari N, Kannan B, Glasgow ED, Dias MB. Robust indoor localization on a commercial smart phone. In *Procedia computer science*, Vol. 10, Elhadi Shakshukia MY (ed.). Elsevier: Niagara Falls, Ontario, Canada, 2012; 1114–1120.
5. Bahl P, Padmanabhan VN. Radar: an in-building RF-based user location and tracking system, 2000.
6. Bruno R, Delmastro F. Design and analysis of a bluetooth-based ion system. *In Personal Wireless Communications, IFIP-TC6 8th International Conference, PWC 2003*, Venice, Italy, 2003; 711–725.
7. Diaz JJM, Maues RdA, Soares RB, Nakamura EF, Figueiredo CMS. Bluepass: an indoor bluetooth-based localization system for mobile applications. *2010 IEEE Symposium on Computers and Communications (ISCC)*, Riccione, Italy, June 2010; 778–783.
8. Perez Iglesias H, Barral V, Escudero C. Indoor person localization system through RSSI bluetooth fingerprinting. *2012 19th International Conference on Systems, Signals and Image Processing (IWSSIP)*, Vienna, Austria, April 2012; 40–43.

9. Pei L, Chen R, Liu J, Tenhunen T, Kuusniemi H, Chen Y. Inquiry-based bluetooth indoor positioning via RSSI probability distributions. *2010 Second International Conference on Advances in Satellite and Space Communications (SPACOMM)*, Athens/Glyfada, Greece, June 2010; 151–156.
10. Bargh MS, de Groote R. Indoor localization based on response rate of bluetooth inquiries. *Proceedings of the First ACM International Workshop on Mobile Entity Localization and Tracking in GPS-Less Environments*, MELT '08, ACM, New York, NY, USA, 2008; 49–54.
11. IHP innovations for high performance microelectronics, Germany, Humboldt University Berlin, Germany. *Bluetooth indoor localization system*, 2004.
12. Kasabov N, Song Q. Denfis: dynamic evolving neural-fuzzy inference system and its application for time-series prediction. *IEEE Transactions on Fuzzy Systems* 2002; **10**(2):144–154.
13. Jang J. *Fuzzy inference systems*. Prentice-Hall: Upper Saddle River, NJ, 1997.
14. Zimmermann H. *Fuzzy Set Theory–and Its Applications*. Kluwer Academic Publishers: Dordrecht, the Netherlands, 2001.
15. Mendel JM. Fuzzy logic systems for engineering: a tutorial. *Proceedings of the IEEE,* IEEE 1995; **83**(3):345–377.
16. van Deursen A, Klint P, Visser J. Domain-specific languages: an annotated bibliography. *SIGPLAN Not.* 2000; **35**(6):26–36.
17. Spinellis D. Notable design patterns for domain specific languages. *Journal of Systems and Software* February 2001; **56**(1):91–99.
18. Berners-Lee T, Connolly D. Hypertext markup language - 2.0, 1995.
19. Harvey B. Reasoning with computers: inference vs. backtracking. *International Journal of Computers for Mathematical Learning* 1998; **3**(1):81–92.
20. Patwari N, Hero AO III, Perkins M, Correal NS, O'Dea RJ. Relative location estimation in wireless sensor networks. *IEEE Transactions on Signal Processing* 2003; **51**(8):2137–2148.

