2013 IEEE 14th International Conference on Mobile Data Management

Hybrid Indoor Positioning With Wi-Fi and Bluetooth: Architecture and Performance

Artur Baniukevic[†] Christian S. Jensen[‡] Hua Lu[†]

[†]*Department of Computer Science, Aalborg University, Denmark*
[‡]*Department of Computer Science, Aarhus University, Denmark*
artur.baniukevic@gmail.com csj@cs.au.dk luhua@cs.aau.dk

***Abstract*—Reliable indoor positioning is an important foundation for emerging indoor location based services. Most existing indoor positioning proposals rely on a single wireless technology, e.g., Wi-Fi, Bluetooth, or RFID. A hybrid positioning system combines such technologies and achieves better positioning accuracy by exploiting the different capabilities of the different technologies. In a hybrid system based on Wi-Fi and Bluetooth, the former works as the main infrastructure to enable fingerprint based positioning, while the latter (via hotspot devices) partitions the indoor space as well as a large Wi-Fi radio map. As a result, the Wi-Fi based online position estimation is improved in a divide-and-conquer manner.**

We study three aspects of such a hybrid indoor positioning system. First, to avoid large positioning errors caused by similar reference positions that are hard to distinguish, we design a deployment algorithm that identifies and separates such positions into different smaller radio maps by deploying Bluetooth hotspots at particular positions. Second, we design methods that improve the partition switching that occurs when a user leaves the detection range of a Bluetooth hotspot. Third, we propose three architectural options for placement of the computation workload. We evaluate all proposals using both simulation and walk-through experiments in two indoor environments of different sizes. The results show that our proposals are effective and efficient in achieving very good indoor positioning performance.

I. Introduction

Indoor spatial awareness attracts increasing attention from both academia and industry. Providing location based services (LBS) in indoor spaces is of practical importance as we spend considerable portions of our daily lives in indoor spaces, ranging from office buildings to transportation facilities. For example, an indoor navigation system can guide visitors through an interesting yet complex exhibition in a museum. As another example, a location-aware boarding reminder service in an airport can remind passengers of their flights and boarding gates so that they do not miss their flights or delay departures.

Indoor LBSs benefit from the availability of reliable and low-cost indoor positioning akin to how outdoor LBSs benefit from reliable and cheap GPS. Sufficiently accurate indoor positioning can enhance indoor LBSs quality and user experience. In particular, the availability of low-cost indoor positioning, which is cheap and easy to deploy and cheap to operate, has the potential of increase indoor LBS use.

In previous research, much effort has been devoted to developing indoor positioning systems based on a single wireless technology. Several types of wireless communication technologies, notably Wi-Fi [5], [7], [10], Bluetooth [4], [8], infrared [9], and RFID [12], [19] have been utilized for indoor positioning. However, each individual such technology has limitations as a means for supporting indoor positioning.

As Wi-Fi signals diffuse in space and have very large coverage, a Wi-Fi based indoor positioning system requires the use of complex additional techniques [7], [10], [15] to improve positioning accuracy. Next, RFID and Bluetooth based systems only detect objects when they are within the quite limited detection range of an RFID reader or a Bluetooth hotspot. Due to hardware and installation costs, it is typically only feasible to deploy a limited number of such devices in an indoor space. As a result, considerable positioning uncertainty is caused and needs to be properly handled [11]. Finally, infrared based positioning systems suffer from limitations including low data rate and strict directionality.

In previous research [6], we propose hybrid indoor positioning that employs both Wi-Fi and Bluetooth in one system. In such a hybrid system, Wi-Fi works as the main infrastructure to enable fingerprint based positioning, and Bluetooth (via hotspot devices) partitions the indoor space as well as a large Wi-Fi radio map. As a result, the Wi-Fi based position estimation is improved in a divide-and-conquer manner. Also, such a hybrid system can take advantage of the strengths of the individual technologies used to achieve better positioning performance.

Previous research [6] made the following assumptions.

1) Bluetooth hotspots are deployed at pivot reference positions, e.g., by doors, in an indoor space, and such pivots are of equal importance with respect to the positioning effect.
2) When a user leaves a Bluetooth hotspot, one single Wi-Fi based position estimate, obtained by searching the corresponding parts of the radio map, is sufficient to determine the user's partition.
3) A dedicated server offers online positioning estimation for mobile clients through a Web service.

This paper lifts these assumptions and significantly extends this research. First, it is important to be able to determine which reference positions are really important pivots that may influence indoor positioning positively if Bluetooth hotspots are deployed in those positions. This issue becomes more important when only few Bluetooth hotspots are available.

978-0-7695-4973-6/13 $26.00 © 2013 IEEE
DOI 10.1109/MDM.2013.30

Second, a single Wi-Fi based position estimate may not be sufficient when a user leaves a Bluetooth hotspot. When a user leaves a hotspot, one radio map part (corresponding to a partition of the space that results from the particular Bluetooth hotspot) will be selected and used for subsequent Wi-Fi based position estimation. If a wrong map part is chosen, subsequent position estimation suffers very seriously. Therefore, it is important to improve the selection of map parts.

Third, alternative architecture options should be explored. In some cases, users may be disconnected from remote servers, in which case the client-server architecture assumed so far cannot provide the positioning service. It is then desirable that the user devices themselves be able to conduct online position estimation. It is also of interest to compare the energy consumption of different architecture options.

We make the following contributions.

- We propose an algorithm for selecting pivot positions in which to deploy Bluetooth hotspots (Section III).
- We provide improved techniques for partition switching when moving objects leave the ranges of Bluetooth hotspots and enter into a particular partition (Section IV).
- We develop three different architectures for a hybrid indoor positioning that differ with respect to power consumption and coexistence properties of Wi-Fi and Bluetooth technologies (Section V).
- We conduct extensive experimental studies in real environments to test our proposals. The results suggest that our deployment algorithm is effective, and they profile the different architecture options with respect to positioning quality and energy consumption (Section VI).

We review related work in Section II and conclude the paper in Section VII.

II. Related Work

Among the different existing proposals for indoor positioning (e.g., [4], [5], [13], [19]–[22]), Wi-Fi and Bluetooth are two widely used wireless technologies. We briefly review Wi-Fi based and Bluetooth based indoor positioning, followed by hybrid indoor positioning that combines these.

A. Wi-Fi Based Indoor Positioning

Due to the very good space coverage of signals, Wi-Fi based indoor positioning systems often employs a scene analysis technique called *fingerprinting* [15], [16]. Fingerprinting makes use of Wi-Fi radio signal strengths and consists of an offline and an online phase. The offline phase always occurs before the online phase. Specifically, signal strengths from all n detectable Wi-Fi access points are collected at fixed, pre-selected indoor reference positions. For each reference position, a n-dimensional signal strength vector is obtained as a fingerprint. The fingerprints for all m reference positions form a database that is called a *radio map* for the indoor space.

Subsequently, in the online positioning phase, a user's position is estimated based on the n-dimensional signal strength vector that is obtained at the current location. The current signal strength vector is compared to all the fingerprints in the radio map, and the reference position with the best matching fingerprint is returned as the estimated user location.

Two types of methods for online position estimation can be distinguished: probabilistic methods and deterministic methods [14]. Probabilistic methods make estimates using a random process that models measurements at all reference positions. In contrast, deterministic methods compare the current signal strength vector ssv with all fingerprints in the radio map in a discrete way. The Nearest Neighbor in Signal Space (NNSS) method is a typical deterministic method. From among all reference positions in the radio map, it returns the one whose fingerprint has the shortest vector distance to ssv. According to a previous performance study [16], deterministic methods incur lower computation cost and give higher positioning accuracy than do probabilistic methods.

B. Bluetooth Based Indoor Positioning

Bluetooth hotspots can be deployed in an indoor space in order to enable positioning. However, due to the relatively limited space coverage of Bluetooth hotspots, Bluetooth based indoor positioning employs *proximity analysis* for position estimation [17], [18]. Specifically, the deployed locations of all Bluetooth hotspots are recorded as reference positions. When a user holding a Bluetooth-enabled device enters a particular hotspot's range, the device identifies the hotspot (or vice versa), and the corresponding reference position is returned as the estimate of the user's position.

It is also possible to use the entire detection range of a Bluetooth hotspot to approximate a user's current position. In that case, when multiple Bluetooth hotspots are seen simultaneously at a position, the intersection of all hotspot detection ranges is used to approximate the user's position [11]. However, such techniques are complex because they may return irregular regions as position estimates, and they involve expensive geometrical computations.

Although fingerprinting can also be applied to Bluetooth, it is very expensive to cover an entire indoor space with a sufficient number of Bluetooth hotspots.

C. Hybrid Indoor Positioning

Wi-Fi base stations have coverage regions that are up to 100 times larger than the region of a Bluetooth hotspot, but at the expense of energy consumption. Consequently, Wi-Fi is not suitable for accurately detecting presence of moving users the way Bluetooth can. Table I gives an non-exhaustive comparison of Wi-Fi and Bluetooth based indoor positioning. More details can be found elsewhere [6]. We highlight two important observations. First, it requires a high infrastructure investment to cover an entire indoor space with Bluetooth hotspots, especially when the space is large. Second, the characteristics of Wi-Fi and Bluetooth enable them to complement each other in one single, hybrid indoor positioning system. Such a hybrid system requires that user mobile devices are equipped with both Wi-Fi and Bluetooth interfaces.

Two approaches have been proposed to combine Wi-Fi and Bluetooth into one positioning system. One approach is to

Table I
NON-EXHAUSTIVE COMPARISON OF WI-FI AND BLUETOOTH BASED INDOOR POSITIONING

	Wi-Fi	**Bluetooth**
Pros	• Widely deployed and available infrastructure • Good coverage of the indoor space • Quick signal scanning	• High positioning accuracy • Simple position estimation • Low energy consumption
Cons	• Highly infrastructure dependent • Time-consuming radio map creation • Low positioning accuracy • Complex position estimation • High energy consumption	• Extra infrastructure investment • Extra specific mobile devices • Slow signal scanning

employ scene analysis for both infrastructures [3]. A combined radio map is created for both Wi-Fi and Bluetooth signals at each reference position in the offline phase. Subsequently, the combined radio map is utilized in the online phase to estimate user positions. This combined scene analysis approach needs extra offline fingerprint collection for Bluetooth. Furthermore, signal interference between Wi-Fi and Bluetooth may affect the position estimation that uses a combined radio map.

Alternatively, scene analysis and proximity analysis can be applied to Wi-Fi and Bluetooth, respectively, in one positioning system [6]. Given an indoor space with Wi-Fi deployed as the main wireless infrastructure, a limited number of Bluetooth hotspots are deployed as add-ons to divide the indoor space into disjoint partitions. As a result, the original, large radio map is divided into small ones, each of which corresponds to an indoor partition induced by the deployment of Bluetooth hotspots. This approach reduces the computation cost of online position estimation.

In particular, when a user enters the detection range of a Blueooth hotspot, as indicated by Bluetooth based proximity analysis, the user's position is simply estimated as the hotspot's position without involving signal strengths. After the user leaves a Bluetooth hotspot, a process called *partition switching* is invoked to decide the current possible partition(s) for the user. Afterwards, the online Wi-Fi based position estimation only involves the corresponding part(s) of the radio map. Searching the entire radio map only happens until the first Bluetooth hotspot is seen by the user device.

III. DEPLOYMENT OF BLUETOOTH HOTSPOTS

A. Motivation

Previous work [6], [11] proposes topological connections like doors, staircases, and narrow passages as candidate places for deployment of Bluetooth hotspots. However, such connections are important to different degrees in an indoor space, and the specific choices of places to deploy hotspots can lead to quite different performance of the resulting positioning system. This motivates us to put focus on selecting the most appropriate topological connections to deploy a limited number of available Bluetooth hotspots.

In previous experiments [6], we observed that different reference positions, although far away from each other, may share similar Wi-Fi fingerprints. We call such reference positions *resembling reference positions*. The NNSS method is not effective in distinguishing resembling reference positions. As a consequence, such positions cause an increased error distance[1], especially when they are within the same partition that corresponds to the same part of the radio map. By carefully deploying Bluetooth hotspots, we separate resembling reference positions into different partitions, thus dividing their Wi-Fi fingerprints into different radio map parts. Thus, subsequent position estimation is less likely to encounter resembling reference positions, which then improves the positioning accuracy.

On the other hand, it is desirable that the original radio map is divided into parts that have similar numbers of reference positions. Otherwise, the search cost becomes unbalanced among the different radio map parts when a user is in different indoor partitions. This then results overall in higher search and computation costs.

B. Deployment Algorithm

The overall procedure for deploying Bluetooth hotspots is summarized in Algorithm 1. The algorithm takes three parameters as input: the original radio map R for the entire indoor space, the graph g of all reference positions in R,[2] and the number n of available Bluetooth hotspots. The algorithm employs a while-loop to select n pivots from g for deploying the n available hotspots. Each selected pivot also partitions the (sub)graph into two subgraphs. In each iteration, the algorithm works in three steps as follows.

First, with the help of a max-heap (line 2), the subgraph g_i (initially g) with the largest number of reference positions is chosen to be partitioned (line 4). For the g_i, N Wi-Fi based position estimates are obtained by simulation or real runs.

Second, the estimated results are evaluated (lines 8–17). For each wrong estimate, the error distance together with the true and estimated positions are recorded (lines 8–12). Subsequently, the average error distance is calculated for each unique pair of true and estimated positions, and the pair (p_1 and p_2) that leads to the maximum average error distance is obtained (lines 13–17).

Third, a balanced partitioning is done (lines 18–31). Specifically, positions in g_i are clustered with respect to p_1 and p_2 (lines 18–19). Size based balancing is done if necessary (lines 20–21). Subsequently, a boundary position is found for either cluster, and the one with the higher average error

[1] *Error distance* [6] denotes the distance between a user's true position and the reference position that is returned as the estimated position. It can be measure as a Euclidean distance or an indoor walking distance. We measure the latter in the experiments.

[2] Such a graph is constructed based on the connectivity between reference positions in the radio map [10].

distance in the estimate is chosen as the pivot p_v (lines 22–27). Finally, a Bluetooth hotspot is deployed at p_v, and the current subgraph's radio map is partitioned accordingly (lines 28–31).

Algorithm 1 deployBTHotspots(original radio map R, graph of reference positions g, number of available Bluetooth hotspots n)

1: $count \leftarrow n$; $\mathcal{R} \leftarrow \{R\}$
2: initialize a max-heap $\mathcal{H}$; enheap($\mathcal{H}$, $\langle g, |g|\rangle$)
3: **while** $count > 0$ **do**
4: $\quad g_i \leftarrow$ deheap($\mathcal{H}$)
5: $\quad$ **if** $|g_i| <$ MIN_SIZE **then**
6: $\quad\quad$ **break**
7: $\quad$ make N Wi-Fi based position estimates for g_i using g_i's radio map $R_i \in \mathcal{R}$ $\triangleright$ *simulation or run in the real setting*
$\quad$ *// evaluate estimated results*
8: $\quad results \leftarrow \emptyset$
9: $\quad$ **for** each estimated result **do**
10: $\quad\quad$ **if** the true position $p_t \neq$ the estimated position p_e **then**
11: $\quad\quad\quad \varepsilon_j \leftarrow$ the distance between p_t and p_e
12: $\quad\quad\quad$ add $(p_t, p_e, \varepsilon_j)$ to $results$
$\quad$ *// get the maximum average error distance and the corresponding pair of reference positions*
13: $\quad \varepsilon_m \leftarrow 0$; $p_1 \leftarrow$ null; $p_2 \leftarrow$ null;
14: $\quad$ **for** each unique pair of (p_t, p_e) in $results$ **do**
15: $\quad\quad$ get the average error distance $\overline{\varepsilon_j}$ in $results$
16: $\quad\quad$ **if** $\overline{\varepsilon_j} > \varepsilon_m$ **then**
17: $\quad\quad\quad \varepsilon_m \leftarrow \overline{\varepsilon_j}$; $p_1 \leftarrow p_t$; $p_2 \leftarrow p_t$;
$\quad$ *// balanced partitioning*
18: $\quad g_{i1} \leftarrow \{p \mid p \in g_i \wedge |p, p_1| \leq |p, p_2|\}$
19: $\quad g_{i2} \leftarrow \{p \mid p \in g_i \wedge |p, p_2| \leq |p, p_1|\}$
20: $\quad$ **if** $|g_{i1}|/|g_{i2}| \notin [0.8, 1.2]$ **then**
21: $\quad\quad$ move a corresponding boundary positions from the larger set to the other set
22: $\quad p_{v1} \leftarrow$ the position in g_{i1} that is closest to g_{i2}
23: $\quad p_{v2} \leftarrow$ the position in g_{i2} that is closest to g_{i1}
24: $\quad$ **if** p_{v1} has the higher average error distance in the N position estimates **then**
25: $\quad\quad p_v \leftarrow p_{v1}$; remove p_v from g_{i1}
26: $\quad$ **else**
27: $\quad\quad p_v \leftarrow p_{v2}$; remove p_v from g_{i2}
$\quad$ *// Bluetooth hotspot deployment*
28: $\quad$ deploy a Bluetooth hotspot at p_v
29: $\quad$ enheap($\mathcal{H}$, $\langle g_{i1}, |g_{i1}|\rangle$); enheap($\mathcal{H}$, $\langle g_{i2}, |g_{i2}|\rangle$)
30: $\quad$ partition radio map R_i accordingly to R_{i1} and R_{i2}
31: $\quad \mathcal{R} \leftarrow (\mathcal{R} \setminus \{R_i\}) \cup \{R_{i1}, R_{i2}\}$
32: $\quad count \leftarrow count - 1$

C. Discussion

Online Wi-Fi based position estimates are made in Algorithm 1 (line 7) to identify resembling reference positions. Such estimates can be done in simulation using sufficient Wi-Fi signal strengths that are pre-collected at arbitrary positions in the indoor space. They can also be run in the real setting. Alternatively, we can execute online position estimates for a sufficient number of times at sufficient, arbitrary positions, upon which we pass the positioning results to Algorithm 1 that in turn makes use of the results to identify resembling reference positions (lines 8–12). Therefore, the Bluetooth deployment algorithm provides flexibility to indoor positioning system builders.

Another issue in Algorithm 1 is that of balanced partitioning (lines 18–27). We control the size ratio between two subgraphs to be created (line 20). We set the default ratio to 1.2, which can be tuned by the algorithm user in practice. The boundary positions (line 22) refer to those positions that are in g_{i1}, but are closer to g_{i2} than others in g_{i1}, and likewise those in g_{i2}. The proximity is measured as an indoor walking distance if two reference positions are connected; otherwise, planar Euclidean distance is used for the sake of simplicity.

IV. Enhanced Partition Switching

It is of high importance in the hybrid positioning approach that partition switching is correct. This process identifies the partition of a moving user who is leaving the range of a Bluetooth hotspot. In previous work [6], the partition switching worked by always selecting the two partitions adjacent to the most recently seen Bluetooth hotspot until a Bluetooth hotspot is seen again.

While this approach is safe, it is preferable to be able to choose one single partition because this reduces the number of candidate reference positions to consider in the online Wi-Fi based position estimation. Put differently, the computation cost can be reduced by identifying a single partition in the partition switching. However, the subsequent positioning accuracy is jeopardized if the single identified partition is incorrect.

We proceed to study how to select one single partition when a user leaves the range of a Bluetooth hotspot. A naive approach is to make a single position estimate using one scan of the Wi-Fi signal once no Bluetooth hotspot is seen. If the single, estimated position belongs to partition X, only the radio map part corresponding to X is then used in subsequent Wi-Fi based position estimation. That means that subsequently estimated positions can only be among those positions that are reference positions in partition X.

Partition switching based on a single position estimate is not reliable due to many reasons. For example, in an environment where Wi-Fi signals tend to fluctuate and are noisy, random signal strengths may cause incorrect partition switching. We thus propose two methods that improve the naive approach. We argue that increasing the number of estimates can help select partitions more correctly. Therefore, the two proposed methods make n ($n \geq 1$)[3] estimates. Each method scans Wi-Fi signals, followed by a corresponding position estimate, a total of n times. It then selects the next partition. The methods differ in how they make use of the n estimates to select the next partition. They are both independent of the number of reference positions.

The *frequency method* just counts how frequent a reference position occurs in the n estimates and uses the most frequent one to select the next partition. Ties are resolved by a random choice.

Next, the *weighted method* works as follows. Different weights are assigned to the n estimates, and the correspon-

[3] We use a small, odd number for n, as a large n may cause unnecessary cost. Finding an optimal n is difficult as it involves factors such as user speed, distance between hotspots, and Wi-Fi scanning time.

ding partition is decided for each of the n estimates. Then a weighted score is calculated for each possible partition, and the partition with the highest score is selected. Suppose the n estimated positions are p_1, p_2, ..., and p_n in this temporal order. We assign weights w_i ($i \in 1, 2, \ldots, n$) to them accordingly. Each estimated position n_i falls into a particular partition r_j. For each possible partition r_j that appears in the n estimates, we calculate its score according to the following formula.

$$score(r_j) = \frac{\sum_{p_i \in r_j} w_i}{n} \cdot r_j \tag{1}$$

An example is shown in Table II. Here we have two possible partitions with identifiers 1 and 2. The score for partition 1 is $(3+4+5)/5 \cdot 1 = 2.4$, and that for partition 2 is $(1+2)/5 \cdot 2 = 1.2$. As a result, partition 1 is chosen as the one that contains the current user position.

Table II
ESTIMATED POSITIONS WITH WEIGHTS

Estimated position (p_i)	4	5	6	6	6
Weight (w_i)	1	2	3	4	5
Corresponding partition (r_j)	2	2	1	1	1

V. ARCHITECTURE OPTIONS

A. System Architecture Considerations

When designing a hybrid indoor positioning system, special attention should be paid to the power consumption of handheld devices. The simultaneous use of multiple wireless technologies and online Wi-Fi position estimation can be power-consuming, which can quickly deplete battery power. Thus, efficient energy management is of high importance in regard to usability. In addition, the speed of receiving location estimates or scan readings can be also crucial while building a reliable and accurate system.

Other aspects that can shape the architecture of the positioning system include the coexistence of Bluetooth and Wi-Fi. In previous work [6], we observed the phenomenon of inferior Wi-Fi signal perception. More precisely, performing two scannings simultaneously resulted in Wi-Fi signals being partially blocked on some devices.

Considerations such as the above render it attractive to enable several positioning system architectures.

B. Three System Architectures

A hybrid indoor positioning system fundamentally employs the general client-server architecture. As depicted in Figure 1, there are four kinds of components in a hybrid indoor positioning system, namely mobile devices, an application server, Wi-Fi access points, and Bluetooth hotspots. The server and mobile devices have separate databases, which are utilized in different architecture options.

We present three architectures that involve different degrees of interactions between client and server and also differ in the workloads delegated to the mobile devices. These options are listed and explained in Table III.

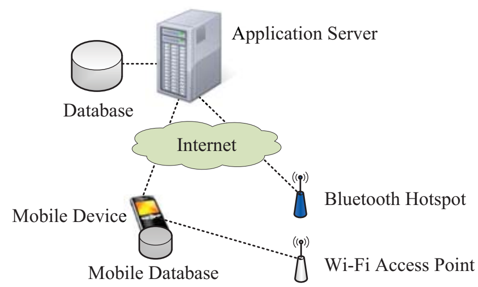

Figure 1. Architecture components

Table III
ARCHITECTURE OPTIONS

	Bluetooth Scanning	Online Position Estimation
Thin client	Client side	Server side
Thick client	Client side	Client side
Medium client	Bluetooth hotspots	Client side

In the *thick-client architecture*, Wi-Fi and Bluetooth scanning are performed on the mobile device. After Wi-Fi readings are obtained through scanning, online position estimation is carried out locally on the client. Only an initial interaction with server is needed to download the database of fingerprints and an indoor space map to the mobile device.

The *thin-client architecture* is designed similarly, except that the online position estimation is done by the server. The client sends its Wi-Fi readings to the server via the Internet. The server compares the readings with the radio map and a best match is returned to the client. Thus, the Wi-Fi based online position estimation occurs on the server. In this architecture, we are interested in position estimation delay observed by the client. This delay is caused by the communication and the server side computation.

Finally, the *medium-client architecture* is designed with the purpose of reducing the interference due to simultaneous Wi-Fi and Bluetooth scanning. Thus, Bluetooth discovery is moved to the Bluetooth hotspots. When a hotspot detects a device in its range, it sends information to the server. To reduce traffic, the server side database is updated only when a device enters or leaves the range of a hotspot.

In this architecture, only Wi-Fi scanning and position estimation occur on the client device. The client acquires Bluetooth readings from the server's database when it needs to. The advantage of this model is its ability to increase the scanning frequency on the Bluetooth hotspots. Usually, mobile devices cannot operate with a high frequency of Bluetooth scanning, which is recommended to be as high as once each 10 or 14 seconds [2]. This architecture decreases client power consumption.

In all of the architectures, we assume that Bluetooth and Wi-Fi scanning are performed independently and asynchronously. In Section VI-D, we evaluate all the architectures with respect to client side energy consumption, positioning delay, and the coexistence of Bluetooth and Wi-Fi.

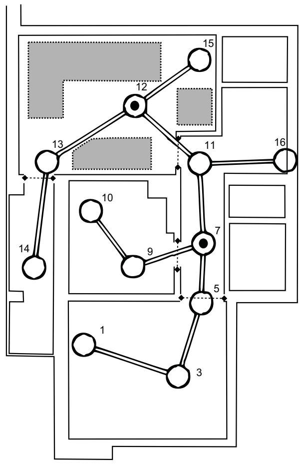

Figure 2. Indoor environment A

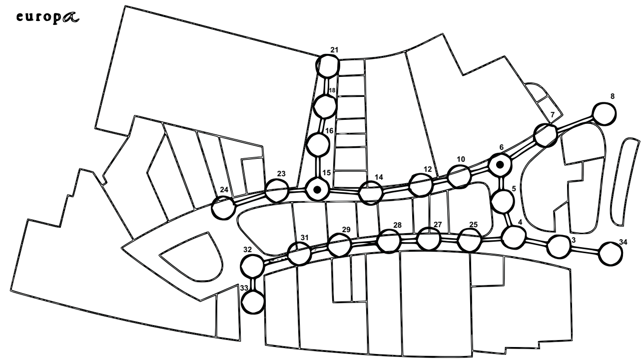

Figure 3. Indoor environment B

VI. Experimental Studies

A. Experimental Setup

We performed experiments in two indoor space environments. The first one (environment A) is a relatively small multi-floor apartment building, where each floor occupies 50 m^2. Its floor plan is shown in Figure 2. A total of more than 30 Wi-Fi access points are visible from some floor during the entire experiment; however, one Wi-Fi scan sees at most 10 access points. We collected fingerprints in 12 different reference positions that were 3 meters apart on average. The Wi-Fi signals in this environment are relatively stable due to the small space.

The second indoor space environment (environment B) is larger and more complex. It is shown in Figure 3 and is a large shopping and entertainment center called "Europa." One floor in B exceeds 500 m^2 and has multiple shop sections. The total number of Wi-Fi access points in B exceeds 60, but at most 20 are visible in one scan. The Wi-Fi signals in this environment exhibit more visible fluctuations. We picked 23 reference positions, and the average distance between two reference positions is around 10 meters. We did the experiments at the end of the day when the flow of people in B was low, which brings about less effects on the wireless signals.

In both environments, we performed 30 to 40 Wi-Fi scans in each reference position and stored the signal strength data as radio maps. In the online positioning phase, we walked through pre-defined routes multiple times. The route in environment A is $\langle 1, 3, 5, 7, 9, 10, 9, 7, 11, 16, 11, 12, 15, 12, 13, 14\rangle$, where each number represents a reference position, and the route in environment B is $\langle 24, 23, 15, 16, 18, 21, 18, 16, 15, 14, 12, 10, 6, 7, 8, 7, 6, 5, 4, 3\rangle$. The route in B excludes other positions that are in relatively huge (sub)spaces. Experiments in those spaces require too much time to finish.

In the experiments, we used a mobile device running Android 2.3.1, namely a Samsung Nexus S smartphone with a 1 GHz ARM Cortex-A8 processor and 512 MB RAM. The capacity of its standard battery is 1500 mA.

On the server side we used a HP NW8440 PC with an Intel Core 2 Duo 2.0GHz processor and 3GB RAM. In addition, two mobile Bluetooth hotspots with detection range of up to 10 meters were used in the experimental environments. Furthermore, for the medium-client architecture (see Section V), we used an Asus EEE PC 901 with a few Bluetooth USB dongles to act as a "smart" Bluetooth hotspots that can communicate with the server and send data about detected mobile devices. We wrapped aluminum foil around those dongles to render their detection ranges comparable to those of the other Bluetooth hotspots used in the experiments.

B. Experiments on Bluetooth Hotspot Deployment

We first evaluate the Bluetooth hotsopt deployment algorithm (Algorithm 1) in environment A. As indicated in Figure 2, the algorithm deploys the two hotspots at positions 7 and 12. The manual deployment puts the hotspots at positions 5 and 11, as they are doorways. For each setting, we conduct hybrid indoor positioning using the weighted partition switching method (Section IV). The resulting average error distance for each reference position is reported in Figure 4.

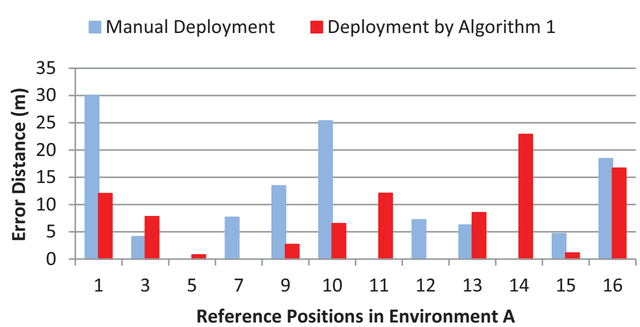

Figure 4. Effect of Bluetooth hotspot deployment

For most reference positions, the deployment by Algorithm 1 achieves smaller error distances. In particular, the average error distance in the manual deployment is 9.75 meters, while that in the algorithmic deployment setting is 7.57 meters. Therefore, Algorithm 1 is effective in identifying resembling reference positions and distributing them into different partitions. This way, the Wi-Fi based online positioning is more accurate.

In environment B, we first deployed two Bluetooth hotspots at reference positions 10 and 25 (see Figure 6) and got an average error distance of 4.45 meters. Then we deployed the hotspots at references positions 5 and 15, and got an average error distance of 5.99 meters. Finally, we used Algorithm 1, deploying the hotspots at positions 6 and 15. This resulted in an average error distance of 2.63 meters. The Bluetooth hotspot deployment algorithm thus clearly helps improve the overall indoor positioning accuracy.

C. Experiments on Partition Switching

We tested the two partition switching methods (Section IV) in both indoor environments.

We report the results of a simulation in environment A in Figure 5, and we report the results of a real-life walk-through in environment B in Figure 6. Here, "Hybrid-n" means that n position estimates are made for switching. We use Hybrid-3 in environment A and Hybrid-5 in environment B, since the distances between Bluetooth hotspots (and reference positions) are larger in environment B.

From Figure 5, we see that hybrid indoor positioning outperforms the pure Wi-Fi positioning in most cases. In particular, pure Wi-Fi positioning leads to an average error distance of 3.15 meters across all reference positions, while Hybrid-1 achieves 2.49 meters and Hybrid-3 with weighted partition switching achieves 1.75 meters. Thus, the weighted

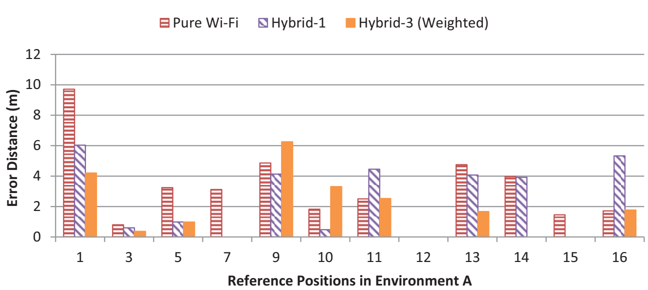

Figure 5. Positioning accuracy in environment A

partition switching method reduces the error distance by approximately 30% and 40%, compared to Hybrid-1 and pure Wi-Fi, respectively. These results indicate that the weighted method is very effective in finding the correct partition in the experiments. Note that in simulation-based experiments, we see zero error distance in reference positions with Bluetooth hotspots (positions 7 and 12).

According to the results shown in Figure 6, the performance of all methods degrades in environment B, where the wireless signals are considerably less stable due to the flows of people. In such an environment, the Hybrid-1 method performs the worst, as one Wi-Fi position estimate is insufficient to switch to the correct partition, which yields long error distances. When Bluetooth and Wi-Fi co-exist in a dynamic environment,

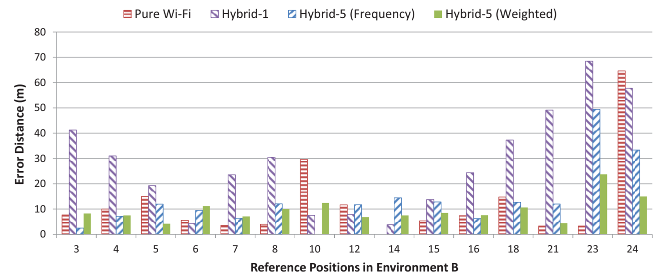

Figure 6. Positioning accuracy in environment B

one Wi-Fi scanning and position estimation faces high fluctuation and instability. We see that significantly improved performance is achieved by the frequency and weighted Hybrid-5 methods. In most cases, weighted partition switching is able to select the correct partition and thus achieves smaller error distances.

D. Experiments on Architecture Options

We test the three architectures, namely thin, thick, and medium client, as presented in Section V. Thick client is the standard architecture used in all previous experiments, where Bluetooth and Wi-Fi scanning are performed simultaneously on the mobile device. In the thin-client architecture, the online Wi-Fi based position estimation is done on the server side. The medium-client architecture is an in-between option, where Bluetooth scanning is performed externally by the "smart" Bluetooth hotspots. Consequently, Bluetooth-related data is received from the server by the client device in the medium-client architecture. We investigate the effects of the architectures on mobile device power consumption, position estimation delay, and Bluetooth and Wi-Fi coexistence.

1) Power Consumption: There are two ways to measure power consumption on a mobile device: using an external power meter or using a built-in battery sensor. According to previous research [23], using battery voltage sensor on an Android phone shows accurate results compared to using an external power meter. Therefore, we have installed the Battery Monitor Widget Application [1] on the mobile device to perform power consumption measurements.

With the help of this application, battery status data was collected in the background while different architectures were tested. The logged data includes the date, mA, percentage of battery left, and mV. Specifically, mA measures the electric current strength in the device,[4] and mV measures the remaining voltage of the device's battery. According to Joule's Law and Ohm's Law, $P = V \cdot I$. Thus, the electric power

[4]The absolute value of mA indicates the strength. When the device battery is discharging due to running applications, mA values are negative. When the battery is charging, the mA values are positive. Our experiment results show negative mA values to indicate the amount of energy used by the device.

consumption is proportional to the voltage V and the current I. Therefore, we provide our observations about power consumption based on the mA and mV values collected during the tests.

All experiments were conducted for a 2-hour period. The power consumption measurements were logged every minute, so that the battery status change can be seen clearly. During the whole period, the device display was always active, and other programs were turned off.

For comparison purposes, we also report the power consumption for the following three settings: *Wi-Fi*, meaning that the hand-held devices only activates normal Wi-Fi networking without positioning; *BT*, meaning that the device only activates normal Bluetooth connections without positioning; and *None*, meaning that the device only activates the display.

As shown in Figure 7, the device consumed the least power in setting *None*, which is as expected. Although no scanning is performed in this setting, the screen display still requires power to keep running. Due to the different characteristics of Bluetooth and Wi-Fi, Bluetooth requires the least power. The difference is also seen in Figure 7, where Bluetooth sometimes uses 40mA less current than does Wi-Fi.

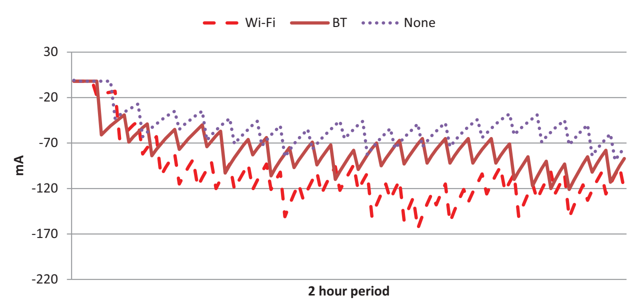

Figure 7. mA measurements for settings without indoor positioning

The indoor positioning applications generally consume more energy as is seen in Figure 8. Overall, the thick-client architecture consumes the most device energy. This is reasonable as a thick client carries out not only the online position estimation, but also the Bluetooth scanning. The energy consumption by the thin client is generally close to that of the medium client. This is because a thin client is complementary to a medium client. A thin client scans the Bluetooth hotspots, but delegates the position estimation to the server, whereas a medium client delegates the scanning to the Bluetooth hotspots, but estimates the positions.

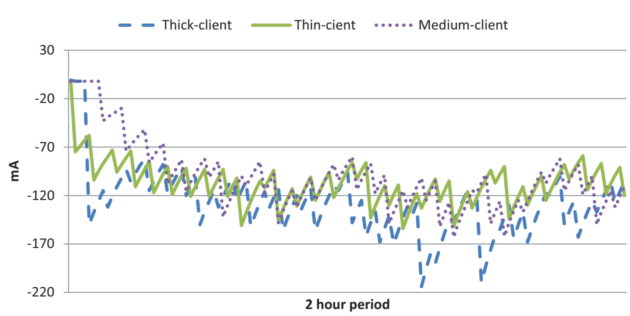

Figure 8. mA measurements for settings with indoor positioning

To give a more illustrative picture of how the architectures consume power differently, Figure 9 plots the accumulated mA values each minute during the 2 hours starting from the beginning of the 2-hour period. Clearly, the thick client is the most energy-consuming, which is due to its heavy computation load and scanning. Further, the medium client uses less energy than the thick client. This is attributed to the extra Bluetooth scanning done by the thick client.

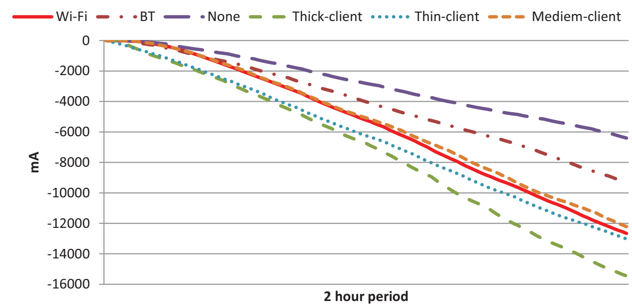

Figure 9. Summed mA for all settings

The remaining mV values of the device, i.e., the voltage of the device battery, during the 2-hour experiment period are plotted in Figure 10. From the results, we see that a thick client dries up the battery the fastest. Nevertheless, the energy consumption trends among the thick client, the medium client, and a pure Wi-Fi client are very similar. This indicates that a hybrid indoor positioning system with both Wi-Fi and Bluetooth consumes energy in a reasonable manner, since it does not impose significant extra burden to the device battery compared to a pure Wi-Fi client without positioning.

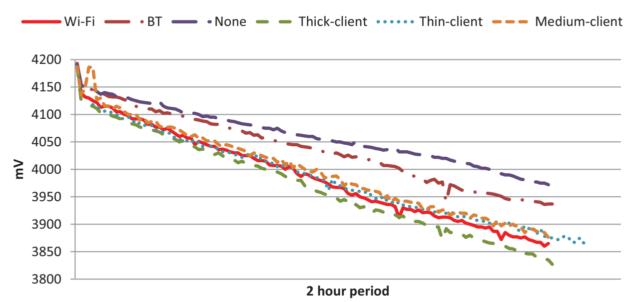

Figure 10. mV for all settings

We also investigate the average battery power consumption using an online power model for smartphones [23]. This model estimates the battery power consumption between two given times t_1 and t_2 ($t_2 > t_1$). In particular, the model measures the state of discharge (SOD), which is the percentage of the rated battery energy that has been discharged, at each time.

As a result, the power consumption is estimated as follows.

$$P = E \times \Delta SOD \tag{2}$$

Here, E is the battery power capacity, and ΔSOD is the difference of SOD from t_1 to t_2.

The battery used in our hand-held device is new, and its power capacity E is 1500mA. In the 2-hour test, the Battery Monitor Widget [1] allows us to measure the SODs for each architecture using the before and after values. We calculate the estimated average battery power consumption using Equation 2, and report the results in Table IV.

Table IV
AVERAGE BATTERY POWER CONSUMPTION DURING TWO HOURS

Options	ΔSOD_1	Consumed Power (mA)
None	19%	285
Bluetooth	24%	360
Wi-Fi	31%	465
Medium client	29%	435
Thin client	32%	480
Thick client	34%	510

Compared to the client without indoor positioning (None), the pure Wi-Fi client consumes 63% more power, whereas the thick client consumes 79% more power. Among all hybrid indoor positioning architectures, the lowest power consumption is by a medium client that delegates the Bluetooth scanning to hotspots.

By subtracting 285mA from the power consumption of Bluetooth and Wi-Fi, we get 75mA and 180mA, respectively. This suggests that Wi-Fi in general is twice as power-hungry as Bluetooth. It is also clear that a hybrid indoor positioning system incorporating both Wi-Fi and Bluetooth does not necessarily consume more power than a purely Wi-Fi based system. According to the experimental results, the medium client actually consumes less power than the pure Wi-Fi client. Even the thick client that performs all major computations locally consumes only slightly more power than the pure Wi-Fi client.

These results show that a hybrid indoor positioning system is feasible from a power consumption perspective.

2) Positioning Delay: The *positioning delay* is the time from when a new position estimate is requested to when that estimate is available to the user. This delay occurs generally due to the communication and computation involved in online positioning. As the positioning delay affects the user experience more noticeably than the execution time of a positioning algorithm, we investigate how the delay is affected by the system architecture and the number of reference positions. For a thin-client architecture where the positioning is done by the server, the delay involves network transfer time.

To investigate the causes of the positioning delay, we vary the number of reference positions from 12 to 62 and obtain four different radio maps. We compare the positioning delay between the thin client (server-side position estimation) and the thick client (client-side position estimation). We perform 100 positioning estimations and report average delays in Figure 11.

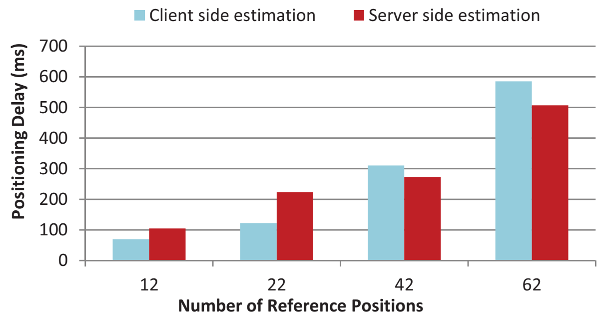

Figure 11. Online positioning delay

The results support the following observations. Client-side position estimation is faster when the number of reference position is not large because the mobile device can handle all the small radio maps. Server-side position estimation is advantageous when the number of reference position is large (42 or higher in the experiment). This indicates that the mobile device does not cope well with larger radio maps, whereas the server is capable of fast online position estimation, which makes network transmission pay off.

For the thin-client architecture, we also investigate how the number of visible Wi-Fi access points affects the amount of data that the client sends to the server. The results are reported in Figure 12. The amount of transmitted data does not reach 1KB even with 9 visible access points. Further, the amount of transmitted data increases in a very moderate, linear fashion as the number access points increases. These results indicate that the thin-client architecture is efficient and scalable in terms of data transmission between client and server.

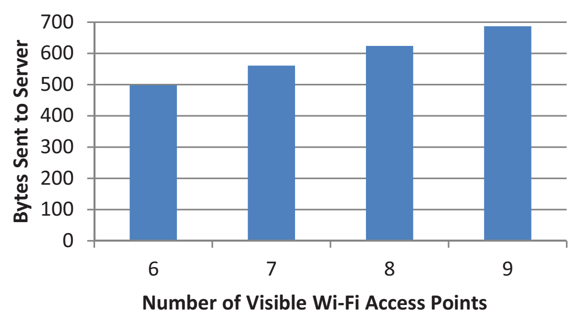

Figure 12. Transmitted data amount from client to server

3) Bluetooth and Wi-Fi Coexistence: In previous research [6], we observed interference between simultaneous Wi-Fi and Bluetooth scanning on the same device. Here, we investigate this interference in the different architectures. We test the five cases listed in Table V. Specifically, case a) is a pure Wi-Fi client without Bluetooth. Cases b) and c) concern a medium client for which Bluetooth scanning is done by the hotspots. In cases e) and f), the client does the Bluetooth scanning. Also, only in case e), the client device is connected to a Wi-Fi access point before the test is started. For each case, we perform 200 to 250 Wi-Fi scans. Figure 13 reports the average number of visible access points.

Table V
EXPERIMENTAL CASES

Case	Wi-Fi	Bluetooth Scanning	Bluetooth Distance
a	Not connected	No Bluetooth	–
b	Not connected	By hotspots	0.3–0.4 meters
c	Not connected	By hotspots	2–3 meters
d	Not connected	By client	2–3 meters
e	Connected	By client	2–3 meters

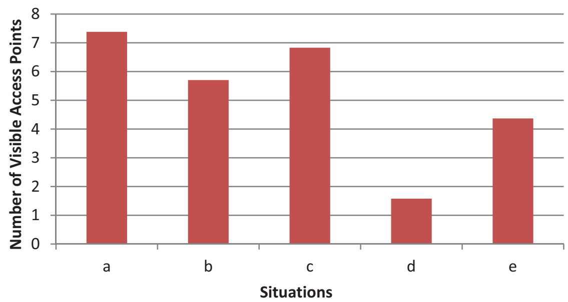

Figure 13. Number of visible Wi-Fi access points

One interesting observation concerns the impact of Bluetooth scanning performed from different sources. According to Figure 13, the effect on Wi-Fi visibility is less if Bluetooth scanning is done by hotspots (cases b) and c)) instead of by the client (cases d) and e)). More specifically, compared to a pure Wi-Fi setting (case a)), the Wi-Fi visibility is reduced by 22.72% if the Bluetooth hotspots are within 0.3–0.4 meters from the client (case b)). The Wi-Fi visibility is reduce only by 7.47% if the hotspots are further away (case c)).

The worst result is seen in case d), where fewer than 2 Wi-Fi access points are detected. When the Wi-Fi scanning is initiated from a cold start on a device, the sensitivity to simultaneous Bluetooth scanning on the device is the highest. The situation is improved when the Wi-Fi scanning is initiated beforehand, as indicated by the result for case e).

VII. CONCLUSION

We propose techniques that improve hybrid indoor positioning that uses both Wi-Fi and Bluetooth. We propose an algorithm that identifies locations in which to deploy a limited number of Bluetooth hotspots in order to achieve the best positioning. We design methods that improve the partition switching that occurs when a user leaves the detection range of a Bluetooth hotspot. In addition, we also present three architectures for a hybrid indoor positioning system that serve different user needs and user hardware constraints.

We conduct extensive experiments in real settings to evaluate our proposals. The results show that the Bluetooth deployment algorithm is effective and contributes to better positioning accuracy when compared to manual deployment. The results also show that the new partition switching methods work better than the straightforward method for finding the correct indoor partition, and thus the corresponding smaller radio map. In addition, the results profile the three architectures with respect to device energy consumption and positioning quality. The results verify that hybrid indoor positioning is energy efficient and effective in returning accurate estimated positions.

ACKNOWLEDGMENTS

This research was supported in part by the Indoor Spatial Awareness project, funded by the Korean Land Spatialization Group and BK21 program, as well as by Geocrowd, funded by the European Commission under grant agreement number 264994.

REFERENCES

[1] Battery Monitor Widget. http://www.3c71.com/android/?q=node/1.

[2] G. Ananthanarayanan and I. Stoica. Blue-Fi: enhancing Wi-Fi performance using Bluetooth signals. In *Proc. MobiSys*, pp. 249–262, 2009.

[3] E. Metola, S. Aparicio, P. Tarrio, and J. R. Casar. Comparison of localization methods using calibrated and simulated fingerprints for indoor systems based on Bluetooth and WLAN technologies. In *Proc. MADRINET*, 10 pages, 2009.

[4] G. Anastasi, R. Bandelloni, M. Conti, F. Delmastro, E. Gregori, and G. Mainetto. Experimenting an indoor Bluetooth-based positioning service. In *Proc. ICDCS Workshops*, pp. 480–483, 2003.

[5] P. Bahl and V. N. Padmanabhan. Radar: an in-building RF-based user location and tracking system. In *Proc. INFOCOM*, pp. 775–784, 2000.

[6] A. Baniukevic, D. Sabonis, C. S. Jensen, and H. Lu. Improving Wi-Fi based indoor positioning using Bluetooth add-ons. In *Proc. MDM*, pp. 246–255, 2011.

[7] S. Bell, W. R. Jung, and V. Krishnakumar. WiFi-based enhanced positioning systems: accuracy through mapping, calibration, and classification. In *Proc. ISA*, pp. 3–9, 2010.

[8] S. Feldmann, K. Kyamakya, A. Zapater, and S. Lue. An indoor bluetooth-based positioning system: concept, implementation and experimental evaluation. In *Proc. ICWN*, pp. 109–113, 2003.

[9] D. Hallaway, T. Höllerer, and S. Feiner. Coarse, inexpensive, infrared tracking for wearable computing. In *Proc. ISWC*, pp. 69–78, 2003.

[10] R. Hansen and B. Thomsen. Efficient and accurate WLAN positioning with weighted graphs. In *Proc. MOBILIGHT*, pp. 372–386, 2009.

[11] C. S. Jensen, H. Lu, and B. Yang. Graph model based indoor tracking. In *Proc. MDM*, pp. 122–131, 2009.

[12] G.-Y. Jin, X.-Y. Lu, and M.-S. Park. An indoor localization mechanism using active RFID tag. In *Proc. SUTC*, pp. 40–43, 2006.

[13] H.-Y. Kang, J.-S. Kim, and K.-J. Li. Strack: tracking in indoor symbolic space with RFID sensors. In *Proc. GIS*, pp. 502–505, 2010.

[14] M. B. Kjaergaard. A taxonomy for radio location fingerprinting. In *Proc. LoCA*, pp. 139–156, 2007.

[15] M. B. Kjærgaard and C. V. Munk. Hyperbolic location fingerprinting: A calibration-free solution for handling differences in signal strength (concise contribution). In *Proc. PerCom*, pp. 110–116, 2008.

[16] T.-N. Lin and P.-C. Lin. Performance comparison of indoor positioning techniques based on location fingerprinting in wireless networks. In *Proc. IWCMC*, pp. 1569–1574, 2005.

[17] H. Liu, H. Darabi, P. Banerjee, and J. Liu. Survey of wireless indoor positioning techniques and systems. *IEEE TSMC, Part C: Applications and Reviews*, 37(6):1067–1080, 2007.

[18] K. Muthukrishnan, M. E. Lijding, and P. J. M. Havinga. Towards smart surroundings: Enabling techniques and technologies for localization. In *Proc. LoCA*, pp. 350–362, 2005.

[19] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil. Landmarc: indoor location sensing using active rfid. *Wirel. Netw.*, 10:701–710, 2004.

[20] S. C. Spinella, A. Iera, and A. Molinaro. On potentials and limitations of a hybrid WLAN-RFID indoor positioning technique. *International Journal of Navigation and Observation*, 11 pages, 2010.

[21] R. Want, A. Hopper, and V. Gibbons. The active badge location system. *ACM TOIS*, 10(1):91–102, 1992.

[22] Z. Xiang, S. Song, J. Chen, H. Wang, J. Huang, and X. Gao. A wireless LAN-based indoor positioning technology. *IBM J. Res. Dev.*, 48(5/6):617–626, 2004.

[23] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. P. Dick, Z. M. Mao, and L. Yang. Accurate online power estimation and automatic battery behavior based power model generation for smartphones. In *Proc. CODES/ISSS*, pp. 105–114, 2010.