Computer Science and Artificial Intelligence Laboratory

Technical Report

MIT-CSAIL-TR-2008-075 December 30, 2008

Organic Indoor Location Discovery

Seth Teller, Jonathan Battat, Ben Charrow, Dorothy Curtis, Russell Ryan, Jonathan Ledlie, and Jamey Hicks

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

Organic Indoor Location Discovery

Seth Teller
MIT CSAIL
teller@mit.edu

Jonathan Battat
MIT CSAIL
battat@mit.edu

Ben Charrow
MIT CSAIL
bcharrow@mit.edu

Dorothy Curtis
MIT CSAIL
dcurtis@mit.edu

Russell Ryan
MIT CSAIL
rryan@mit.edu

Jonathan Ledlie
Nokia Research
jonathan.ledlie@nokia.com

Jamey Hicks
Nokia Research
jamey.hicks@nokia.com

ABSTRACT

We describe an indoor, room-level location discovery method based on spatial variations in "wifi signatures," *i.e.,* MAC addresses and signal strengths of existing wireless access points. The principal novelty of our system is its *organic* nature; it builds signal strength maps from the natural mobility and lightweight contributions of ordinary users, rather than dedicated effort by a team of site surveyors. Whenever a user's personal device observes an unrecognized signature, a GUI solicits the user's location. The resulting location-tagged signature or "bind" is then shared with other clients through a common database, enabling devices subsequently arriving there to discover location with no further user contribution.

Realizing a working system deployment required three novel elements: (1) a human-computer interface for indicating location over intervals of varying duration; (2) a client-server protocol for pre-fetching signature data for use in localization; and (3) a location-estimation algorithm incorporating highly variable signature data. We describe an experimental deployment of our method in a nine-story building with more than 1,400 distinct spaces served by more than 200 wireless access points. At the conclusion of the deployment, users could correctly localize to within 10 meters 92% of the time.

Categories and Subject Descriptors

C.2.4 [**Computer Communication Networks**]: Distributed Systems—*Distributed Applications*; H.5.3 [**Information Interfaces and Presentation**]: Groups and Organization Interfaces—*Collaborative Computing*

General Terms

Algorithms, Experiments, Human Factors, Measurement

Keywords

Localization, Crowd-Sourcing, Pervasive Computing, Geo-Tagging, Shared Sensing, Location-Based Services

MobiSys '09 June 22-25, 2008, Kraków, Poland

1. INTRODUCTION

Incorporation of information about a user's location can enhance a variety of applications, including calendars, reminders, navigation assistants, and communication tools. For example, the Locale application automatically adjusts mobile phone behavior based on the user's location [19].

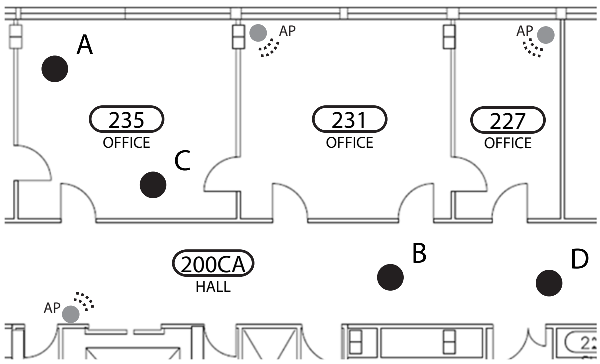


Figure 1: Organic Indoor Location Discovery. Any user (e.g., A, B) whose hand-held client observes an unrecognized wifi signature is asked by the client GUI for room-level location. The client sends time-stamped user-provided locations and concurrent wifi scans to a shared database. Clients of other users (e.g., C, D) later discover location by matching current wifi scans to stored, bound signatures. Users C and D need not contribute location data.

Most current location-aware applications depend upon GPS infrastructure [18], which provides location information only outdoors. Other methods have emerged to support indoor location discovery (*e.g.* [1, 14, 22, 26]). Like GPS, indoor location-determination methods require some kind of existing infrastructure (hardware, data, or both). These techniques occupy different regions of the space of design trade-offs spanning installation and maintenance burden, accuracy, precision, latency, user privacy, and client device requirements.

One class of indoor location discovery methods, based on radio frequency (RF) *signatures* – the source identities and signal strengths of ambient radio signals – has become widespread. Signature-based methods characterize the spatial variation in available radio signals (such as 802.11 broadcasts, cellular networks, bluetooth, etc.), compiling this information into a map. Each mobile device can then estimate its location by identifying the space(s) within the map whose signature(s) most closely match any signatures recently observed by the device. Researchers have reported room-level location accuracies of 95% within a building-sized testbed region [14]. Because signature-based methods can provide coarse accuracy without the need for additional dedicated infrastructure beyond what is needed for basic 802.11 connectivity, they have formed the basis for several commercial products [11, 24, 31].

Current signature-based methods have a high fixed cost, in that they require an initial "site survey" to build and populate the signal-strength map. This deployment burden, typically requiring a few person-days of careful and spatially comprehensive survey effort by skilled technicians, has prevented these methods from achieving significant penetration into indoor spaces. Survey-based methods face a cultural barrier as well, in that many members of a community may feel reluctant to allow unknown technicians into "private" areas such as offices. Finally, the site survey data itself may become outdated over time, *e.g.* through access point reconfiguration, repositioning or replacement, each of which may degrade or invalidate subsequent location estimates.

We adopt the following terminology. A *reading* is a single observation of an access point (AP) MAC address and its signal strength at some client. A *scan* is any set of readings produced by the client's wireless device driver. A *signature* is the union of one or more non-empty scans, i.e., a non-empty set of readings. Scans, readings and signatures are *unbound* if they are not yet associated with a location; otherwise they are *bound*.

The method described in this paper eliminates the initial, comprehensive site survey, replacing it with *organic* survey data collected on-the-fly by individual users. Client software running on each user's commodity personal digital assistant periodically gathers a signature of nearby wireless sources. This signature is checked against a locally-maintained signature cache, which is populated asynchronously from a shared server. If no match is found or if the match has low confidence, the device requests the user's current location. (In our system, the user indicates location by selecting a room outline on a labeled floorplan, but other indication methods are possible.) We call this user action a "bind" because it associates a signature (from the device's radio) with a location (from the user). The idea is that as more and more binds populate the system, the typical user will enjoy both a diminished interface burden, and higher-quality background location discovery — *i.e.,* eventually users will not need to do anything for their devices to determine where they are.

This approach brings to the fore a number of design and interface challenges. What is the appropriate persistent representation for spatially-varying signal strength data? What user interface should be used to collect organic signature data? What infrastructure is needed for effective and timely sharing of signature data in order to support client location discovery?

This paper makes the following contributions:

- An organic indoor location discovery framework;
- A graphical display and interface to capture location-tagged signatures with little user effort;
- An algorithm for room-level location estimation given organically-collected signature data of varying density; and
- An effective client-server protocol for pre-fetching signatures to be used in location discovery.

2. RELATED WORK

A central goal of location-discovery research and development is the realization of a hand-held device that can report its location in indoor environments. An effective method for room-level location estimation in indoor environments is the focus of the present paper.

Location determination is of fundamental interest to people, attracting human attention throughout recorded history. The most widely-used modern location determination system is GPS [18], which depends on a U.S.-government-deployed and -maintained ground tracking stations, satellites, and published updates of satellite ephemerides. Client GPS receivers incorporate a radio to receive GPS satellite transmissions and sufficient computation and storage resources to estimate georeferenced location. The reported location is not typically directly useful to humans, but is instead viewed in the context of a map, itself expressed in georeferenced coordinates. GPS receiver chipsets have become inexpensive and widely available, but function well only in outdoor regions with substantial sky visibility. Moreover, even if GPS service were somehow to be extended indoors, its utility would be limited due to the relative lack of georeferenced maps available for indoor regions.

In the absence of an absolute external coordinate reference like GPS, some location information can be maintained through dead-reckoning using odometry, pedometry, and/or inertial sensing. However, such methods require initialization from an external data source and calibrated, dedicated hardware, and can incur position errors of between one and ten percent of the total distance traveled [36]. Since location methods based on dead-reckoning incur error that grows without bound, they are unsuitable for use by applications that require location data over arbitrarily long time scales.

An alternative to dead-reckoning is the use of dedicated infrastructure, such as passive or active fiducial markers or beacons, along with matched client hardware, to support location discovery. Such systems can rely on active clients, whose motion is tracked and relayed by the infrastructure to other clients (*e.g.* [10, 33, 34, 35]), or active infrastructure, from which each client can compute its own location (*e.g.* [4, 15, 26]). Possible measurement infrastructures include ultrasound, modulated light fixtures, cameras, and RF (*e.g.* Bluetooth). Such approaches imply a deployment burden which, when contemplated solely to support indoor location services, may be deemed unacceptable or impractical in many settings.

Wireless and cellular network coverage has grown immensely over the past decade [5]. Accordingly, researchers have developed a number of methods that exploit infrastructure already in place for independent reasons, while performing some additional configuration or data-collection activity to enable indoor location determination.

2.1 RF Beacons and Signatures

Early work on localization that relied on existing RF infrastructure assumed that the location of RF transmitters was known and fixed. This research centered on determining a client's location *relative* to these *beacons*. Bahl *et al.'s* RADAR system estimated the distance from the client to each beacon using its received signal strength indication (RSSI), then trilaterated a location estimate [1]. Niculescu *et al.* used relative angles, rather than distances, from clients to beacons [25]. Others used time-difference-of-arrival techniques [12, 28] to generate precise distance estimates than RSSI alone could provide. Hightower *et al.* focused on removing the need for fixed infrastructure through rapid, flexible RF deployments [17]. A major difficulty in these approaches is that reflections and diffractions, multipath effects, and the presence of new objects, *e.g.* people, often stymied signal models and, in turn, distance and angle estimates.

Subsequent work by the RADAR group circumvented the problems of signal modeling and triangulation by shifting to RF signatures [2]. Figure 2 illustrates the RF signatures observed in four indoor spaces. Each of the four access points is received in each space with varying strengths. Due to walls, distance, and other factors, the observed signals in a particular space will be different from those in other spaces, even adjacent ones. Together, the RF signals observed in a space form that space's *signature*. Because many RF sources, *e.g.* wireless access points and cell towers, are geographically fixed, the set of signals observed in most spaces will be (fairly) consistent over time.

RF-signature-based location estimation works as follows. First, a database is constructed that contains signatures from all of the spaces in some region of interest. This database is a mapping from per-space signatures to spaces. Creating this database is typically called the "survey" phase. Given this mapping, a user device can gather a current RF signature, then find the closest match from this current RF signature to signatures in the database. The space with the closest match is returned as the result. This is called the "use" phase. Our organic data collection process merges these two phases, prompting for a user-generated "survey" only when localization confidence is low.

Haeberlen *et al.* demonstrated location estimation have found that RF signatures were 95% accurate to within $1-2$ meters in their indoor deployment [14]. LaMarca *et al.* have examined the compounding effect of using RF signatures from cellular towers in addition to 802.11 signals [22]. They found $20-30$ meter accuracy with nearly 100% coverage in a major metropolitan area. Castro *et al.* and Roos *et al.* examined methods for determining the closest match during the "use" phase [6, 27]. Krumm and Platt observed that, for many applications, human-scale spaces, *e.g.* rooms, formed a more natural location partition than grid points [21]. Outdoors, GPS can be used to build the signature-to-place mapping; this process, called *wardriving*, generates signatures that can later be used for location determination by devices that lack GPS but have wifi [31, 39]. While wardriving can be considered a form of organic data collection, its dependence upon GPS limits it to outdoor use.

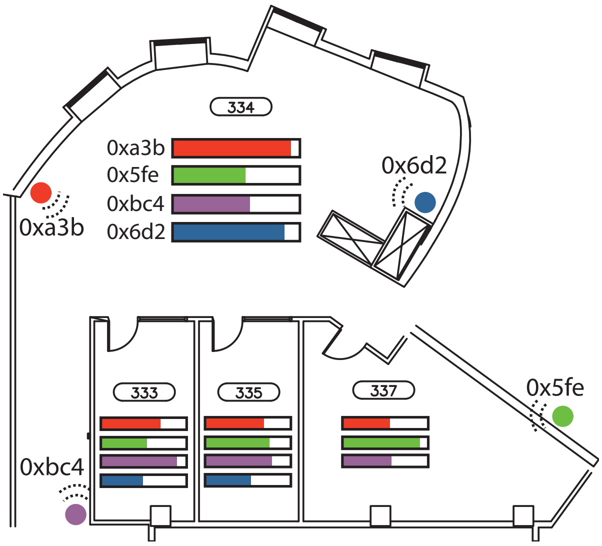


Figure 2: RF Signatures. The bars in each space illustrate the RSSI from each in-range AP. The observed per-MAC signal strengths constitute the *signature* for a particular space. Even though room 337 **is physically near access point** $0x6d2$ **(in blue), RF signals from the AP are dampened by structures in the environment. Capturing these effects with radio propagation models is a difficult problem that using RF signatures avoids.**

2.2 Sensor Networks

In contrast to the approaches described above, which focus on location discovery for human users, sensor network methods seek to localize nodes that are too numerous, too inaccessible, or too mobile for a human to directly contribute information about each node. In this domain, GPS is generally too costly or energy-intensive, or irrelevant, when relative location among sensors is more important than absolute location. RF-signature-based methods can be ill-suited for sensor networks due to node mobility over short time scales, requirements for self-configuration, or the need for location at finer than room grain.

Sensor network localization methods often focus on relative or hybrid positioning, where only a few beacons know their own absolute positions [29]. For usage scenarios such as disaster relief, researchers have designed sensor localization methods to be decentralized and robust to beacon failure [23] and to support rapid beacon redeployment [42]. Boukerche *et al.* provide a comprehensive overview of localization in sensor networks [3].

3. SYSTEM DESIDERATA

Our architectural and implementation efforts incorporated a number of preliminary observations, assumptions, and design principles arising from considerations of: the novel aspects of the problem we are addressing (§ 3.1); what infrastructure will typically be present (§ 3.2); expected user behavior (§ 3.3); and representational and algorithmic challenges (§ 3.4). This section reviews these desiderata.

3.1 Novel Aspects

There are two major characteristics of organic data collection that distinguish it from a survey approach. First, organic collection is integrated into the use phase of the service. There is no separation between "survey" mode and "use" or "location discovery" mode. Second, organic contribution should not overly burden users, *i.e.,* it should not cause users to alter their behavior significantly. As a result, a user who is organically collecting data would make only occasional contributions to the database, each time performing minimal effort. Organic data collection parallelizes database creation and maintenance — its speed scales with the number of contributors — but it does require active user contributors. Without them, data collection could potentially be much slower than a dedicated survey. Because maintenance is requested only when changes have actually occurred and because data is collected only for spaces that are actually used, organic data collection is, in these respects, more efficient than survey-based collection.

Studies performed by the PlaceLab group at Intel found that most individuals spend 80% to 90% of their time in one of a handful of places, such as their home or office [16]. These are often semi-private spaces to which a survey team might have difficulty gaining access. Assuming that users find location-enabled applications sufficiently compelling, users are likely to contribute at least some data for locations that they typically occupy within the first few days of using the service. The potential improvement in service accuracy is high in private spaces, which one user or a few users occupy frequently, but other users occupy rarely. Lastly, since most users are not specialists, we sought a contribution mechanism that could be performed easily by an untrained user, and could be seamlessly ignored if the user so chose.

From the considerations above, we reasoned that a light user burden, an intuitive interface, clearly-stated privacy policies, and perhaps public encouragement and acknowledgement of user contributions could be critical elements of a successful deployment.

3.2 Hardware and Data Infrastructure

Our method assumes that wireless access points supply connectivity throughout the region of interest, and are placed densely enough to produce signatures with sufficient variability to support room-grain discrimination. While it is difficult to quantify the required density precisely, it is certainly more than the minimum required for network connectivity, and less than one per distinct space. Good results for survey-based location estimation have been reported by others in regions with three floors, 510 spaces and 37 access points [14] (*i.e.* about 12 APs per floor, or one AP for every 14 spaces); our deployment setting is both larger and denser, with nine floors, $1,458$ spaces, and 224 access points (*i.e.* about 25 APs per floor, or one AP for every 7 spaces).

We further assume that 2-D floorplans of the region of interest are available (so that a GUI can render them to support user selection), and that each space has a unique text label (so that users can match space labels in the GUI to their surroundings). Many organizations have a Physical Plant unit that maintains suitable floorplans. If such floorplans are not available, their creation would be a prerequisite for the use of our method, imposing a substantial one-time burden ranging from a few hours (to author softcopy floorplans of a small building for which hardcopy floorplans exist) to much longer (to produce floorplans of a multi-floor, multi-building complex from scratch). One hopeful aspect of the latter (worst-case) scenario is that the authored floorplans need not be metrically precise, but rather only notionally accurate and correctly labeled, to support user recognition and selection of spaces.

3.3 User Behavior

A central challenge inherent in organic data is that it cannot be trusted to the same degree as data collected by a skilled, dedicated surveyor. In general, most users will make occasional mistakes; some will act maliciously; and a few may even collude to fool the system. We have attempted to minimize user error through an intuitive room-selection interface (§ 4.1), and to structure signatures so that data from errant users can be identified. Detection of malicious or collusive organic data is beyond the scope of this paper.

In addition to minimizing spurious user entries, we have attempted to minimize user burden through two means. First is a standard "snooze" mode, through which users can silence prompts for any length of time (*e.g.* for a meeting). Second, we have provided a way for users to specify their locations for extended periods of time. These *interval binds* (§ 4.1.1) gather more data than "I am here right now" snapshot mechanisms. Our bind interface can capture hours of scans from a single user action. It is particularly useful for spaces such as offices, conference rooms and lounges, which users typically occupy for long periods.

3.4 Algorithmic Challenges

Algorithms designed to localize using organic data must address several challenges that do not exist with surveyed data. Although we prompt the user when there is insufficient data to form a confident location estimate, our localizer must handle spotty data: spaces will have a widely-varying number of scans, or no scans at all. Unlike survey-based schemes, in which survey data periodically refreshes the entire database, in an organic set-

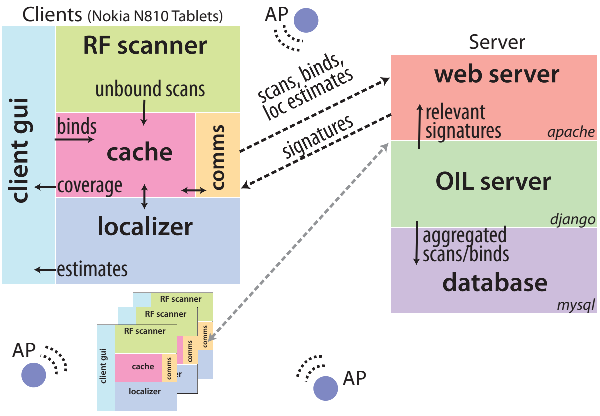


Figure 3: OIL System Architecture. OIL client software on a tablet PDA collects signal strengths from in-range APs, time-stamping and periodically transmitting them to the OIL server. When prompted, the user indicates, via the client GUI, the tablet's current location. This *bind* action associates the location with scans collected by the client around the same time. Each bind is sent to the server, where it is aggregated with other binds for the same location (typically from other users) to form a wifi signature for that location. To speed client-side localization, the server responds to signature requests by returning to the client only those signatures with high approximate relevance.

ting individual users regularly contribute, and depend upon, freshly-acquired data. Thus we must provide a mechanism for efficient sharing of relevant novel contributions.

4. ARCHITECTURE

Organic Indoor Location (OIL) clients and servers cooperate to enable each client to quickly and locally determine its location. These twin goals of quick localization and client-side localization, together with the goal of minimizing user effort, guided our design efforts.

The goal of minimizing user effort led to an interface with which users can supply past, present and (expected) future location (§ 4.1), and a client-server API that assembled raw scan data with user binds on the server (§ 4.2) to allow sharing with other clients (§ 4.3). The goal of client-side localization resulted in our client cache and additions to the API to populate it (§ 4.4). For lower-latency client-side localization, we developed a server-side algorithm to filter outgoing cache entries by likely relevance (§ 4.5). Lastly, we constructed a client-side localization algorithm that does not require the whole corpus (§ 4.6). Figure 3 provides an overview of how these components interact. Before we discuss these elements in depth, we briefly summarize the additional infrastructure ingredients required to realize our OIL discovery method:

Existing 802.11 Deployment. Our method requires that the region of interest contain an existing deployment of 802.11 wireless access points configured to broadcast "presence" information. The deployment should be comprehensive; *i.e.,* a wireless device anywhere within the region should be able to detect, and measure the signal strength of, several access points [2, 21]. The organic nature of our system imposes the further restriction that each client device must be able to establish occasional connectivity with the OIL server in order to exchange signature and bind data.

Wireless-Enabled Devices. Our system, like others based on 802.11 signal strength, makes use of the ability of the wireless device driver to perform a "scan" for available access points. Each scan returns a list of access point MAC addresses and, for each access point, a numerical signal strength. The range of this value depends on the driver, but can be calibrated to a common scale [14].

Contributing Users. Our system also requires a group of users willing to contribute location information either on their own initiative or when prompted by their devices.

4.1 Location Specification User Interface

User-enabled signatures and binds are essential to realizing a working organic location discovery system. Signatures for each space are built up from *binds.*

As each user responds to the device's input prompts, the user's client creates one bind from each response. A bind associates a set of RF observations with a user-supplied location, an optional user-supplied time interval, a timestamp (according to the client's clock), and the client device's MAC address. The system records the device's MAC address in order to distinguish signatures collected by different users or devices. Our user interface gathers and sends a stream of binds to the server, using a *prompting* mechanism that solicits the user's location, and a *reporting* mechanism that queues captured binds for transmission to the server.

We support both time-based and confidence-based user prompting, and enable the user to configure the interface to choose which type of prompting to perform, or to suppress prompting altogether. Time-based prompting generates a user alert at fixed, user-configurable intervals, *e.g.* every five minutes. Confidence-based prompting solicits the user's location only when the device observes an unfamiliar signature, *i.e.* whenever the location-discovery method produces a low-confidence location estimate (§ 4.6). Users may also glance at the device display to note the device's current location estimate, and perform a bind if the device is mistaken. We anticipate that general organic deployments will primarily use confidence-based alerts.

In order to perform a bind a user must provide, at a minimum, a space name. In principle, an organic location discovery system could be based on user-supplied plain-text (or spoken) space names, such as room numbers or conference room names. We chose not to ask

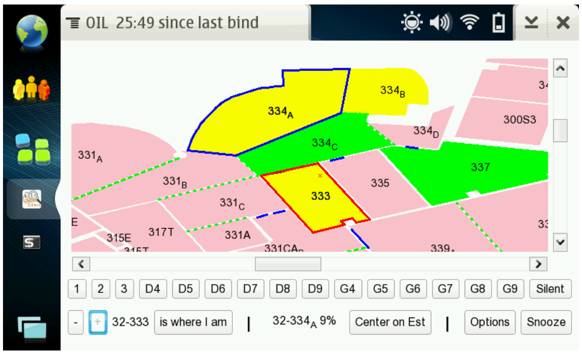


Figure 4: User Interface. A users can select, from a floorplan, the currently-occupied space by clicking on that space (the buttons underneath, *e.g.* D4, enable the user to change floorplans). The selected space, here room 32-333, is then outlined in red. Users bind RF scans to the space with the "is where I am" button. Most of the spaces shown here have minimal organic data (signified by their pink coloring). As users contribute more data, room colors shift to yellow, then green. Also shown are doorways (blue) and subdivisions of large open areas (dashed lines).

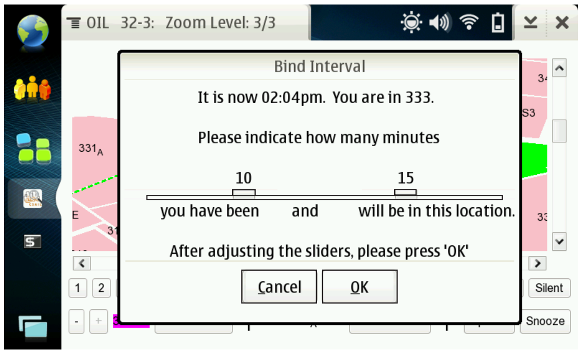


Figure 5: Interval binds. Here the user specifies a time interval extending from 10 minutes in the past to 15 minutes in the future.

users to specify place names as plain-text, for two reasons. First, collecting unstructured text from a large community of users would produce a confounding variety of names for any given space, based on variations in case, hyphenation, spelling, and even common usage. For example, one conference room in our building has three "names:" an official room number; a formal room name derived from its donor; and a colloquial room name derived from the room's shape. While such variety could be useful in future systems, for example to enable automatic discovery of synonyms among multiple names for the same space, we chose simplicity over naming variety for our prototype system. The second reason we eschewed plain-text space name specification is that many users can not conveniently enter text on the keypad of the hand-held personal digital assistants used in our deployment.

To support structured space indications, we collected a set of floorplans for our nine-story building including both educational and research spaces. After extracting polygonal space contours and enclosed text labels from AutoCAD files [37], we designed a user interface enabling users to indicate location by clicking on a graphical map (Figure 4).

4.1.1 Interval Binds

Developers of previous site survey methods have chosen to represent user location in two distinct ways, as discrete snapshots or smoothly interpolated waypoints.

Snapshot-based location methods (*e.g.* [14]) treat user location as a series of brief discrete locations. In this framework, a dedicated site surveyor visits every space in turn, standing in each space for a prescribed interval (typically two minutes). The survey application then associates any scan data collected during that interval with the user-specified space. The disadvantage of such methods, as mentioned earlier, is that a dedicated site surveyor is required; given a new surveying task every two minutes, that person can accomplish little else.

Smooth-location methods (*e.g.* [11]) treat user location as a linear interpolation of a series of instantaneous locations specified "on-the-fly" by a slowly walking user. Since scans and binds are time-stamped, analysis software can later associate interpolated readings and user locations. As well as requiring a skilled surveyor, such methods have additional disadvantages. First, the surveyor must move with roughly constant speed so that linear interpolation of positions is valid. Thus, the surveyor must take special measures to inform the interface when motion is paused or restarted. Second, the surveyor must take care to indicate locations with sufficient spatiotemporal density to ensure that interpolated positions will be sensible.

Our interface has an additional novel aspect in that it, in addition to soliciting the user's present location, it solicits the user's (recalled) past duration and (expected) future duration in that space (Figure 5). At the cost of a slight increase in interface complexity, this "interval bind" mechanism accrues several advantages. First, in many instances we can collect far more signature data than the few seconds or minutes collected by interpolative or snapshot-based methods; users often spend an hour or more in a given location, enabling our application, scanning at about $\frac{1}{2}$ Hertz, to collect tens of thousands of readings. Second, we can suppress user prompts for the future interval specified by the user, further lowering the user burden. Note that when no interval information is provided, the interface effectively takes a snapshot of the user's location, one that may or may not implicate any scans.

The disadvantage of interval binds is, of course, that the supplied information may be inconsistent or incorrect. This occurs, for example, if the user's specification of current location (or of an interval at some past location) differs from that user's previous specification of future location. Our prototype system assumes honest, competent users, *i.e.,* users who intend to report their locations accurately and do so by correctly operating the user interface. Any user who changes location so as to partially or fully invalidate an active future interval bind can later supply updated location information by creating a new interval bind extending into the past. We leave as future work interface improvements to better capture moving users, such as those walking along corridors.

4.2 Signature Server

The cooperative nature of our system necessitated a centralized repository for each collection of tagged location data. In future, less centralized designs, different organizations could run their own localization servers. Our data store is implemented as a network-accessible server with which the OIL client exchanges data in order to share contributions and facilitate location discovery. The server's main roles are to: *store* scans and interval binds reported by OIL clients; *provide* OIL clients with representative signatures for nearby locations; *record* localizer estimates generated by the client for evaluation of localizer performance by comparison to "ground-truth" interval binds; and *generate* useful summaries and visualizations of server activity.

We required a scalable, flexible architecture for our prototype, since we aspire to organic data collection with contributions from thousands of users. We use a standard three-tiered server architecture, with a web server, an application server, and a database. We selected components that will facilitate deployment growth over time: an Apache webserver, as the front end for OIL client requests; an application framework for processing application logic; and a MySQL database for storage. While our prototype has only single instances of each component, using these standard components should allow us to scale our system to scores of buildings at a minimum. We use Django as the application framework [8]; this enabled rapid prototyping and simple sharing of objects between clients and the server, both of which were implemented in Python.

4.3 Server API

We developed a client-server API that assembles raw scan data with user binds. In our current prototype, this process of building signatures occurs on the server, but could occur locally for disconnected operation. To populate the server signature database, the client sends three types of data, which are packed into messages and queued according to priority:

- An **unbound scan** is a set of access point MAC addresses and time-stamped signal strengths as observed by the device. Unbound scans are collected at about $\frac{1}{2}$ Hertz, and are buffered into groups of sixteen for transmission to the server.

- An **interval bind** is a user-selected location name, along with a time-stamped "now" and past and future time intervals. Queued interval binds are transmitted to the server every minute or so.

- A **location estimate** is a time-stamped estimate from the localization algorithm – *i.e.*, a space name, or `None` – along with a list of AP MAC addresses sufficient to reconstruct, at the server, the input data that the algorithm used to estimate its location. Server knowledge of location estimates is not necessary for client location discovery. Rather, we implemented this data stream to enable debugging and to produce the experimental results presented later in this paper. Our prototype system performs location discovery about once every five seconds, and transmits queued location estimates roughly once per minute.

Figure 6: Only spaces (yellow, green) with signatures similar to those recently observed by the client are pre-fetched for use by the localizer.

4.4 Signature Caching and Pre-Fetching

Another system goal was for clients to be able to estimate location without necessarily contacting the server. That is, the localization algorithm should be able to run locally on each client, and should return correct results provided that appropriate data is locally available. This goal led to a client-side pre-fetching cache, and additions to the API that provide hints to the server about what signatures to transmit to the client.

Each device maintains a cache of stored signatures, organized by distinct space. Each signature has an associated expiration time, after which it is slated for refresh from the server. The contents of the signature cache are managed by the device, in cooperation with the signature server, as follows (Figure 6). The device provides the server a set of recently observed MAC addresses, along with a list of any cached spaces not pending expiration. The server returns a set of locations with signatures compatible with the query signature (§ 4.5).

This management policy implies a brief lag as the initially empty cache is populated with signatures involving the very first MAC addresses discovered by the device, but tends to eliminate lag for subsequent user motions from space to space (since signatures of nearby spaces are typically fetched before those spaces are entered). To support caching and pre-fetching, we added the following message to the client-server API:

- A **request locations** message is a query from an OIL client for locations (including bound signatures) that may be relevant to the localizer. The client provides a list of recently-observed MAC addresses, with a signal strength range for each, and a list of its already-cached locations.

4.5 Signature Compatibility

The pre-fetching algorithm described above requires a rough compatibility metric for signatures, implemented at the server as follows:

1. The server first selects locations whose signatures include per-AP signal strengths within some delta of the strengths in the query signature. Our prototype uses a delta of ± 2, which preliminary experiments showed to be an effective value.

2. For each resulting signature, the server calculates the compatibility $c(l, q) \in [0, 1]$ of each candidate location l to the client-provided signature q. Let A and B represent the set of MAC addresses in the signatures of l and q respectively. Then

$$c(l, q) \equiv \frac{1}{2}\left(\frac{|A \cap B|}{|A|} + \frac{|A \cap B|}{|A \cup B|}\right)$$

3. The server then returns to the client each location l such that $c(l, q) \geq \frac{1}{3}$, an effective experimentally-determined threshold.

To limit network traffic and client memory and CPU usage, the server sends no more than 2000 readings total in any signature, of which no more than 500 may come from a single bind. Our prototype implementation of this policy simply selects readings in order of recency (newest first). Only these server-selected readings, delivered to the client cache, are considered by the localization algorithm. A more sophisticated heuristic for server-side reading selection might involve aggregating different scans based on the current time of day and day of week, or other temporal predictors of activity.

4.6 Indoor Location Discovery Algorithm

Each client device runs a location discovery algorithm that uses cached locations, together with recently observed signatures, to estimate which space (if any) is currently occupied by the device. If no estimate can be produced with high confidence, because the device is in a location with insufficiently many associated signatures, the location discovery algorithm reports `None`.

Each client tablet running the OIL application periodically invokes a location estimator, which collects the MAC addresses and strengths recently observed by the device (through channel scanning), and identifies the cached space whose corresponding signature is most similar. We sought a computationally simple procedure that would not overtax the tablet's CPU resources, adopting a simple matching and voting scheme.

Each cached location L has an associated signature containing a set of MAC addresses M, and for each MAC address $m \in M$ a set of signal strengths $s(m)$ observed within L. At the start of localization, the cache contains n locations $L_1 \ldots L_n$. The localizer has a current query signature Q, consisting of the 4 most recent scans the tablet has made, and for each MAC address $q \in Q$ a set of signal strengths $s(q)$.

The localizer proceeds by distributing one "vote" per access point among the cached spaces, in proportion to the similarity of each space's signature to Q. The first step is to compute the similarity between each cached location and each access point individually. For a single MAC address, we define the similarity $f(q, c)$ of two integer signal strengths q and c as

$$f(q, c) \equiv b \text{ if } (|q - c| \leq m) \quad (1)$$

$$\equiv b^{||q-c|-m|} \text{ if } (|q - c| > m) \quad (2)$$

where b and m are experimentally chosen parameters (we use $b = 0.8$ and $m = 3$). Next, we define the similarity $F(\mathcal{Q}, \mathcal{C})$ of *sets* $\mathcal{Q}$ and $\mathcal{C}$ of signal strengths:

$$\frac{1}{|\,Q\,| + |\,C\,|} \cdot \left[\sum_{q \in \mathcal{Q}} \max_{c \in \mathcal{C}} f(q, c) + \sum_{c \in \mathcal{C}} \max_{q \in \mathcal{Q}} f(q, c)\right]$$

This function incorporates both presence and absence of matching signal strengths.

For each AP within the query signature, the localizer computes F over all cached signatures, aggregating each AP's per-space vote in an $M \times N$ matrix of non-negative votes representing the affinity of each AP in the query signature to each cached location. The column with the maximum sum – *i.e.* the space with the highest vote total – is then returned by the localizer. This procedure runs in $O(MN)$ time for $O(1)$-size signatures. More sophisticated localization methods using multi-dimensional search data structures (*e.g.* [13]) are possible.

5. TEST DEPLOYMENT

We launched a test deployment of the organic indoor location discovery system by identifying about twenty Stata Center occupants as candidate "test users." We organized a few half-hour briefing sessions during which we distributed tablets and gave short demonstrations of the OIL application. In addition, we gave each user a one-page sheet summarizing the OIL GUI operations, *e.g.,* switch floors, pan, zoom, space select, and interval bind. We emphasized that the deployment region was limited to the Stata Center, but that users would be free to take the tablet wherever they wished. We made an offline record of which user got each tablet to enable us to identify users with broken hardware or software. We asked users to carry their tablets for a few weeks, and urged them to "make a bind whenever you have been in a single place for a few minutes, or intend to stay in a single place for a few minutes." At the end of the first briefing, we zeroed the database. Some users chose to begin participating well after the first briefing session; some users never contributed any binds. Table 1 summarizes the organic dataset contents on the twentieth day of deployment. We examined the deployment characteristics along two dimensions: (a) how coverage and accuracy changed over time (§ 5.1), and (b) how various user behaviors affected the system's organic growth (§ 5.2).

Map Spaces	1,458
Contributing Users	16
User Binds (from users)	640
Scans (from devices)	882,118
Bound Scans	117,231 (13.3%)
Readings (from devices)	17,169,461
Bound Readings	1,987,721 (11.6%)
Spaces with Bound Readings	169 (11.6%)

Table 1: Summary statistics for test deployment.

5.1 System Utility

We studied the logged location estimates in order to characterize the utility of the system according to a number of metrics.

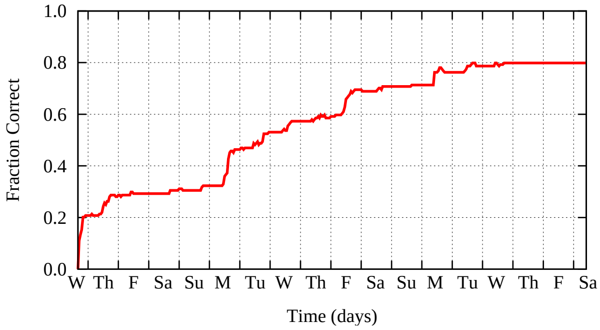
The **Coverage** metric characterizes the fraction of map spaces to which readings had been bound by the user community (Figure 7). One day into the deployment, our user group had covered 107, or about 7.3%, of the 1,458 spaces in the corpus. Six days later, that figure had grown to 166 (about 11.4%). By the end of the deployment, almost all covered spaces had sufficiently many readings to support accurate localization. More broadly, in a building with approximately a thousand daily occupants, some sixteen users – less than two percent – were able to cover more than ten percent of the building by expending less than a minute of effort each per day, on average.

The **Global Accuracy** metric characterizes the quality of the location estimate computed by the client, aggregated over all spaces (Figure 8(a)), *i.e.,* the number of correct location estimates in a given sample period divided by the total number of spaces as of the end of the prototype deployment. At the start of the deployment, the global accuracy was zero, as there were no locations known to the system. The global accuracy increased thereafter, reaching a maximum of 80% on the final day of deployment.

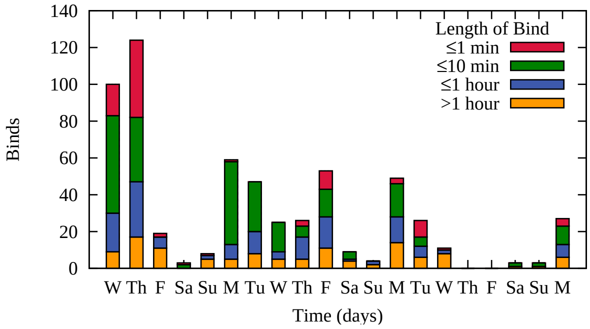
Figure 8(b) shows the distribution of bind lengths for each day of the deployment. Figure 8(c) shows the distribution of bind-minutes per space, again per day. (Note that some spaces with binds accumulated no readings, due either to wireless device driver errors, or to users' later superseding binds.) We highlight two interesting aspects of these data. First, one enthusiastic user generated some 25 short (single-minute or shorter) binds in distinct spaces on the first Thursday. These spaces were never bound again by other contributors. This type of behavior could produce poor location accuracy in our system. Second, on the first Monday of the deployment, global accuracy jumped significantly (Figure 8(a)), due to many binds longer than one minute entering the database (Figure 8(b)).

The **Local Accuracy** metric characterizes the quality of the location estimates computed by the client on a per-space basis. We generated a random subset of 25 of the 169 bound spaces as of the end of deployment. In each selected space, we placed a tablet and made an interval bind extending 15 minutes after the time of placement. (We modified the server to suppress any binding of new readings, in order to prevent fresh scans from being incorporated into any signatures.) We then analyzed each resulting 15-minute bind, and its approximately 100 location estimates, to determine location estimation accuracy in that space (Figure 9).

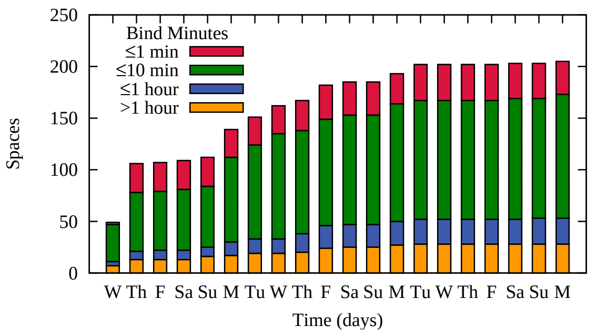
In some spaces (8 out of 25) the localizer chose the correct space in greater than 90% of its estimates. The localizer performed poorly in eleven spaces, failing to identify the correct space more than 50% of the time. Most of its failures occurred in large, open areas that had been manually sub-partitioned. Accuracy in corridor spaces varied widely, ranging from 98% to 3%.



(a) Global Accuracy

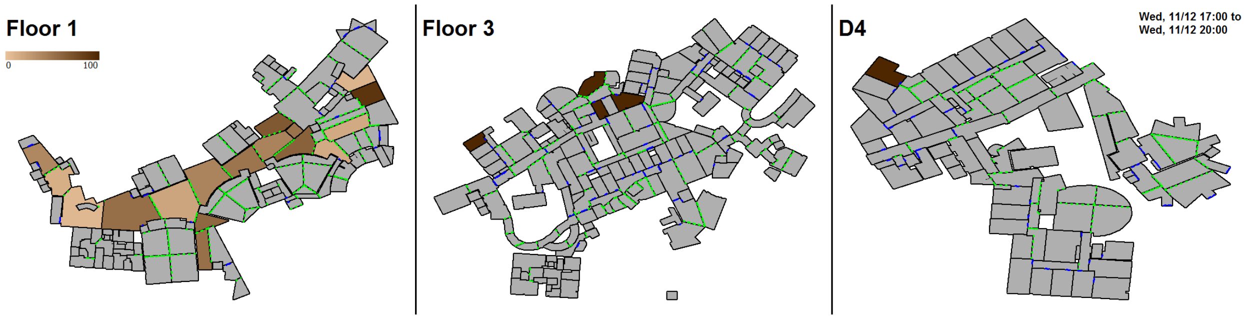


(b) Durations of Interval Binds



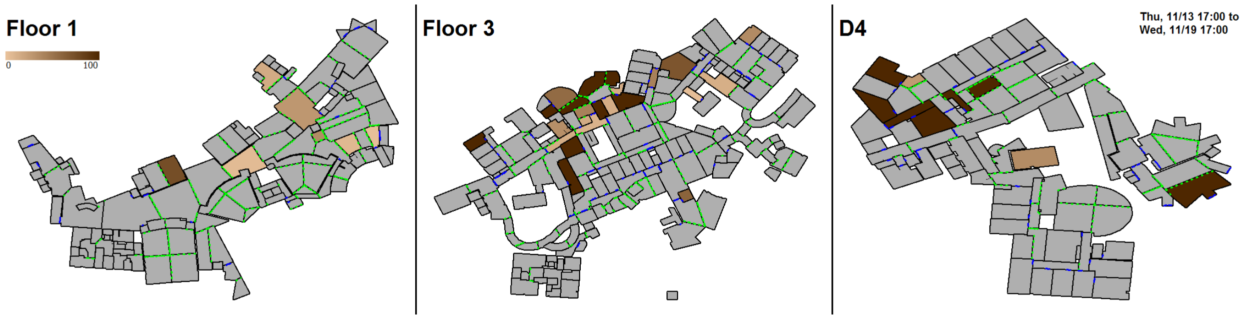
(c) Cumulative Per-Space Bind-Minutes

Figure 8: Organic growth over the deployment: overall localization accuracy grew directly with user contributions.



(a) First 3 Hours

(b) Rest of Day One



(c) Days 2 - 7 Inclusive

(d) Days 1 - 20 Inclusive

Figure 7: Organic contributions to Floors 1, 3 and D4 during the first three hours (a); hours 4-24 of the first day (b); days 2-7 (c); and for the entire 20-day deployment (d). Color indicates number of bound scans per space; uncolored spaces accumulated no bound scans during the displayed interval.

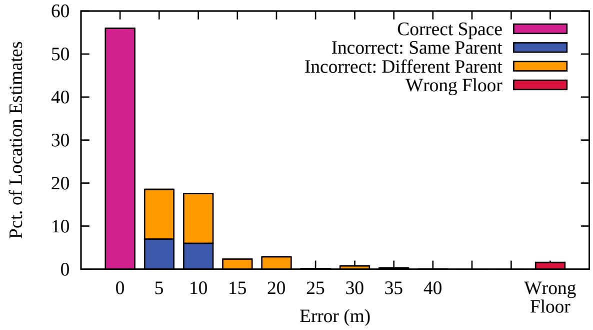


Figure 9: Localization Error. Spot checks of localizer accuracy showed that for 92% of all tests, the localizer was correct within 10m. Subspaces that are the result of manual partitions are considered to have the same "parent." The localizer got the floor wrong less than 2% of the time.

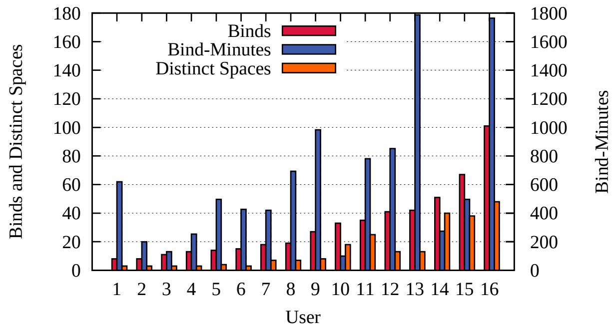


Figure 10: Signatures per user. Like Wikipedia and other user-contributed resources, a handful of users were significantly more active contributors than the rest. Note that some users with relatively few binds (*e.g.,* 9 and 13) contributed a large fraction of the total bind-minutes. Data shown are from the first eleven days of deployment.

5.2 User Characteristics

We studied the resulting logged binds in order to characterize user behavior. One notable example of an organically-grown resource is the Wikipedia on-line knowledge repository [41]. Previous studies of user-contributed repositories have described a 1/9/90 classification of users by contribution level [38]. In our setting, we expected a few users to perform at least one large-scale survey or to contribute data nearly everywhere they go. Some users might perform a few small-scale surveys, for instance walking the corridors on one floor of their building or providing updates after a change to the local network. The remaining majority of users might not contribute any data at all; these "free riders" would enjoy the service based on the efforts of the more active minority.

While the size of our user study is too small to state conclusively that this breakdown in user behavior persists with system scale, we did in fact observe something like this classification in the number of signatures contributed per user (Figure 10). Other slices of the data, such as the distinct spaces covered per user, showed a similar division.

This suggests that the majority of data used by an organic location service will not be from a uniform cross-section of users. Instead, the data will more likely consist of multiple, overlapping, small-scale amateur "surveys." That is, because this 1% will become relatively experienced at contributing data, their contributions will, in general, be of high quality — more like an expert surveyor. A preliminary conclusion is that the majority of the data in the database used by an organic location service will likely resemble multiple, overlapping, small-scale surveys. The more "organically" collected data would be the quick touch-up contributions by the 9% and 1% groups that help patch these surveys together and keep them up to date. The main advantage of our service is that it can integrate both kinds of contributions without interfering with the user experience.

In theory, one could distinguish survey data from organic data by looking at the distribution of its density. Survey data will be evenly spread over each space due to the systematic effort of surveyors. Organic data should clump according to several factors, including the density of users in an area, the savviness of those users, and perhaps the volatility of the network signature, due to prompting users for contributions when localizer performance is degraded.

Another useful figure of merit in assessing the system is the average user burden over the deployment period. We have no direct way to measure the amount of time a user's attention is typically absorbed by the tablet GUI, but we can conservatively upper-bound by thirty seconds the time required for the user to select a space, invoke the interval bind GUI, optionally adjust its sliders, and press "OK" (this neglects the half hour that most users spent in our briefing session). Since there were 16 distinct users contributing over 20 days, and those users made a total of 640 binds, the fraction of each user's time absorbed by GUI activity was 0.07%, or about one minute per user per day on average, with the most active user devoting an average of almost three minutes per day to author 101 binds.

5.2.1 *Interval Bind Interface*

We examined how users used the interval bind mechanism by dividing the range of future and past selections for bind intervals into $12 \times 12 = 144$ five-minute buckets. Figure 11 illustrates the percentage of binds for each combination of past and future interval. Almost half of all binds had zero past and future intervals, the default. Since these most common binds are also the least valuable (since they implicate the fewest readings), we plan to study how to maximize the information that can be gleaned from such binds.

The data also show that users more often indicated that they were *going to be* in a space, rather than that they *had been* in a space. For example, about 13% of all future-only binds indicated that the device would remain in the same space for the next half-hour or longer,

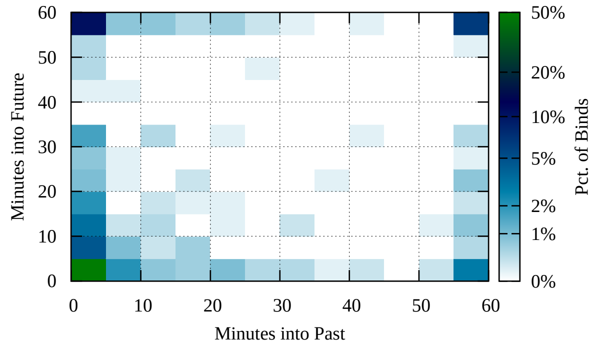


Figure 11: Distribution of user-selected future and past interval pairs. The upper right-hand corner shows, for example, the fraction of binds for which users asserted their devices had been in the same space for 55-60 minutes, and would remain in the same space for 55-60 minutes.

while only about 4% of past-only binds indicated that the device had been in the same space for the past half-hour or longer.

The fact that more than half of all binds did extend into either the future or the past suggests that our test users attempted to actively contribute more than current position data. It also suggests that effective methods for validating these extended user contributions need to be developed.

6. FUTURE WORK

This paper demonstrated organic location discovery in a specific setting and user community. There are a number of additional issues to be considered if the method is to be applied in a broad variety of settings. In particular, issues that require addressing in future work include: the interval-bind interface (§ 6.1), organic maps (§ 6.2), estimator performance (§ 6.3), and privacy (§ 6.4).

6.1 Interval Bind Interface Alternatives

The user interface is a central aspect of any system depending on user input, as in our "organic" framework. We adopted an interval bind mechanism in which users indicated location in the present moment, and could optionally supply past and future times at which they entered, or at which they intended to leave, a given location.

Some users suggested alternative interfaces. For example, each user could supply a location upon entering and leaving each space, and the system could bind, to that space, any signatures acquired during the resulting interval. The potential problem, of course, would be with users who neglect to tell the tablet when they are leaving a given space, causing the system to mistakenly associate signatures from elsewhere with the initially-specified location.

Some users commented that they were not sure how much time they would remain in a given location, so were unable to use the future interval bind mechanism effectively. An alternative would be to restrict the system to accepting only retroactive binds, *i.e.* providing the ability to specify past, but not future, location. In fact, this is possible in our current interface (by leaving the *future* slider unused), and some users exercised the interface this way. Although our prototype discards superseded *future* binds when users rebind to different locations, some users indicated they would prefer to be able to directly cancel the current bind, or even all binds back to some specified time.

Our interface lacks an effective method for capturing pass-through spaces such as hallways. Adding a one-button, instantaneous bind mechanism (*e.g.* by double-tapping on a mapped space) would ameliorate this shortcoming, as would modeling user movement, either through interpolation, wifi estimation, or use of on-device data from accelerometers, imaging or acoustic sensors.

6.2 Organic Maps

Our prototype requires polyline-based floorplans as input. In order to grow beyond managed spaces such as malls, offices, and schools, users themselves must be empowered to author their own maps. While we have shown how to share and reduce the deployment burden previously required for indoor localization, a much more comprehensive space maps will be needed to achieve practical, widespread indoor location services. To make organic maps, users could potentially draw from public resources such as city databases (*e.g.* [7]), or they could draw simple pictures using existing tools (*e.g.* [30, 40]). As noted in Section 3.2, floorplans need not be metrically precise to be useful in our framework. In addition, users could link indoor spaces to outdoor ones by "grounding" exterior doorways with geo-referenced coordinates. To more easily incorporate floorplans in the future, we will seek heuristics to automatically divide map spaces into sub-spaces that make sense to humans — in contrast to the manual subdivisions we made for our prototype deployment.

Interestingly, many applications for indoor localization and location-based services, in general, do not require either a building-relative or absolute notion of location. For example, user-defined names of their current location, *e.g.* home, work, would allow applications such as Locale [19] to adjust behavior based on context.

6.3 Location Estimator Performance

An important issue for any location discovery system is the performance of the location estimation method – primarily its accuracy, latency and CPU requirements. While these were not central issues in the current effort, which was focused on developing a proof-of-concept prototype, they will certainly emerge as we deploy the system to larger, multi-building environments and many more users.

One intriguing possibility for improving localization within sparse organically-captured floorplans is to exploit user mobility between bound spaces. We also plan to further investigate the localizer's vote allocation method, and the availability of topological floorplans to infer an intermediate location for the user when the localizer assigns high confidence to two or more spaces. Lastly, we may use previous user behavior — stored on the client — as an additional hint for current location.

Another hurdle for location estimation is accommodation of disparate devices and wireless device drivers. Our framework raises the possibility of an "organic" calibration procedure that uses naturally-overlapping binds produced by many devices in public spaces to infer the relationship between devices' signal strength measurements.

As signatures change due to repositioning or replacement of RF sources, match confidence will drop. This will cause the client to observe unfamiliar signatures, automatically triggering prompts of the user to enter updated binds. Over time, user bind activity should return the system, for a given spatial region, to a state in which no user input is required. Finally, we note that modern access points that actively regulate broadcast power clearly will pose a challenge to location algorithms based on assumptions about the spatiotemporal persistence of RF signatures.

6.4 Privacy

Gains in personal indoor location discovery capability can seem to come only at the expense of individual users' privacy. Because organic data must be shared to generate a useful system, our proposal complicates the already tricky domain of privacy of personal location and movement. Indeed, even clarifying the meaning of privacy in a localization context seems challenging [32].

In our prototype in particular, the client's transmission of its MAC address to the server with each bind, and its transmission of recent scan signatures in order to prefetch cache entries, may reveal some some information about the user's present location. We capture the client's MAC for the purposes of correctly organizing the database, discounting data from consistently mistaken users, tracking software versions, and, potentially, for discarding malicious organic input. Eliminating MAC transmission might make each of these issues more challenging. Similarly, eliminating signature transmission from the client might force the server to maintain per-client state in order to maintain current levels of performance. While MAC information does not leave the server, and user movements are de-identified through the aggregation of binds into signatures, that the server stores implicit information about user movement at all is problematic. We plan on investigating methods to keep signatures validated while preserving at least some aspects of individual and group privacy. As in other domains, such as the Internet, anonymizing users often comes at the expense of accountability [9]. Note that single users could profitably use our system with reasonable accuracy and complete privacy if they did not contribute signatures and exclusively used their own locally-generated data.

Krause *et al.* propose principles of *community* sensing, finding a balance between the benefit of the mass character of user-generated data — GPS data from cars in traffic, in their case — and the potential privacy problems from storing and using that data [20].

7. CONCLUSION

This paper described a new method for building indoor location discovery systems: using user-generated, or *organic*, data as the basis for RF signature formation. Starting with an empty base map, we exploit users' natural mobility while prompting each user to inform the system about current locations on a map displayed on a personal device. Over time, these organic contributions lead to localizing with sufficient confidence that no user input is required. To better handle organic data, we developed three novel elements: (1) a human-computer interface for indicating location over intervals of varying duration; (2) a client-server protocol for pre-fetching signature data for use in localization; and (3) a location-estimation algorithm incorporating highly variable signature data. We contrasted our work against previous approaches and current commercial work, which require expertly-performed site surveys. After twenty days of usage in an experimental deployment, our organic indoor location discovery service could correctly localize to within 10 meters in 92% of estimates.

Acknowledgements

We thank: Nokia Research Center Cambridge for their financial support of, and technical engagement with, this effort; the early adopters in the Stata Center who volunteered to test the system; David Lambeth for his assistance in determining effective values for the localizer parameters; and Ye Wang for her assistance in generating Figure 7.

8. REFERENCES

[1] P. Bahl and V. N. Padmanabhan. RADAR: An In-Building RF-Based User Location and Tracking System. In *INFOCOM*, Tel Aviv, Israel, Mar. 2000.

[2] P. Bahl, V. N. Padmanabhan, and A. Balachandran. Enhancements to the RADAR User Location and Tracking System. Technical Report MSR-TR-00-12, Microsoft Research, February 2000.

[3] A. Boukerche, editor. *Algorithms and Protocols for Wireless Sensor Networks*. Wiley-IEEE Press, 2008.

[4] B. Brumitt, B. Meyers, J. Krumm, A. Kern, and S. A. Shafer. EasyLiving: Technologies for Intelligent Environments. In *Handheld and Ubiquitous Computing*, Bristol, UK, 2000.

[5] V. Bychkovsky, B. Hull, A. Miu, H. Balakrishnan, and S. Madden. A Measurement Study of Vehicular Internet Access Using In-Situ 802.11 Networks. In *MOBICOM*, Los Angeles, CA, Sept. 2006.

[6] P. Castro, P. Chiu, T. Kremenek, and R. R. Muntz. A Probabilistic Room Location Service for Wireless Networked Environments. In *Ubiquitous Computing*, Atlanta, Georgia, September 2001.

[7] City of Cambridge: Property database. cambridgema.gov.

[8] Django project. djangoproject.com.

[9] J. R. Douceur. The Sybil Attack. In *International Workshop on Peer-to-Peer Systems (IPTPS)*, Cambridge, MA, USA, March 2002.

[10] N. Eagle and A. Pentland. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing*, 10(4), 2006.

[11] Ekahau positioning engine. ekahau.com.

[12] R. J. Fontana, E. Richley, and J. Barney. Commercialization of an Ultra Wideband Precision Asset Location System. In *Ultra Wideband Systems and Technologies*, Germantown, MD, Nov. 2003.

[13] A. Guttman. R-trees: a dynamic index structure for spatial searching. *SIGMOD Rec.*, 14(2):47–57, 1984.

[14] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, and L. E. Kavraki. Practical Robust Localization over Large-Scale 802.11 Wireless Networks. In *MOBICOM*, Philadelphia, PA, Sept. 2004.

[15] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster. The Anatomy of a Context-Aware Application. In *MOBICOM*, Seattle, Washington, 1999.

[16] J. Hightower, S. Consolvo, A. LaMarca, I. E. Smith, and J. Hughes. Learning and Recognizing the Places We Go. In *Ubiquitous Computing (Ubicomp)*, Tokyo, Japan, September 2005.

[17] J. Hightower, R. Want, and G. Borriello. SpotON: An Indoor 3D Location Sensing Technology Based on RF Signal Strength. Technical Report UW CSE 2000-02-02, University of Washington, Seattle, WA, February 2000.

[18] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. *GPS Theory and Practice*. Springer, 1997.

[19] C. Jernigan, C. Bayley, J. Lin, and C. Wright. Locale. http://people.csail.mit.edu/hal/mobile-apps-spring-08/, May 2008.

[20] A. Krause, E. Horvitz, A. Kansal, and F. Zhao. Toward community sensing. In *IPSN*, St. Louis, Missouri, Apr. 2008.

[21] J. Krumm and J. Platt. Minimizing Calibration Effort for an Indoor 802.11 Device Location Measurement System. Technical Report MSR-TR-03-82, Microsoft Research, Nov. 2003.

[22] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. E. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. N. Schilit. Place Lab: Device Positioning Using Radio Beacons in the Wild. In *Pervasive Computing*, Munich, Germany, May 2005.

[23] K. Lorincz and M. Welsh. MoteTrack: A Robust, Decentralized Approach to RF-Based Location Tracking. In *Location- and Context-Awareness (LoCA)*, Oberpfaffenhofen, Germany, May 2005.

[24] Navizon: Peer-to-peer wireless positioning. navizon.com.

[25] D. Niculescu and B. R. Badrinath. Ad Hoc Positioning System (APS) Using AOA. In *INFOCOM*, San Franciso, CA, Mar. 2003.

[26] N. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket Location-Support System. In *MOBICOM*, Boston, MA, Aug. 2000.

[27] T. Roos, P. Myllymäki, H. Tirri, P. Misikangas, and J. Sievänen. A Probabilistic Approach to WLAN User Location Estimation. *Interational Journal of Wireless Information Networks (IJWIN)*, 9(3), 2002.

[28] A. Savvides, C.-C. Han, and M. B. Srivastava. Dynamic fine-grained localization in Ad-Hoc networks of sensors. In *MOBICOM*, Rome, Italy, July 2001.

[29] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz. Localization from mere connectivity. In *MobiHoc*, Annapolis, Maryland, 2003.

[30] Sketchup. sketchup.google.com.

[31] Skyhook wireless. skyhookwireless.com.

[32] D. Solove. *Understanding Privacy*. Harvard University Press, 2008.

[33] Sonitor indoor positioning. sonitor.com.

[34] Talking lights. talking-lights.com.

[35] R. Want, V. Falcao, and J. Gibbons. The Active Badge Location System. *ACM Transactions on Information Systems*, 10:91–102, 1992.

[36] L. Whitcomb, D. Yoerger, H. Singh, and J. Howland. Advances in Underwater Robot Vehicles for Deep Ocean Exploration: Navigation, Control and Survey Operations. In *Ninth International Symposium on Robotics Research (ISRR)*, London, UK, 1999.

[37] E. Whiting, J. Battat, and S. Teller. Generating a Topological Model of Multi-Building Environments from Floorplans. In *Proc. of Computer-Aided Architectural Design (CAAD) Futures*, pages 115–128, July 2007.

[38] S. Whittaker, L. G. Terveen, W. C. Hill, and L. Cherny. The Dynamics of Mass Interaction. In *Conference on Computer-Supported Cooperative Work (CSCW)*, Seattle, WA, November 1998.

[39] Wigle: Wireless geographic logging engine. wigle.net.

[40] Wikimapia. wikimapia.org.

[41] Wikipedia. wikipedia.org.

[42] Y. Zhang, K. Partridge, and J. Reich. Localizing Tags Using Mobile Infrastructure. In *Location- and Context-Awareness (LoCA)*, Oberpfaffenhofen, Germany, September 2007.