Learning and Recognizing the Places We Go

Jeffrey Hightower[1], Sunny Consolvo[1], Anthony LaMarca[1], Ian Smith[1], and Jeff Hughes[2]

[1] Intel Research Seattle
Seattle, WA, USA
[2] University of Washington
Computer Science & Engineering
Seattle, WA, USA

Abstract. Location-enhanced mobile devices are becoming common, but applications built for these devices find themselves suffering a mismatch between the latitude and longitude that location sensors provide and the colloquial place label that applications need. Conveying my location to my spouse, for example as (48.13641N, 11.57471E), is less informative than saying "at home." We introduce an algorithm called BeaconPrint that uses WiFi and GSM radio fingerprints collected by someone's personal mobile device to automatically learn the places they go and then detect when they return to those places. BeaconPrint does not automatically assign names or semantics to places. Rather, it provides the technological foundation to support this task. We compare BeaconPrint to three existing algorithms using month-long trace logs from each of three people. Algorithmic results are supplemented with a survey study about the places people go. BeaconPrint is over 90% accurate in learning and recognizing places. Additionally, it improves accuracy in recognizing places visited infrequently or for short durations—a category where previous approaches have fared poorly. BeaconPrint demonstrates 63% accuracy for places someone returns to only once or visits for less than 10 minutes, increasing to 80% accuracy for places visited twice.

1 Introduction

Devices that can automatically figure out their geographic coordinates are becoming common. Many mobile phones are now location-enhanced due to U.S. E911 and European E112 initiatives requiring location capability for calls placed to emergency services. The Global Positioning System (GPS) covers most of the earth's surface, and GPS chipsets are continually decreasing in cost, making it feasible for them to be integrated into many mobile devices. Technologies like RightSpot [1] and Place Lab [2] have shown that beacon-based location can allow a device to compute its position with high availability throughout someone's day—including indoors and in environments like the "canyons" formed by high-rise buildings where GPS is unreliable. Applications such as mapping and way-finding are straightforward to build using any of these location technologies.

M. Beigl et al. (Eds.): UbiComp 2005, LNCS 3660, pp. 159–176, 2005.

Many emerging location-enhanced applications, however, want colloquial place names like "Home," "Work," "Movie Theater," or "Tony's Pizzeria" instead of latitude and longitude coordinates. An example is a dynamic instant messaging (IM) client that can set its status message to its user's current place. Using place names as the IM status is likely more informative to IM buddies than raw coordinates like 48.13641N, 11.57471E. We call this disconnect between the coordinates that devices provide and the place names emerging applications desire as the problem of *moving from location to place*.

One step in the move from location to place is to use databases such as Yahoo! Yellow Pages, Microsoft MapPoint, or governmental map and census repositories. Each of these databases can translate a coordinate into a corresponding business name, street address, map image, geographic feature, political subdivision, or other label. This process is called *geocoding*. Applications like tour guides, recommender systems, and franchise store locators (*e.g.* "Where is the nearest McDonald's Restaurant to my current position?") have been built using geocoding. A shortcoming of geocoding is that without the context of a specific query, geocoded place names can be as challenging to interpret as raw coordinates—particularly if the names are shared with others or reviewed in hindsight in a log of the places someone went. For example, revealing to my spouse that I am currently in the residential area 142 meters north of the cash machine located at the corner of Broadway and Main, although a precise description of my position, is less useful than saying I am at home. The problem is that geocoded information, like a raw coordinate, does not correspond to someone's mental model of their personal routine nor to the terminology they use when discussing the places they go.

To start to overcome these challenges in geocoding, the research community has proposed ways of enabling people's mobile devices to automatically *learn* the places they go and then to *recognize* whenever they return to those places. This paper's contributions are on this topic. Note that this work is not about automatically assigning semantics or names (*e.g.* "home") to places, but in providing the mechanism for learning the physical destinations in someone's life and detecting whenever their devices return to those places.

Place learning algorithms take as input a sensor log gathered from a mobile device and produce as output a list of the places the device went. The sensor information collected about each of these places is called a *waypoint*. A recognition algorithm uses a place's waypoint to detect when the device returns to that place. Figure 1 illustrates the learning and recognition cycle. An effective learning algorithm can do two operations:

1. **Segment** a sensor log into times when the device was in a stable place and assign a waypoint.
2. **Merge** waypoints which are captured from repeat visits to the same place.

Likewise, an effective recognition algorithm has two capabilities:

1. **Recognize** when the device returns to a known place using a waypoint list.
2. **Recognize** when the device is *not* in a place. We refer to this state as mobile.

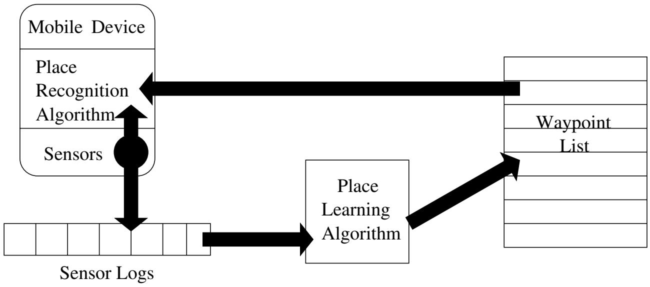


Fig. 1. The flow of data through the place *learning* and *recognition* cycle

This paper contributes a new place learning and recognition algorithm called *BeaconPrint*. BeaconPrint uses 802.11 and GSM radio response-rate fingerprints to learn and recognize places more accurately than previous approaches. We demonstrate BeaconPrint's effectiveness with a thorough comparative evaluation to three published place algorithms using one month of multi-sensor trace logs collected from each of four people. To supplement these algorithmic tests, we also conducted a survey study about the places people go. Because the data collectors for the algorithmic experiments were all members of our research team, the survey study allows us compare the places they went during the month to the number, type, and visit frequency of places reported by survey participants. We show that this comparison finds no obvious idiosyncrasies in the data collectors' habits, thus lending credence to the claim that BeaconPrint generalizes.

2 Related Work

Place learning algorithms can be divided into two classes: *geometry* and *fingerprint*. Geometric algorithms produce coordinates, circles, or polygons to describe the significant places the algorithm believes the user went. Examples of geometric place algorithms are recurring GPS dropout used in the comMotion system from Marmasse et al. [3], Ashbrook and Starner's GPS dropout windows plus hierarchical clustering [4], and the sensor-agnostic temporal point clustering contributed by Kang et al. [5]. Fingerprint algorithms, in contrast, produce a waypoint list with no geography. Fingerprint waypoints are a "signature" of each place which allows the device to detect when it returns to the place, but provides no direct information about where that place is geographically located. Fingerprint place algorithms include BeaconPrint, the graph clique-based GSM cell fingerprints of Laasonen et al. [6], and cell-ID matching which has been studied in depth by Trevisani and Vitaletti [7].

A variety of sensors can produce input logs for learning. These sensors range from traditional technologies like WiFi network interfaces and GPS receivers that are found on many mobile devices to sensors like accelerometers, barometers, or altimeters that are custom built or currently not found on most devices.

Although data from more exotic sensors may support accurate place learning, BeaconPrint shows that high accuracy is possible with commodity hardware. Microphones are a commodity sensor and there is promising work on using recordings of ambient audio as a sensor log [8]. However, continuous mobile recording of audio (or video for that matter) presents social and legal challenges that have hampered their adoption.

BeaconPrint is structurally similar to Krumm and Hinckley's NearMe location technology [9]. NearMe is a service allowing pairs of devices to compare the 802.11 radio signatures they hear to decide whether they are in physical proximity to one another. Although learning and recognizing places is a different task then determining proximity, the similarities between NearMe and BeaconPrint will be described in Section 3.3.

In evaluating BeaconPrint we compare it to the Ashbrook and Starner, comMotion, and Kang et al. algorithms. The following subsections discuss our implementation and use of each of these algorithms. A post facto discussion of each algorithm appears in Section 4 following the experimental results. We elected to exclude the GSM fingerprint approach of Laasonen et al. from our evaluation. This algorithm is clever but complex due to what we see as an artificial restriction. The complexity in the Laasonen approach seems to come about because programming interfaces on GSM phones only allow access to the unique ID of the single tower to which the phone is connected. This restriction is an artificial one imposed by the phone manufacturers and service providers. Industrial-grade or low-level GSM modems can see multiple nearby towers. Using only a single beacon requires significant complexity to build, merge, traverse, and interpret hand-off graphs—ultimately to approximate the capability that 802.11 and low-level GSM radios get for free: performing full parallel scans for all nearby radio beacons. BeaconPrint uses parallel scans to achieve high accuracy in learning and recognizing places with a significantly simpler approach.

2.1 Ashbrook and Starner's GPS Dropout plus Hierarchical Clustering Algorithm (A&S)

Ashbrook and Starner's algorithm [4], henceforth referred to as A&S, learns places people go in order to construct a Markov model predicting where they might go next. A&S exploits the variable availability of GPS in real environments to learn a person's places. It segments the GPS sensor log by marking positions at the beginning of every window of at least t minutes where the person's GPS receiver loses the satellite signal or indicates a speed continually below 1 mile per hour. Both of these situations indicate the person probably stopped or entered a building where the GPS signals cannot penetrate. These candidate positions are then merged using a variant of k-means clustering. The clustering is repeated hierarchically to identify sub-places. Clustering produces the waypoint list. Our experience agrees with their observation that choosing $t = 10$ minutes (9.6 actually) maximizes the number of correctly learned places.

As published, A&S does not explicitly discuss or evaluate recognition, however, it seems clear that using A&S waypoints for recognizing places means using live GPS data to lookup the nearest waypoint that is no further away than the radius at which the place was learned during clustering. If no place meets the criteria then the device is labeled as mobile. This distance-constrained containment search is the recognition approach used for evaluating A&S in this paper.

2.2 The comMotion Recurring GPS Dropout Algorithm

The comMotion system [3] is a GPS-equipped wearable or hand-held device which can present to-do lists or other information that is relevant to the person's current place, for example, a list of home chores when the person arrives home. Systems like comMotion are often called remembrance agents.

The comMotion system learns places using GPS in a different way than A&S. In comMotion, a place is a position where the GPS signal is lost three or more times within a given radius. The segmentation step extracts these timestamped GPS drop points, and the merge step groups them into places. The paper does not reveal what radius was used, however our iterative refinement experiments revealed 100 meters to be a good choice.

To perform recognition with comMotion for our experiments, we use live GPS data to lookup the nearest 3-time GPS drop point place that is no further away than the radius. If no place is found, the device is labeled as being mobile.

2.3 Kang et al.'s Sensor-Agnostic Temporal Point Clustering Algorithm (KSAC)

Both of the previous algorithms depend on properties of the GPS satellite signals to work properly. To avoid this dependence, Kang et al. designed a sensor-agnostic algorithm using temporal point clustering. We refer to this approach as KSAC. KSAC takes as input a stream of timestamped coordinates derived from any location system. It performs the segmentation and merging steps simultaneously using time-based clustering. For our experiments, we use the suggested clustering parameters of time $t = 300$ seconds and distance $d = 300$ meters.

KSAC's technology independence, although a good characteristic for an algorithm to have, presents the challenge of first turning sensor logs into coordinates. Using GPS alone provides poor results because GPS coverage is low and spotty in daily-life usage—a fact exploited by A&S and comMotion. Therefore, we duplicated the approach used in KSAC by first running all of our traces through the Place Lab location system (acquired from `www.placelab.org`) to produce coordinates. For recognition using KSAC, we ran our testing traces through the Place Lab system and used the coordinates it emitted at each time step to search for the nearest cluster within the cluster separation radius. If no cluster is found, the device is labeled as mobile.

3 The BeaconPrint Algorithm

Previous algorithms like A&S use exceptions experienced by the underlying location system to learn places. For example, A&S logically reasons that the loss of a GPS signal means the person entered a building, and entering a building means the person is at a significant place. Reactive approaches, however, can be sensitive to errors in the technology. For example, if a GPS signal is lost and later acquired in the same position on multiple days as someone's walk to work takes them into an urban canyon, A&S could erroneously identify this point as a place.

In contrast, BeaconPrint continually gathers statistics about the radio environment around the mobile device and uses this data to learn, merge, and recognize waypoints. This proactive approach allows BeaconPrint to recognize both indoor and outdoor places and be more robust to errors as our experiments will show. Although not completely sensor agnostic like KSAC, BeaconPrint uses as its sensors the commodity 802.11 or GSM wireless radios which are already built into most mobile devices.

WiFi access points and GSM towers both broadcast unique identifiers for the purpose of discovery and hand-off. For example, WiFi access points transmit periodic beacon frames containing the AP's unique MAC address. We refer to these fixed radio sources as *beacons*. Mobile devices can periodically scan for the IDs of nearby beacons. WiFi devices can scan without connecting to the network or listening to any data traffic and can see the IDs of access points even if the network is WEP encrypted. A timestamped log of these beacon scans is the input to BeaconPrint's learning phase.

BeaconPrint operates as follows: Define a time window w. Stable scans seen continuously for at least w indicate a significant place. A stable scan is one that contains no beacon IDs not seen in time w. The fingerprint (a histogram of all the beacons seen during w) is the waypoint that allows BeaconPrint to recognize return visits to the place. After constructing the waypoint list, BeaconPrint then identifies and merges similar waypoints (those inferred to have come from repeat visits to a single place). The recognition phase compares the live fingerprint seen by the device to histograms in the waypoint list. By examining the degree of the match, BeaconPrint is able to present a weighted and ordered list of device's most likely current places.

3.1 Learning

Making BeaconPrint work in practice requires a bit more logic than the basic outline presented above. The problem with the basic version is that it cannot distinguish beacons seen infrequently while a device is in the same physical place (*e.g.* a beacon that is only detected every 100th scan because of signal attenuation or multipath effects) from new beacons seen as the device is physically leaving a place. Consequently the basic version fails to learn some places and the ones it does find are erroneously divided into multiple places by the detection of low response-rate beacons. To fix this problem, we define a certainty parameter c with range $0 \ldots c_{max}$ and divide the window w into dwell increments of length $d = w/c_{max}$. If d time passes without seeing a new beacon, i.e., the scans are stable, then start collecting a fingerprint. This fingerprint's certainty c increments every additional dwell d that passes without seeing a new beacon. When

$c = c_{max}$, the fingerprint becomes valid. Seeing a new beacon decrements c. If c reaches 0, the fingerprint is recorded if valid or else discarded. Beacons are considered new if they are not seen for at least w and are not in the current fingerprint.

3.2 Parameters in BeaconPrint

BeaconPrint takes two parameters: window size w and confidence depth c_{max}. Window size is the adjustment offered by most place learning algorithms to specify how long the person must stay somewhere for it to be considered a place. Choosing w under 2 minutes works poorly in practice because spurious places such as stoplights and crosswalks where the person stopped only briefly will be discovered.

The other parameter, c_{max}, determines the dwell time and confidence. Choosing c_{max} too low results in the problems of the basic version where infrequently seen beacons cause places to be erroneously fragmented. Choosing c_{max} too high causes distinct places with a short physical travel distance between them to be incorrectly grouped together because an insufficient number of new beacons to end the fingerprint is seen during transit between the places. When the person starts moving and leaves a place, the algorithm should detect new beacons and end the fingerprint gathering within the time length of one dwell window $d = w/c_{max}$. Therefore, if r is the average arrival rate of new beacons seen when moving between places, this goal will be met if:

$$\begin{gathered} \textit{new beacons during one dwell time} > \textit{max confidence} \\ r \cdot \frac{w}{c_{max}} > c_{max} \\ c_{max} < \sqrt{rw} \end{gathered}$$

Choosing $c_{max} = \sqrt{rw}$ works well in practice for any window size.

To investigate the rate r, we examined our trace logs for the new beacon arrival rate in portions of the log where the person was mobile. We first tried to determine mobility only considering portions of the log with a valid GPS lock indicating non-zero ground speed, however this approach proved unreliable because the amount of time with a GPS lock is very low with receivers carried by a typical person in his daily life. Instead, we determined mobility by extracting portions of the log where each data collector's ground-truth diary (see Section 4.1) indicated they were in transit between two places. This results in an r value of 0.0631Hz, 0.0055Hz, 0.0686Hz for WiFi, GSM, and both together. These values are used in subsequent experiments in this paper. For example, using GSM and WiFi with a time window w of 2 minutes (120 seconds) specifies a c_{max} of 3.

It may be possible to update r in real time based on beacon arrival rate. However, detecting when the person is mobile without supervised labeling is difficult. Although BeaconPrint itself detects mobility, the feedback loop created by using the algorithm to tune its own r value presents a challenge. Alternatively, a classification method such as the one provided by the Opportunity Knocks to infer and predict the person's mode of transportation (*e.g.*, car, bus, bike, or walk) [10] might make it possible to compute different r values for each mode and improve BeaconPrint further. It is not clear, however, that adding dynamism to r is even needed. If we repeat the rate analysis broken down by each individual person, the consistency of r across our four data collectors suggests that instead of online adjustments, it is appropriate to simply set r based on the beacon technologies in use (WiFi, GSM, or both) and perhaps according to the general beacon density of the region (*e.g.*, beacons per square kilometer in the greater San Fransisco Bay Area for all local residents). Examining these ideas in greater detail is future work.

3.3 Fingerprints: Merging and Matching

Place learning algorithms using fingerprints often choose signal strength as their metric. Instead, BeaconPrint follows the conclusions offered by LaMarca et al. [2] and constructs its fingerprint using a response-rate histogram where response-rate is (1-beacon loss rate). Response-rate is an aggregate statistic based on MAC layer characteristics, signal fading, multipath, and interference. LaMarca et al. showed that when a device is stationary, the percent of scans which see a particular beacon can be more effective in predicting the distance to that beacon than the signal strength values reported by the wireless network interfaces of both WiFi cards and GSM phones.

Choosing response-rate fingerprints adds the final piece of logic to the learning algorithm: When c_{max} reaches 0 and a valid fingerprint is recorded, beacons with less than c_{max} entries in the fingerprint histogram are discarded. This approach trims outliers and discards the new beacons which caused the fingerprint to end when the user left the place being fingerprinted.

In implementing the NearMe location server [9], Krumm and Hinckley studied four ways to detect if two wireless signatures are similar:

1. Number of beacons the fingerprints have in common.
2. Spearman rank-order coefficient of the ordered relative signal strengths.
3. Sum of squared differences of the signal strengths.
4. Number of beacons the fingerprints do not have in common.

BeaconPrint applies the first technique, extended to operate on response-rate histograms instead of simple sets. During the merging phase of BeaconPrint, if the overlapping set of beacons in the two fingerprints contains more than 68% of the weight of both histograms (68% is 1 standard deviation of a standard normal), then the fingerprints are deemed similar and merged. For each new fingerprint, all pairs of fingerprints are iteratively compared and merged until the set stabilizes.

During recognition, BeaconPrint considers the device to be in a place if more than 1 standard deviation of the weight in the observed fingerprint overlaps with any part of the place's fingerprint. The fingerprint currently seen by the device thus might match multiple learned places. In this case, the list of recognized places can be ordered by the weight of the shared beacons in each matching place fingerprint. If no fingerprints match, the device is labeled as mobile.

4 Algorithm Evaluation

To evaluate BeaconPrint, we compare it with the three algorithms described in Section 2.

4.1 Data Collection

To accurately evaluate BeaconPrint and overcome any startup effects, we collected a substantial amount of multi-sensor trace data as well as ground-truth about the places people actually went and the times they were there.

Sensor Logs We collected 24x7 GPS, WiFi, and GSM trace logs for one month from each of four members of our research team as they went about their normal lives. We choose members of our team as the data collectors instead of recruiting external participants because the data collection task required substantial effort as well as technical expertise to diagnose and fix any problems. Furthermore, the data is inherently sensitive making it challenging to recruit external volunteers. However, as we will describe in Section 5, we also conducted a small, in depth survey study of the places people go using recruited participants to compare our data collectors to people not on our research team.

Each data collector carried a backpack containing a laptop, mobile phone, and 16-hour battery. The laptop had attached to it a standard WiFi network card and a GPS unit which was modified to send data into and draw power from the laptop's PC card slot. As described previously, WiFi access points and GSM towers both have unique identifiers. Our data collection software scanned for these unique identifiers at 2Hz and kept a timestamped log of all the nearby beacons heard in each scan. GPS data arrived in a serial stream and was logged at 1Hz. The GSM scanning occurred on the mobile phone and was relayed to the laptop over a Bluetooth data link. In total, we collected 3.4GB of multi-sensor data amounting to over 1,440 hours of sensor logs. On later analysis we discovered that one of our data collector's laptops experienced periodic hardware failures and its logs were segmented and incomplete. This data was excluded from our analyses.

Ground-Truth To collect ground-truth, each data collector was given a clip-on watch and a small paper notebook to carry with them everywhere. In the notebook, they kept a diary of the name and time they entered and left every place they went during the month. At the end of the data collection these diaries were coded and each data collector used map software to indicate the coordinates of every unique place in their diary. These diaries and maps provide the ground-truth information about the coordinates of the actual places the data collectors went as well as the actual times they arrived and left those places. Finally, each data collector completed the survey study described in Section 5.

Data Collector Demographics Our data collectors are assigned the pseudonyms Adam, Bob, and Charles.

Adam is a parent of 2 children. Many of his places involve driving his kids to school, doctors, restaurants, and extra curricular activities. He usually walks to work but chooses to drive about one time in four. He stops for coffee on his way to work every day and typically goes out for coffee at least once more during the day. He usually eats out for lunch.

Bob is a challenging person for any place learning algorithm. He lives in an area of dense urban high-rises. The places he goes for errands and entertainment around his home are tightly clustered and close to one another. Bob typically takes public transit to work which is outside the urban area. He occasionally drives to work. He eats out for lunch at a wide variety of restaurants nearly every day. Bob frequently goes out for coffee, although less often than Adam.

Charles walks to work every day but frequently drives to specialty shops and other destinations located several miles from his house. He packs his lunch many days but is also a regular patron at a small set of restaurants near his home and work. Charles is a frequent traveler, particularly on weekends. During our logging he made two train trips to the same destination over 100 miles from his home and a plane trip to a destination over 600 miles from his home. Obviously Charles' log has a gap during the time he was in flight since radio technology is prohibited on airplanes.

4.2 Experimental Results

We divided each data collector's sensor logs in half. Each algorithm was given the first half of the log to learn the data collector's places. The second half of the log was then used to evaluate how well the algorithm could recognize based on the places it learned. Places which the data collector only visited in the second half of their log were time-spliced out. Table 1 summarizes the results of this experiment. It shows the percent of time each algorithm correctly identifies the data collector's actual place. For example, the percent of time an algorithm correctly identifies place p is the total amount of time the algorithm predicts p divided by the total amount of time the data collector actually spent at p. The total percentage appearing in the table is the aggregate of this statistic across all the data collector's places.

Because our data collectors, like most people, spend most of their time at home, at work, or in transit, Table 1 also shows the correctness statistics when home and work and then when all three of these periods are omitted from the analysis. Factoring out these periods reveals the effectiveness of each algorithm at recognizing less frequented places.

For the total percent of the time each algorithm was incorrect, we also analyzed the data further to understand why it erred. To present these results, the table shows a breakdown of the percent of time the algorithm made each of the four possible errors:

Wrong means the data collector was in a place but the algorithm reported they were in a different place.

Missed means the data collector was in a place but the algorithm reported they were mobile.

Spurious is the result of a learning error. A spurious error occurs when the data collector was in a place but the algorithm reported they were in another place which does not correspond to anywhere they actually went in learning. Merging or clustering errors are usually the cause of spurious recognition errors.

False Positive means the data collector was actually mobile but the algorithm reported they were in a place.

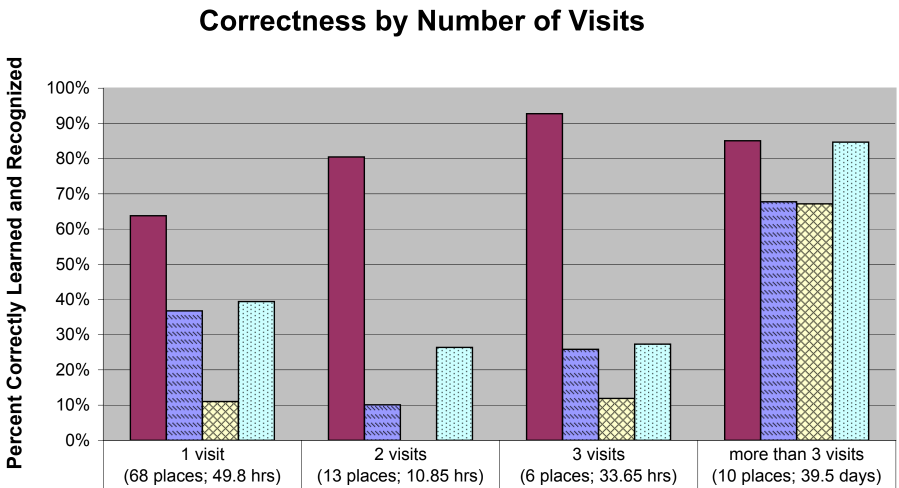
BeaconPrint performs well. It has the highest overall accuracy for all three data collectors in all three situations presented in the table. The percent of time it chooses the wrong place is also strictly the lowest. BeaconPrint is less accurate for Bob when compared to its performance for Adam and Charles, but the other algorithms are significantly worse on Bob's data. Bob's urban neighborhood has a density of nearly 2000 beacons per square mile. Beacon density in and of itself can be a good thing for a fingerprint algorithm like BeaconPrint, but density combined with unpredictable radio propagation is not. The concrete and steel canyons formed by urban buildings wreak havoc on the signals of GPS and other radio technologies. Unpredictable radio propagation makes it difficult to get a GPS lock and challenging to acquire clean beacon fingerprints with good place discrimination capability. Consequently, learning and recognition errors rise for Bob. Despite these challenges, BeaconPrint still performs much better for Bob than we anticipated.

BeaconPrint's improvement in the accuracy of recognizing places other than home and work is particularly notable as shown by the second and third sections of the table. To investigate these infrequent places further we examined the percent of time the algorithm was correct, broken down by the number of visits the data collector made to each place and the amount of time they spent in each place. These results are presented in Figures 2 and 3. Values in parentheses are the total number of places and amount of time our data collectors spent in places with the given visit frequency or dwell characteristics. The right-most set of columns on both graphs is obviously dominated by time spent at home and work.

From these graphs it is clear that BeaconPrint provides a significant improvement in the ability to recognize places visited infrequently or for short visits. These infrequently visited places are also quite numerous. When examined strictly by count, they make up the majority of the places our data collectors went. We believe BeaconPrint's ability to recognize places that are visited infrequently or for short durations is the most significant contribution of the overall accuracy improvements offered by the approach. We will examine this conclusion further in Section 5 in the context of our survey study.

Person	Adam				Bob				Charles			
Algorithm	BP	A-S	cM	KSAC	BP	A-S	cM	KSAC	BP	A-S	cM	KSAC
	Full Analysis											
Correct	**96.39%**	**92.29%**	**90.50%**	**91.05%**	**85.12%**	**25.52%**	**49.85%**	**65.38%**	**91.13%**	**80.94%**	**79.33%**	**85.95%**
Incorrect	**3.61%**	**7.71%**	**9.50%**	**8.95%**	**14.88%**	**74.48%**	**50.15%**	**34.62%**	**8.87%**	**19.06%**	**20.67%**	**14.05%**
Wrong	0.25%	3.84%	2.18%	1.38%	4.68%	43.74%	6.29%	25.98%	0.38%	5.58%	4.10%	1.27%
Missed	0.98%	1.29%	5.41%	1.00%	5.81%	0.15%	40.44%	2.78%	2.17%	3.26%	6.90%	7.19%
Spurious	0.70%	0.01%	0.00%	1.44%	1.42%	24.42%	0.58%	0.05%	3.35%	6.46%	6.95%	3.34%
False Pos	1.68%	2.57%	1.91%	5.14%	2.97%	6.17%	2.84%	5.82%	2.97%	3.76%	2.71%	2.25%
	Home and Work Omitted											
Correct	**72.26%**	**58.58%**	**43.52%**	**29.34%**	**60.41%**	**21.05%**	**40.10%**	**24.87%**	**68.19%**	**26.17%**	**23.90%**	**42.68%**
Incorrect	**27.74%**	**41.42%**	**56.48%**	**70.66%**	**39.59%**	**78.95%**	**59.90%**	**75.13%**	**31.81%**	**73.83%**	**76.10%**	**57.32%**
Wrong	1.98%	14.05%	7.96%	11.63%	1.72%	21.04%	4.20%	24.95%	2.14%	14.92%	12.81%	6.84%
Missed	5.63%	4.51%	31.50%	3.49%	16.30%	0.44%	45.33%	14.88%	9.32%	2.29%	15.85%	34.21%
Spurious	5.93%	0.00%	0.00%	12.13%	6.60%	18.10%	0.00%	0.10%	3.65%	35.50%	32.19%	3.63%
False Pos	14.19%	22.86%	17.02%	43.41%	14.97%	39.37%	10.37%	35.19%	16.70%	21.12%	15.25%	12.63%
	Home, Work, and Transit Time Omitted											
Correct	**72.98%**	**60.77%**	**17.05%**	**45.64%**	**56.00%**	**29.02%**	**11.17%**	**28.61%**	**77.17%**	**20.35%**	**8.05%**	**32.49%**
Incorrect	**27.02%**	**39.23%**	**82.95%**	**54.36%**	**44.00%**	**70.98%**	**88.83%**	**71.39%**	**22.83%**	**79.65%**	**91.95%**	**67.51%**
Wrong	3.95%	29.69%	16.73%	23.20%	3.08%	37.73%	7.54%	44.60%	3.23%	22.55%	19.36%	10.34%
Missed	11.23%	9.54%	66.21%	6.97%	29.13%	0.79%	81.29%	26.61%	14.09%	3.46%	23.94%	51.69%
Spurious	11.83%	0.00%	0.00%	24.19%	11.79%	32.46%	0.00%	0.18%	5.51%	53.64%	48.64%	5.48%

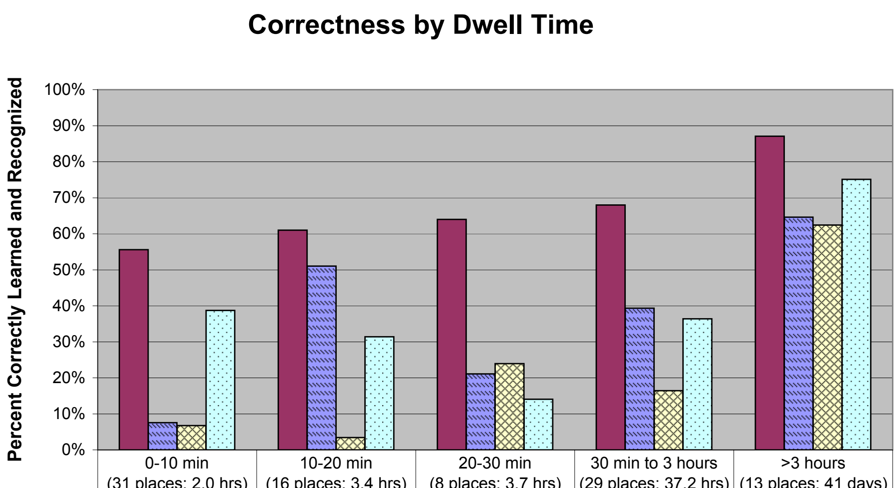
Table 1. Percent of time each algorithm correctly identifies the data collector's places. BeaconPrint has the highest correctness and the least percent of the time choosing the wrong place



	1 visit (68 places; 49.8 hrs)	2 visits (13 places; 10.85 hrs)	3 visits (6 places; 33.65 hrs)	more than 3 visits (10 places; 39.5 days)
BeaconPrint	63.79%	80.46%	92.78%	85.09%
A&S	36.74%	10.08%	25.85%	67.74%
comMotion	11.03%	0.00%	11.89%	67.17%
KSAC	39.42%	26.36%	27.33%	84.68%

Total Number of Visits to the Place

Fig. 2. Success of each algorithm by the number of visits to the place



	0-10 min (31 places; 2.0 hrs)	10-20 min (16 places; 3.4 hrs)	20-30 min (8 places; 3.7 hrs)	30 min to 3 hours (29 places; 37.2 hrs)	>3 hours (13 places; 41 days)
BeaconPrint	55.60%	61.00%	64.00%	68.00%	87.10%
A&S	7.57%	51.01%	21.11%	39.31%	64.60%
comMotion	6.73%	3.41%	23.95%	16.44%	62.42%
KSAC	38.70%	31.38%	14.06%	36.37%	75.10%

Total Time Spent in the Place

Fig. 3. Success of each algorithm by time spent at the place

comMotion The comMotion algorithm was reasonably successful at learning and recognizing the 2 or 3 places our data collectors visited regularly, but worked poorly for other places. The requirement for the user to go to a place at least three times before it can be recognized is overly restrictive. The small amount of success comMotion appears to have in recognizing a one-visit place in Figure 2 is due to a coincidental learning error which turned out to be beneficial during recognition.

Although comMotion uses GPS loss to find places, we found that the availability of GPS during transit between places was far less continuous than comMotion seems to assume. Bob, our data collector who lives in the urban high-rise area, would often be mobile for a substantial period without achieving GPS lock. He could travel from his home to the bus stop, ride the bus for 20 minutes, and then walk a few blocks to work with little to no GPS lock—not surprising since he lacked a clear view of the satellites for almost the entire journey. Adam and Charles fared a bit better, probably because both of them commute by foot in less urban areas.

A&S As in comMotion, the low availability of GPS limited the amount of available data, although this does not hurt the accuracy of A&S as much as might be expected. Filtering for GPS dropout windows and then also for GPS readings with velocity less than 1mph is a clever enhancement. For example, both Adam and Charles live in wood-frame houses where GPS had weak and intermittent but not lost coverage. Because of its velocity filter, A&S found these places quickly.

Hierarchical clustering of locations and sublocations did not prove useful. It was difficult to choose a constant pair of radii for the hierarchical clustering. Indeed, the approach yielding the highest accuracy was to only cluster locations (not sublocations) using the single radius of 200 meters—close to the sublocation radius proposed in the original paper. The success of single-level clustering is logical because our data collectors do not have readily identifiable groups of nearby places connected by longer commutes. Their places are dense in certain areas but there are many scattered places in between. To capture these weaker place hierarchies it may be possible to alter A&S to choose different radii for locations and sublocations in different regions of the map, however this is not a trivial change because it would require a different clustering algorithm. The published clustering algorithm depends on having a single constant radius per level.

KSAC The KSAC algorithm performed reasonably well in our experiments. Although its performance lagged behind BeaconPrint, it was able to recognize infrequently or briefly visited places somewhat better than A&S or comMotion because of its use of temporal clustering. KSAC does seem to have a weakness in its ability to distinguish mobility as seen by the high number of false positive errors. We believe this issue arises because its clustering tries to account for large regions resulting in a good set of place waypoints, but a lack of sharp demarcation between places.

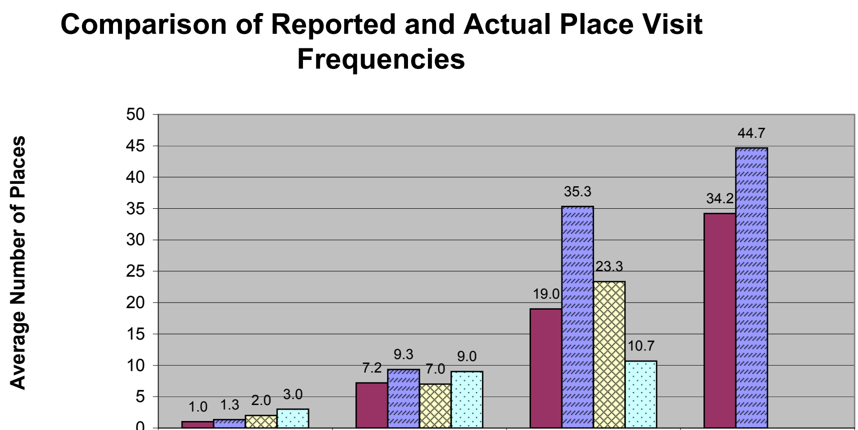
5 Survey Study

The data collectors mentioned previously were all members of our research team. Therefore the question remains whether they, and hence the sensor logs they collected, are representative of a wider population. To address this issue and supplement the algorithmic evaluations, we conducted a survey study investigating the places people go.

We recruited 6 participants of varying ages and professions. Note that we use the term *participants* here to distinguish the people who participated in this survey study from the data collectors who wore the backpacks and gathered the sensor trace logs. The data collectors were also participants and completed this study bringing the total number of survey study participants to 9. Recruited participants consisted of 2 males and 4 females, ranged in age from 25 to 40, and were drawn from a variety of professions including homemaker, scientist, writer, and retail clerk. All participants were from the Seattle area. Participants received a US $75 American Express gift cheque as a thank-you for their participation.

The study asked participants to recall all the places they go at least twice a year (once per year for medical-related places) and within 50 miles of home. We used two techniques to help reduce the recall bias associated with survey studies. First, we used categorical prompting. We structured the survey as a packet where each page was a broad category such as restaurants, medical, shopping–food, shopping–non-food, etc. The head of each page then contained check-boxes for many subcategories which the participant could check off one-by-one as they completed the page to remind them about the types of places they go within categories (*e.g.*, Asian-fusion vs. French restaurants). Second, we allowed participants to complete the survey packet on their own time over the course of several days, thus allowing them the opportunity to fill in places periodically as they remembered more. By using a carefully designed prompting exercise and leisurely homework, we hope that we were able to gather a more complete set of places then would have been generated from a straight recall task with participants brought to our lab. However, we realize the list of places generated likely represents a lower bound, as participants may have forgotten about some places they go that met the criteria and may have deliberately left off other places. For each place, participants recorded the subcategory name, the name they use to refer to the place, the frequency with which they visit the place, and a description or address of its location.

We used participants' data to search for idiosyncrasies in our data collectors' places or habits that would cast suspicion on their data logs and hence our conclusions about the success of BeaconPrint. Fortunately, there are no suspicious mismatches to be found. In fact, the participants' survey data, the data collectors' survey data, the data collectors' diary records of the places they went during data logging, and even the BeaconPrint algorithm's predictions are very consistent as shown by Figure 4. This graph demonstrates that places our data collectors went are not distributed in visit frequency significantly different from the places external participants report going. This comparison supports the claim that there is no reason to believe the BeaconPrint results will not generalize beyond the three data collectors whose sensor logs were used in the algorithmic analyses. The fourth column of this graph shows no data for diaries and BeaconPrint because sensor data collection only occurred for one month.



	at least 1/day	at least 1/week, but less than 1/day	at least 1/month, but less than 1/week	less than 1/month
Participants' Surveys	1.0	7.2	19.0	34.2
Data Collectors' Surveys	1.3	9.3	35.3	44.7
Data Collectors' Diaries	2.0	7.0	23.3	
BeaconPrint Algorithm	3.0	9.0	10.7	

Visit Frequency

Fig. 4. The average number of places visited by visit frequency. Participants in the survey study reported an average of 72.3 total significant places

Analyzing Figure 4, we hypothesize that the data collectors might have reported more infrequent places (columns 3 and 4) because, being members of our research team who frequently participated in our "hallway" discussions about this work, they were primed to recall all the odd places they go once or twice a year. The data collectors' diaries, which recorded where they actually went, however, are quite consistent with the participants' surveys. BeaconPrint clearly does well. The fact that the third grouping is lower for BeaconPrint can be explained by the characteristics of the experiment. BeaconPrint's recognition ability was only tested on the evaluation half of the month-long sensor logs, so multiplying the value in this case by 2 is probably appropriate, thus raising it to a comparable level.

6 Future Work

Armed with statistics about the amount of time BeaconPrint is correct, we plan to conduct additional user studies to understand the ways in which people respond to wrong, missed, spurious, and false positive errors. We hope to understand which types of errors are benign and which are egregious in the context of different applications and scenarios from the users' perspective. This new study will both inform our design of place-enhanced applications and help us focus future work on improving the BeaconPrint algorithm.

Algorithmically, we plan to extend BeaconPrint to support partially supervised learning of places. Active Campus Explorer (ACE) [11] has shown the value of allowing users to click or tap the screen of their mobile device to indicate their actual position when the location system is unsure. ACE uses this correction to improve its accuracy whenever the user is again near the corrected position. ACE has shown that these manual corrections have low cognitive overhead on users and are a well liked feature. We believe that manual correction techniques extend quite naturally to BeaconPrint and have the potential to significantly improve its accuracy by avoiding any ambiguity about when to collect a fingerprint for a place. We also imagine that the mechanism for manual correction in BeaconPrint could be combined with an interface for adding semantics or names (*e.g.*, "home") to the places BeaconPrint learns.

7 Conclusion

The BeaconPrint algorithm presented in this paper addresses the problem of automatically learning the places a person takes their mobile device and then being able to recognize whenever the device returns there. BeaconPrint uses 802.11 and GSM response-rate histograms to learn and recognize places using radio fingerprints. Using 802.11 and GSM radios as its sensors allows BeaconPrint to run on commodity hardware, since many mobile devices have these radios built in. BeaconPrint can begin to recognize a place after the first time the devices goes there. We evaluated BeaconPrint using 1 month of multi-sensor trace logs from each of three people.

BeaconPrint increases the accuracy of place learning and recognition to over 90%. When it does err, the percent of time BeaconPrint chooses the wrong place is also lower than previous approaches. The largest contribution of BeaconPrint, however, is its success in learning and recognizing places visited infrequently or only for short durations. People in our studies averaged 72.3 places they go at least twice a year (or once per year for medical-related places). Only 1 or 2 of these places are visited every day (usually home and work) and only 7 or 8 others are visited at least weekly. The other 63 places are visited infrequently. Although places visited most frequently are arguably the most personally significant, previous algorithms are generally quite poor at learning and recognizing anything except those most frequented places. Their accuracy with infrequent places averages 5-35%. BeaconPrint patches this deficiency by demonstrating an accuracy rate of over 63% even for places someone returns to only once or visits for less than 10 minutes, increasing to 80% accuracy for places visited twice.

References

1. Krumm, J., Cermak, G., Horvitz, E.: Rightspot: A novel sense of location for a smart personal object. In: Proceedings of the Fifth International Conference on Ubiquitous Computing (Ubicomp), Springer-Verlag (2003) 36–43
2. LaMarca, A., Chawathe, Y., Consolvo, S., Hightower, J., Smith, I., Scott, J., Sohn, T., Howard, J., Hughes, J., Potter, F., Tabert, J., Powledge, P., Borriello, G., Schilit, B.: Place lab: Device positioning using radio beacons in the wild. In: Proceedings of the Third International Conference on Pervasive Computing. Lecture Notes in Computer Science, Springer-Verlag (2005) to appear.
3. Marmasse, N., Schmandt, C.: Location-aware information delivery with commotion. In: Proceedings of the Second International Symposium on Handheld and Ubiquitous Computing (HUC). Volume 1927., Springer-Verlag (2000) 151–171
4. Ashbrook, D., Starner, T.: Using GPS to learn significant locations and predict movement across multiple users. Personal and Ubiquitous Computing **7** (2003) 275–286
5. Kang, J.H., Welbourne, W., Stewart, B., Borriello, G.: Extracting places from traces of locations. In: Proceedings of the Second ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots (WMASH 2004), Philadelphia, PA, ACM Press (2004) 110–118
6. Laasonen, K., Raento, M., Toivonen, H.: Adaptive on-device location recognition. In: Proceedings of the Second International Conference on Pervasive Computing. Volume 3001 of Lecture Notes in Computer Science., Springer-Verlag (2004) 287–304
7. Trevisani, E., Vitaletti, A.: Cell-id location technique, limits and benefits: An experimental study. In: Proceedings of the 6th IEEE Workshop on Mobile Computing Systems & Applications (WMCSA 2004), IEEE Computer Society Press (2004)
8. Clarkson, B.P., Pentland, A.: Unsupervised clustering of ambulatory audio and video. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). Volume 6., Springer-Verlag (1999) 3037–3040
9. Krumm, J., Hinckley, K.: The nearme wireless proximity server. In: Proceedings of the Sixth International Conference on Ubiquitous Computing (Ubicomp), Springer-Verlag (2004) 283–300
10. Patterson, D.J., Liao, L., Gajos, K., Collier, M., Livic, N., Olson, K., Wang, S., Fox, D., Kautz, H.: Opportunity knocks: a system to provide cognitive assistance with transportation services. In Davies, N., Mynatt, E., Siio, I., eds.: Proceedings of the Sixth International Conference on Ubiquitous Computing (Ubicomp 2004). Volume 3205 of Lecture Notes in Computer Science., Springer-Verlag (2004) 433–450
11. Griswold, W.G., Shanahan, P., Brown, S.W., Boyer, R.T.: Activecampus—experiments in community-oriented ubiquitous computing. Computer **37** (2004) 73–81