Find My Stuff: Supporting Physical Objects Search with Relative Positioning

Jens Nickels, Pascal Knierim, Bastian Könings, Florian Schaub,
Björn Wiedersheim, Steffen Musiol, and Michael Weber
Institute of Media Informatics, Ulm University
Ulm, Germany
firstname.lastname@uni-ulm.de

ABSTRACT
Searching for misplaced keys, phones, or wallets is a common nuisance. Find My Stuff (FiMS) provides search support for physical objects inside furniture, on room level, and in multiple locations, e.g., home and office. Stuff tags make objects searchable while all other localization components are integrated into furniture. FiMS requires minimal configuration and automatically adapts to the user's furniture arrangement. Object search is supported with relative position cues, such as "phone is inside top drawer" or "the wallet is between couch and table," which do not require exact object localization. Functional evaluation of our prototype shows the approach's practicality with sufficient accuracy in realistic environments and low energy consumption. We also conducted two user studies, which showed that objects can be retrieved significantly faster with FiMS than manual search and that our relative position cues provide better support than map-based cues. Combined with audiovisual feedback, FiMS also outperforms spotlight-based cues.

Author Keywords
Localization; physical artifacts; relative positioning; RFID; RSSI; search; ubiquitous computing; ZigBee.

ACM Classification Keywords
H.4.m Information systems applications: Miscellaneous; H.5.2 Information interfaces and presentation: User interfaces.

General Terms
Design, Experimentation, Human Factors.

INTRODUCTION
An average person misplaces up to nine items per week, most frequently mobile phones, keys, and sunglasses, and spends about 15 minutes per day searching those objects [3]. Yet, in contrast to searching information online, locating physical objects is rarely supported by technology. Simplistic key finders exist which emit acoustic or visual signals to aid search. More advanced variants [18] use Bluetooth tags to locate objects with smartphone apps. However, search support is limited to the user's proximity in such approaches. Mobile phones can be localized without physical proximity, but require GPS and cellular network connectivity, which may not be available indoors. As a result, indoor localization and search support have received considerable attention in the research community.

In general, a support system for physical-object search must balance search range, feasibility, and ease of use, resulting in a number of requirements. Localization of a sought-for object should not require the user's physical proximity in order to also support search in remote places (*user-independent localization*). Localization of objects should be *robust against occlusion and inclusion* to provide search cues that also aid retrieval of covered objects, or objects placed inside furniture. In general, *intuitive search cues* are required that effectively guide users in retrieving sought-for objects efficiently, and preferably faster than with manual searching. A search system should be easy to setup, function without manual calibration and making objects searchable with the system should be easy and unobtrusive (*seamless configuration*). To ensure practicality, components should have low energy consumption and equipment costs, especially those attached to, or integrated into searchable objects (*energy and cost efficiency*). Finally, objects should only be locatable and searchable by their respective owners to prevent abuse (*privacy and security*). At the end of the paper, we discuss how our system and related work match up to these requirements.

In this paper, we propose the indoor search system *Find My Stuff (FiMS)*. Our main contribution is the concept of supporting object retrieval with relative position cues instead of highly accurate localization. In our system, search components are integrated into furniture and objects are located in relation to such smart furniture. The combination of multiple equipped furniture pieces facilitates advanced relative position cues, such as "the wallet is between couch and dresser, near couch." We envision that users purchase furniture already equipped with FiMS components, which can then be setup in the home with minimal configuration. Our system is robust against furniture rearrangement and further supports object localization in multiple environments in a robust system.

UbiComp'13, September 8–12, 2013, Zurich, Switzerland.

ACM 978-1-4503-1770-2/13/09...$15.00.
http://dx.doi.org/10.1145/2493432.2493447

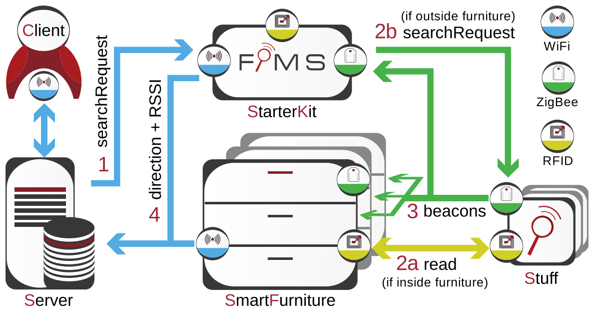


Figure 1: FiMS architecture: When the client sends a search request, the server starts the search process. SmartFurniture components locate Stuff modules with RFID and ZigBee.

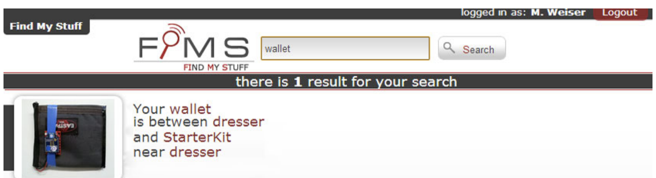


Figure 2: Search interface providing a relative position cue.

We first describe the concepts and architecture of FiMS, before presenting our prototype system along with a functional evaluation, showing the viability of our approach. We further conducted two user experiments in which we analyzed the effectiveness of FiMS' search cues in comparison to manual search, as well as to other cues proposed in related work, which will be discussed afterwards. We conclude with a discussion of advantages, limitations, and potential extensions of our approach.

PHYSICAL-OBJECT SEARCH WITH FIMS

Our system consists of multiple components (see Fig. 1). A user's objects are managed by a *server*, which provides a Web-based search interface and coordinates the search process. Figure 2 shows the FiMS search interface with the result of a search query for an object (*"wallet"*). The object's position is indicated by a relative position cue.

Physical objects are made searchable with FiMS by equipping them with a *Stuff* tag, which may be attached to or integrated with the object. Figure 3 shows our Stuff prototype. Stuff tags consist of a ZigBee module, a passive RFID transponder, marginal processing capabilities, and a battery. Stuff tags must be small and have low energy consumption.

FiMS locates objects with specifically equipped furniture pieces, called *SmartFurniture*. Our SmartFurniture supports localization inside itself and in its proximity. Inside furniture, passive RFID is used to locate objects on a compartmental level corresponding to the furniture's layout, e.g., inside different drawers to enable drawer-specific localization (see Fig. 4). Outside furniture, FiMs provides relative position cues in relation to furniture. For this purpose, all SmartFurniture pieces in a FiMS environment form a ZigBee mesh network and each SmartFurniture measures a Stuff's ZigBee signal strength with up to four directed antennas (front, left, right, behind). Each FiMS environment has a privileged SmartFurniture, called *StarterKit*, that additionally coordinates the local mesh network and supports registration of new objects. A FiMS environment can range from a single room with one StarterKit to a large apartment with lots of SmartFurniture. Server and SmartFurniture communicate via WiFi, which has the advantage that existing communication infrastructure can be leveraged. While the server could also reside locally, Internet access is required to enable users to search for objects when not at home or in other authorized environments, e.g., the office or a friend's house. Next, we explain the search process in more detail, before outlining registration of new furniture and searchable objects.

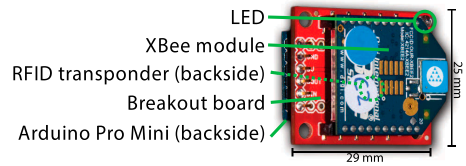


Figure 3: Stuff prototype based on Arduino.

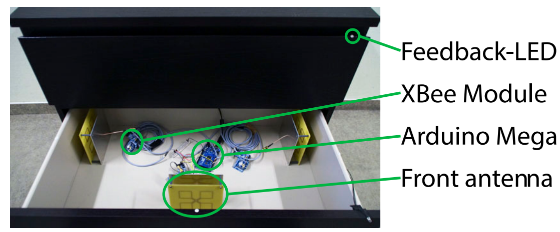


Figure 4: SmartFurniture prototype with three directed ZigBee antennas and Arduino controllers in the middle drawer.

Hierarchical Search

Energy and privacy implications prohibit continuous tracking of objects. Instead, FiMS employs a hierarchical search model to process a search query for a specific object. The server sequentially triggers different search layers until the sought-for object has been located, at which point the search process is stopped and the user is presented with a retrieval cue provided by this layer. Search layers are ordered to favor localization methods that provide quick results with no or little energy consumption. Localization methods consuming more time and energy are only used if previous layers returned no result. As layers are independent from each other, the hierarchical model can be easily extended by adding additional localization methods as new layers.

Our current search model supports four layers. First, the server tries to contact the object's Stuff at a *cached position*. If a cache entry exists, the server tries to locate the Stuff with the same localization method. If unsuccessful, the search is extended to the *insides of SmartFurniture* near the last known position, using RFID. If the object cannot be located in this manner, SmartFurniture is used to locate the object *in the environment* with ZigBee. If also unsuccessful, the search is extended to other environments associated with the user, such as other rooms in the apartment, the user's office, or authorized environments of friends. In *external environments*, the search process follows the same steps as described above. Search order of external environments could be prioritized based on heuristic models of the user's previous searches to reduce energy consumption. Here, we focus on search in one environment and leave extensive consideration of external environments for future work.

As mentioned above, localization within SmartFurniture is based on RFID. Each SmartFurniture is equipped with a number of low-range RFID readers, corresponding to a furniture's shape and function, e.g., to distinguish between multiple drawers or locate objects placed on top of surfaces in the case of tables or shelves. The server sends an RFID search request with the Stuff's identifier to a specific SmartFurniture, if a cached position exists, or sequentially to all SmartFurniture in an environment. As Stuff modules have a passive RFID transponder, answering the request consumes no energy of the Stuff and is fast. When using RFID, SmartFurniture detect all Stuff inside them, and their position is cached by the server. Therefore, the server likely already has a cached position of an object if it is inside any SmartFurniture. The search is terminated as soon as the sought-for Stuff is located with RFID by a SmartFurniture.

On the next layer, ZigBee is used to locate the object in the environment. In contrast to RFID search, the server addresses the sought-for Stuff directly. The request is sent out by all SmartFurniture in the specific environment. If the sought-for Stuff receives the request, it sends a beacon sequence, which is received by one or more SmartFurniture in proximity. All other Stuff in the environment remain in sleep mode to conserve energy. The relative positioning algorithm for ZigBee localization is detailed below.

Relative Positioning with ZigBee RSSI

Localization with ZigBee encompasses SmartFurniture measuring the received signal strength from sought-for Stuff and the server estimating the Stuff's relative position, as shown in Figure 1 (steps 1, 2b, 3 & 4). First, the StarterKit receives a search request for the sought-for Stuff from the server via WiFi. The request is disseminated in the local ZigBee mesh network by the StarterKit, other SmartFurniture, and other Stuff modules, to the sought-for Stuff module. If the sought-for Stuff receives the request, it broadcasts a sequence of beacons. Each SmartFurniture that receives one or more beacons, measures the RSSI of each received message with all its directed ZigBee antennas, e.g., three antennas in the case of the dresser shown in Figure 4. RSSI measurements are multiplied with an antenna-specific attenuation factor, which is preconfigured by the furniture manufacturer to account for furniture characteristics, such as wall thickness and used materials. To compensate for variations caused by external influences [21], the median RSSI of all received beacons per antenna and the number of received beacons are used to weight results in subsequent calculations. In our tests, 6 beacons proved to be a reasonable compromise between robust results and search speed. From these 6 beacons per directed antenna, the median RSSI value is determined. Next, the median RSSI values of all antennas of one SmartFurniture are sorted. The highest median RSSI determines the direction estimate which is reported to the server by a SmartFurniture. Each antenna's RSSI measurements are also included to enable consistency checking by the server.

The server receives direction estimates from one or more SmartFurniture. In order to determine the Stuff's position in relation to the SmartFurniture pieces, the server maintains a connected, bidirectional *furniture graph* $G = (E, V)$. Each $v_i \in V$ represents a SmartFurniture, which is connected with up to four neighbors—the closest SmartFurniture pieces to the *left*, *right*, in *front*, and *behind*. Neighbor relations between two SmartFurniture modules are always bidirectional. They are determined in the pairing process, described below, when joining a local environment. Note that a SmartFurniture does not know its actual position in the environment, but the orientation (left, right, front, back) of each of its antennas, which can be preconfigured for furniture pieces with a predominant orientation. For example, the front-facing antenna in a dresser is always the *front* antenna, regardless of how the dresser is positioned in a room. Thus, each SmartFurniture can report direction estimates for a Stuff's position based on its antenna directions without having to know the exact position and orientation of itself.

The server recursively calculates the Stuff's relative position with its furniture graph, and the received RSSI values and direction estimates. By considering measurements of multiple SmartFurniture, and each SmartFurniture contributing multiple median RSSI values (one per directional antenna), the positioning algorithm is robust against erroneous RSSI values and temporary signal distortion. Currently, the resulting relative position cue is either a single direction (*right of*, *left of*, *front of*, *behind*), if only one SmartFurniture received the Stuff's beacons, or a *between X and Y near X/Y* relation for two SmartFurniture pieces. Theoretically, position cues could also relate to more than two furniture pieces. However, reporting concise and helpful cues for multiple furniture pieces requires further research. Note that while position cues only relate to the two closest SmartFurniture, direction estimates from all SmartFurniture that received the Stuff's beacons are considered in the localization process, which increases position accuracy. The resulting relative position cue is returned to the user together with a stored image of the sought-for object, as shown in Figure 2.

Balancing Configuration and Security

Two main configuration tasks arise in our system: registering new Stuff and integrating or rearranging SmartFurniture. The challenge is to minimize manual configuration effort while providing sufficient security to prevent tracking of users or unauthenticated localization of their objects. FiMS leverages spatial proximity of RFID tags and readers to reduce manual configuration to a minimum. The StarterKit in a FiMS environment not only serves as ZigBee coordinator (other SmartFurniture are ZigBee routers), but also has a *registration plate*, consisting of an additional ZigBee module and RFID reader, for the registration of new objects. User authentication is realized with personalized RFID tags; an admin tag is capable of generating user tags and pairing new SmartFurniture.

Stuff registration

Figure 5 shows the process of registering a new object with FiMS. The ZigBee network (N_1) of Stuff and SmartFurniture is encrypted (AES-128) to prevent eavesdropping. Thus, the Stuff of the new object must first obtain the network key to join N_1. For this purpose, the user first authenticates at the StarterKit with a personal RFID tag. If the user has the required privileges, the StarterKit switches into registration mode, indicated by a blinking LED. The StarterKit opens a separate unencrypted ZigBee network N_0, which an unregistered Stuff joins automatically. StarterKit and Stuff establish a shared key, which is used to encrypt the subsequent exchange of configuration information. To prevent eavesdropping on key establishment, the range of N_0 can be physically restricted, e.g., by reducing transmission power.

Once a shared key has been established, the StarterKit obtains the Stuff's ZigBee address, RFID ID, and receives the Stuff's default name and description. If the Stuff is directly embedded in an object, the description can provide the object's semantics. A camera mounted above the registration plate also takes a photo of the new object. All obtained information is stored by the server. The StarterKit provides the Stuff with the *Personal Area Network (PAN)* ID and encryption key for network N_1 and triggers a restart of the Stuff module, which then leaves N_0 and joins N_1. The registration process is transparent for the user, who only places a Stuff-tagged object onto the StarterKit's registration plate and waits for visual feedback (green LED) that the Stuff has been registered successfully. Afterwards, the user can adjust the Stuff's details via the server's Web interface.

SmartFurniture pairing

The process of adding new SmartFurniture is initiated by swiping the admin RFID tag of the local environment over a labeled position on the new SmartFurniture. From the admin tag ID, the new SmartFurniture derives the SSID and encryption key of the local WiFi network (e.g., WPA2-PSK). After the new SmartFurniture joins the WiFi network, it is detected by the FiMS server which requests additional information preconfigured by the furniture manufacturer, including the furniture type and model, as well as information about integrated RFID readers, e.g., their association with a specific drawer. The server sends the PAN ID and encryption key of ZigBee network N_1. When the SmartFurniture has joined N_1, the server initiates the pairing process to determine the new SmartFurniture's position in relation to other present SmartFurniture. The new SmartFurniture sends broadcasts consecutively through all its directed antennas. Similar to Stuff localization, existing SmartFurniture measure RSSI with their directed antennas. The server obtains all RSSI measurements and updates the furniture graph by determining the new furniture's relative position in relation to existing SmartFurniture, or creates a furniture graph if no other SmartFurniture is in range. For diagonal alignments of SmartFurniture (as in Fig. 8b), pairing may result in two correct relations, since two orthogonal antennas might measure nearly the same RSSI

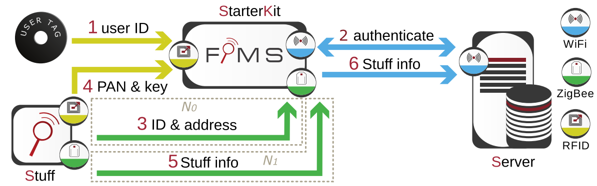


Figure 5: Registration of a new object with Stuff module.

value. The same process is also used to recalibrate the furniture graph upon environmental changes, e.g., when SmartFurniture has been moved. Recalibration can be triggered periodically by the server, manually, or by accelerometers in the SmartFurniture.

This pairing and calibration approach has the advantage that meaningful SmartFurniture relations can be derived and utilized for object localization without requiring a ground truth, such as a room map or explicitly specified furniture positions. Pairing is only based on the orientation of directed antennas of involved SmartFurniture pieces to each other, without knowing a SmartFurniture's actual position. The purely relational information maintained in the furniture graph is sufficient to provide meaningful search cues, because users are able to map the relative search cue onto the physical furniture arrangement.

PROTOTYPE AND FUNCTIONAL EVALUATION

We built a fully functional prototype of FiMS based on Arduino. Our current setup [7] consists of two SmartFurniture (a StarterKit and a dresser) and multiple Stuff modules. We used XBee Series 2 modules for ZigBee; for RFID, we used 125kHz readers and World Tag passive RFID transponders.

Our Stuff modules (see Fig. 3) consist of an Arduino Pro Mini, a passive RFID tag, and an XBee module with integrated antenna on a breakout board. The Stuff prototype is further equipped with a bright LED and a buzzer to optionally guide users with audiovisual feedback. Our SmartFurniture prototype is a dresser (see Fig. 4). Each of the three drawers is equipped with an RFID reader and a blue LED to support retrieval of objects inside drawers. Opening the drawer to take out the sought-for Stuff switches the LED back off. Except for these LEDs, the smart dresser looks like regular furniture. In the middle drawer, we added three planar four-quad antennas with reflectors [1] for ZigBee localization, pointing left, right and to the front. No back antenna was added as drawers are commonly placed against walls. The radiation pattern of these antennas (see Fig. 6), measured while mounted in the drawer, shows that they actually receive signals from different directions. Higher signal strength changes exactly at 45° between front and left antenna. Between front and right antenna, this point is slightly shifted to -40°, due to manufacturing variations. Due to the drawer's metal bearing slides, the gain of the right and left antenna are lower compared to the front antenna, which is compensated in Stuff localization with a negative attenuation offset for the front antenna. Two Arduino Mega boards handle localization and control the XBee modules, RFID readers, and LEDs of

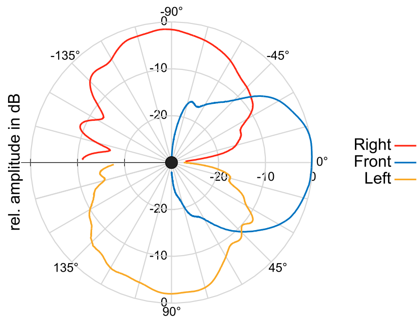


Figure 6: Radiation pattern of the dresser's three build-in antennas, facing front (0°), right (-90°) and left (90°).

Figure 7: Furnished meeting room used for positioning accuracy experiments and first user study.

the SmartFurniture. A WiFi interface handles communication with the server. While the StarterKit can be integrated into normal furniture, our current model is a basic rectangular stand equipped like the dresser. One XBee module acts as ZigBee coordinator. In addition, the StarterKit has a registration plate on top, containing the additional XBee module and RFID reader for Stuff registration.

Position Accuracy Experiments

We performed multiple experiments to evaluate positioning accuracy and reliability of our ZigBee-based relative positioning approach. To provide realistic interference, experiments were performed in a meeting room (55 m^2) furnished with tables, chairs, book shelves, couches, plants, and an interactive display (see Fig. 7). For our measurements, we partitioned the room into 50×50 cm cells (13×16 cells). Due to the book shelves and a concrete pillar on the right side, 8 cells had to be excluded. We also excluded the cells of the SmartFurniture pieces, as they are not relevant for ZigBee localization. For each of the remaining 198 cells, we performed 3 independent search queries without caching, resulting in 3,564 RSSI values and 594 result cues per experiment. The result of each query is calculated by 6 different RSSI measurements as part of our search algorithm. In the middle of each cell, a Stuff module was placed at table height (≈80 cm). In multiple preliminary experiments we found almost no differences between table height and positioning on the floor. The ZigBee nodes operated on channel 15 (2.425 GHz), which was automatically selected by the coordinator node based on energy scans of available channels. Thus, nearby WiFi access points on channels 1 (2.412 GHz), 6 (2.437 GHz), and 11 (2.462 GHz) did not cause notable interference. Further mitigation strategies have also been proposed to improve ZigBee operation in busy channels [13].

Positioning accuracy in opposing arrangement

In the first experiment two SmartFurniture pieces were facing each other (see Fig. 8a). The preceding pairing process correctly determined the furniture graph, indicating that the SmartFurniture pieces were in front of each other. Thus, we compared obtained search cues against the ground truth derived from the furniture graph (left part of Fig. 8a), which provides the expected relative position cue for each cell. The solid lines separate the three main areas *left*, *right* and *in front* of dresser (D) and StarterKit (SK) by 45° each. In the middle, *between D and SK*, we distinguish two subareas which indicate whether an object is *near D* or *near SK*.

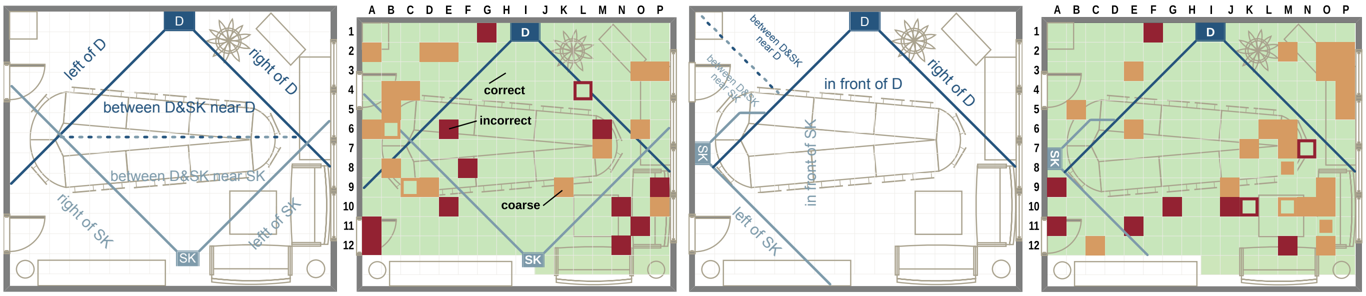
The right part of Figure 8a shows the result accuracy of the 3 search queries per cell. A fully colored cell indicates consistent results over all 3 queries (98%). A differently colored cell border indicates that 1 result deviated and how, e.g., measurements for cell L4 returned 2 correct and 1 wrong result (*left of D*). Green cells indicate a correct position cue. Resulting cues in border regions (e.g., on the line between left and front areas) were rated correct when they matched one of both areas in the ground truth. Red cells indicate incorrect results, e.g, in cell G1 the resulting cue was *between dresser and StarterKit, near dresser* although Stuff was placed left of D. Orange cells indicate coarse results, which deviate from the ground truth, but could still support retrieval. For instance, the result of cell M7 was *right of dresser*, which could still help users when frontally facing the dresser. 87% of all cues were correct (green). Coarse results (orange) were given in 47 cases, incorrect results (red) in 34 cases of 594 total. The incorrect results around the middle table are likely caused by the surrounding chairs, as some cell centers were very close to a chair's backrest, which could have blocked the signal. Interferences caused by the room arrangement or windows are also possible reasons, as some incorrect results are consistent in both experiments.

Positioning accuracy in diagonal arrangement

We further considered a diagonal arrangement of StarterKit and dresser (see Fig. 8b). The relation in the furniture graph was determined as *StarterKit left of dresser* and *dresser right of StarterKit*, due to the diagonal alignment. Thus, the ground truth (left part of Fig. 8b) covers a *between* area. Results were consistent over all 3 queries for 97% of cells. In 85% of the searches, FiMS returned a correct position cue (green). Of the 594 resulting cues, there were only 23 incorrect (red) and 61 coarse results (orange). The results in Figure 8b show three clusters of coarse results (≈1–1.5 m^2). Shading by the TV, and refraction and reflection from the window front and table end opposite of the StarterKit are likely reasons.

Positioning accuracy with obstructing person

The previous experiments show that unequipped furniture (e.g., chairs, tables, and shelves) had only a marginal effect on the accuracy of relative position cues in our setting. To assess the influence of persons in the room on position accuracy, we performed another experiment using the same opposing arrangement and ground truth as shown in Figure 8a. Rather



(a) Experiment with opposing arrangement

(b) Experiment with diagonal arrangement

Figure 8: Ground truth (left) and positioning accuracy results (right) for two arrangements of two SmartFurniture pieces: dresser (D) and StarterKit (SK). A Stuff was placed in the center of each cell and 3 independent search queries (3×6 RSSI measurements) were performed per cell. Green shows correct, red shows wrong, and orange shows coarse results. A fully colored cell indicates the same result for all 3 queries. A colored cell border indicates that 1 query result deviated and how.

than measuring all cells again, we selected 3 Stuff positions at cell boarders directly in front of the dresser (I3/I4, I5/I6, I7/I8). For each Stuff position, a person (male, 1.80 m, 68 kg) was placed at 5 positions, resulting in 15 combinations. The person stood directly in front of D (at I1/I2 and I3), sat on a chair between D and SK (at I5 and I9), and stood directly in front of SK (at I12). At each Stuff position we performed 3 queries, resulting in 45 position cues of which 98% were consistent per cell and 91% correct. Of the 45 cues, there were 4 coarse and no wrong results. Coarse results occurred only if the person stood directly in front of the StarterKit and Stuff was placed further away, e.g., for I7/I8 the cue was *between D and SK, near D* instead of *[...] near SK*. Further analysis of RSSI values revealed that the obstructing person caused the ground truth area to shift towards him by ca 0.5 m, but did not cause wrong results.

Stuff Energy Consumption and Size

To be practical, Stuff modules must be able to operate on batteries for long times and be sufficiently small. Our current Stuff prototype consists of off-the-shelf Arduino components to facilitate flexible experimentation with different hardware. However, these ready-made components increase form factor and energy consumption, which could be reduced by the development of a dedicated board with integrated components. While the current Stuff prototype is smaller than a box of matches (see Fig. 3) and could already be used as a key fob, an integrated Stuff board could be the size of a large coin, fitting unobtrusively into a wallet.

Our hierarchical search model ensures that energy requirements of Stuff are quite low, because the passive RFID tag requires no energy source, and the Arduino and XBee components can be in sleep mode, only waking up when being queried. The XBee module polls for new data every 5 s and wakes the Arduino board if data is available. We measured Stuff's energy consumption in 1h with a precision source measurement unit. Stuff consumed 3.82 mWh; 1.48 mW while sleeping, 78 mW while polling. Reacting to a search request consumes 0.445 mWh. Thus, 1 day standby including one search request would require 92.12 mWh. Using two coin cells (2×1000 mAh 3 V), our Stuff prototype can operate for at least 2 months, under the assumption that it is being searched once a day on average.

USER EVALUATION

The results of the position accuracy experiments shows that our approach and prototype are robust to use in realistically furnished settings and that the effects of occluding persons are negligible. We further performed two independent user studies to assess the effectiveness of the provided search support for the retrieval of objects.

User Study 1: Comparison with Manual Search

We conducted a user study to compare the effect of search support with FiMS on a user's search time compared to manually searching for objects. We were particularly interested in determining if the measured position accuracy (85-87%) was sufficient to reliably assist users in object retrieval.

Setup and Procedure

In order to be able to relate position accuracy to search performance, we conducted the experiment in the same setting as the accuracy measurements, the meeting room shown in Figure 7, with StarterKit and dresser facing each other (see Fig. 8a for ground truth and position accuracy). We performed a between-subjects experiment, in which each participant was randomly assigned to search *with FiMS* or *manually* and had to perform 5 search tasks, in which a Stuff-tagged wallet was hidden at different locations.

We chose realistic locations for a lost wallet that covered areas with varying search cue accuracy (see Fig. 8a). The wallet was hidden near a *trash bin* (A12), behind the *plant* (L2), under the *couch* (O12), under the *interactive display* (M2), on top of a *chair* (I5). The wallet was placed on the floor, except for the *chair*. It was never directly visible, but always peeked out to give manual searchers a realistic chance.

At the beginning of each session, the room was shown to the participant, together with the Stuff/wallet to be searched. We instructed them that we tried to simulate realistic hiding places for a lost wallet and that their search would be timed. Manual search had no additional support. If FiMS was used, we further explained the system's functionality and the user

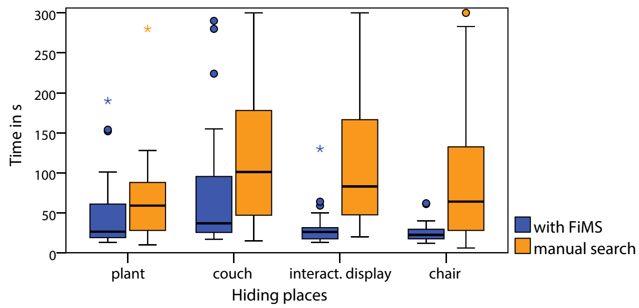


Figure 9: Search efficiency results of FiMS and manual search for the four search tasks in user study 1.

interface displayed on a tablet. Participants had to enter "wallet", hit the search button, and shortly wait for the calculated search cue to be displayed. The Stuff modules did not provide audiovisual feedback and caching was deactivated. We also explained that wrong cues are possible and that, if they suspected a wrong cue, they could trigger a new search (from within the room) or change to manual search. After the instruction and between each task, the participant was asked to leave the room and the Stuff/wallet was hidden at a hiding place. *Trash bin* served as an initial warm-up task for all and was excluded from results. The order of the other four tasks was counterbalanced. Once hidden, the door was closed again and participants started the search on their own. If FiMS was used, timing started when the participant started typing and an interim time was taken when they stopped interacting with the tablet. Otherwise, timing started when the participant opened the door (A4). Timing stopped when the participant signaled retrieval or after 300s, if the wallet had not been retrieved.

We had 48 participants in total, evenly split between the two groups (75% male, 25% female). Most participants were aged 20-26; 82% studied computer science or related subjects, the other participants had non-technical backgrounds.

Search Efficiency Results

Figure 9 shows the search time results of FiMS and manual search for each hiding place, including the delay (Mdn=9s) for interaction with FiMS (starting search, reading result). The box whiskers contain 95% of results, dots/asterisks mark outliers. We tested for statistical significance with the non-parametric Kolmogorov-Smirov Z-test. At the *plant*, median retrieval time of FiMS (Mdn=27s) was twice as fast as manual search (Mdn=59s). Yet, the difference was not significant, because FiMS provided only 50% correct cues (42% coarse, 8% wrong); likely due to the planter, which blocked the Stuff on the floor. At the *couch*, FiMS provided 79% correct cues (4% coarse, 17% wrong), resulting in significantly faster search time (Mdn=37s) than manual search (Mdn=101s) (Z=1.59, p<.05). For the last two tasks all result cues were correct, resulting in lower variance and significantly faster retrieval with FiMS in case of the *display* (Z=2.45, p<.001) and the *chair* (Z=2.17, p<.001).

For all tasks, FiMS sped up retrieval due to relational cues narrowing down the relevant search space. Correct cues resulted in very fast retrieval, but even coarse results provided

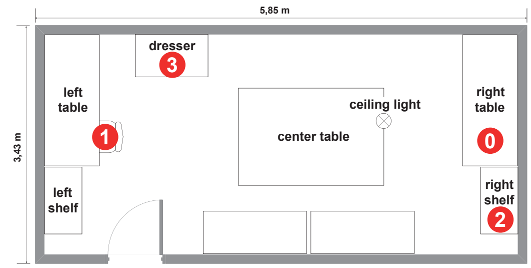


Figure 10: Lab layout and hiding places in user study 2.

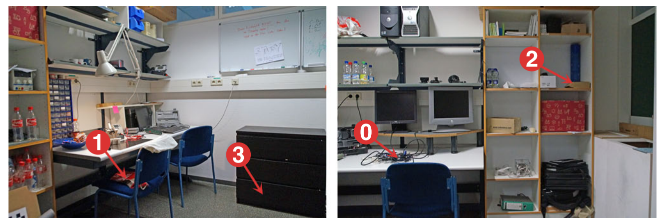


Figure 11: The four hiding places in user study 2.

support. Wrong cues were suspected as such by all participants when the Stuff could not be located after ca. 90s, causing almost 40% of participants to trigger a new search.

Study 2: Comparison with Other Search Cues

The first study showed that relative search cues lead to faster object retrieval than manual search. We conducted a second user study to compare our relative positioning approach to other types of search cues proposed in related work.

Setup and Procedure

In this study, we compared FiMS with two other cue types. Map-based position cues, as used in *IteMinder* [9], display an object's position on a room map. In the *SearchLight* [2] approach, a ceiling-mounted movable spotlight illuminates the object to assist retrieval. We further evaluated two variants of FiMS. Basic *FiMS* provides relative position cues only as in the first user study, while the full feedback FiMS (*FiMS FF*) combines relative position cues with audiovisual feedback by the located Stuff. We performed a between-subjects experiment in another lab (see Fig. 10), in which each participant was randomly assigned one of the 4 systems and had to search a switched-off mobile phone in 4 tasks. Figure 11 shows the hiding places. The mobile phone was hidden on the right *table* (T_0), on a *chair*, covered by a bag (T_1), in the right *shelf* within an open box (T_2), and inside the dresser's bottom *drawer* (T_3). T_0 was a warm-up task for all participants, the other tasks were counterbalanced.

As we were mainly interested in the effectiveness of the different search cues, we used a Wizard of Oz setup with fixed localization results to ensure consistent search cues for all participants. All four systems used the same search interface (see Fig. 2). Participants had to enter "mobile phone" on a tablet to start the search. Resulting search cues were system-dependent. IteMinder showed a room map with a red target mark and the text *"mobile phone localized at marked position."* SearchLight showed the text *"mobile phone has been*

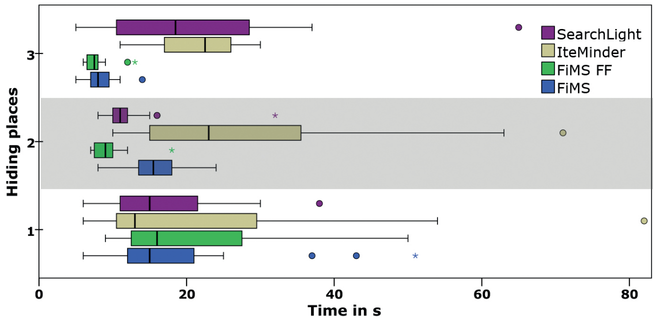


Figure 12: Search efficiency results of user study 2 for the three search tasks (extreme outliers not shown).

localized, a light beam shows you the position." The moving spotlight was simulated with a ceiling-mounted flashlight that was manually arranged between search tasks when re-hiding the mobile phone. FiMS and FiMS FF provided the relative position cues *"mobile phone is on right table"* (T_0), *"mobile phone is between center table and left table, near left table"* (T_1), *"mobile phone is in right shelf on second topmost level"* (T_2), and *"mobile phone is in bottom drawer of dresser"* (T_3). Stuff additionally blinked and beeped when located with FiMS FF.

Before the experiment, each participant was shown the lab and the mobile phone/Stuff to be searched. We further explained the functionality of their assigned system and that their search would be timed. Participants waited outside the room, while the mobile phone was re-hidden between tasks. Once hidden, the door was closed and participants could start search on their own. As starting the search process was the same for all systems, timing started once the search result was displayed on the tablet; it stopped when the participant retrieved the object or after 180s, if the object had not been retrieved.

In total, 96 participants (60% male, 40% female) completed the study, 24 for each system. Most participants were students aged 20-26. Almost 57% studied computer science or related subjects, the others had non-technical backgrounds.

Search Efficiency Results

Figure 12 shows search time results for tasks T_1–T_3. For each task, we used Kruskal-Wallis tests to determine significant differences between systems. T_1 showed no significant differences, the object's concealed position apparently affected all systems similarly. We found significant differences for T_2 ($H(3)$=48.94, p<.05) and T_3 ($H(3)$=55.32, p<.05) and employed Mann-Whitney U-tests with Bonferroni correction for post-hoc analysis.

Search time of FiMS was significantly faster than IteMinder for T_2 (U=170.5, r=-.35) and T_3 (U=2.5, r=-.85), with a median difference of 7.5s and 14.5s; likely due to the missing height information in the map-based cue, which is provided by FiMS with in-furniture RFID localization. SearchLight was significantly faster than FiMS for T_2 (U=118.0, r=-.51), with a median difference of 4.5s; likely because the spotlight could very accurately illuminate the box containing the object, while FiMS was not aware of the occluding box. However, FiMS FF facilitated significantly faster retrieval of the object than SearchLight (U=143.5, r=-.44). For T_3, basic FiMS was also significantly faster than SearchLight (U=87.0, r=-.60) with a median difference of 10.5s. Because of the smart dresser, FiMS could provide very accurate position information, while the angle of SearchLight's spotlight was disadvantageous.

Usability

After completing all tasks, participants completed the post-study system usability questionnaire (PSSUQ) [12]. Perceived usability was consistently high for all systems, without any significant differences; likely because all participants perceived their respective system as useful. We plan to assess potential differences in perceived usability in a future within-subjects study. Yet, qualitative feedback provided by some participants still provided interesting insights. The lack of height information in IteMinder was frequently criticized, which matches the quantitative results. Some participants also had initial orientation problems with the map and would have liked to see themselves on it. The visibility of SearchLight's spotlight was also criticized. One participant switched off the room light to improve it. Another concern was that occluded objects cannot be directly targeted, e.g., objects behind a room divider. One participant stated that the beeping of FiMS FF was confusing and did not aid searching, especially in task T_1.

Overall, the results of study 2 show that the relative position cues of FiMS support physical-object search significantly better than map-based cues. Compared to SearchLight's cues, the relative position cues of FiMS are not affected by lighting conditions. The results of FiMS FF indicate that combining relative position cues with audiovisual feedback is helpful when the sought-for object is occluded or included in another object. While all evaluated cue types were perceived as usable, the relative position cues of FiMS led to shorter retrieval times.

RELATED WORK

Subsequently, we compare previously proposed systems for physical objects search with FiMS and against the requirements identified earlier. Table 1 provides a summary. Several systems require the user's presence and active participation in search. In *FindIT* [14], objects are equipped with low-power optical sensors; a special flashlight emits an optical beam to trigger audiovisual feedback of objects, similar to FiMS. While highly energy efficient due to the use of low-power sensors, objects must be in the user's visual range and cannot be occluded or included in other objects. *FETCH* [6] uses mobile phones and laptops to find objects tagged with Bluetooth modules. Once detected, an object starts beeping, requiring the user to be in range to perceive the search cue. Occlusions or inclusions could also make it difficult to hear the signal. Furthermore, the Bluetooth modules of their prototype require recharging every 2–3 weeks. Frank et al. [4] also utilize Bluetooth devices of other users to cover a larger search area. GPS and UMTS cell of origin are used to provide a rough position when the user is not in range.

Konishi et al. [10] avoid active search by assuming user-carried RFID readers, which periodically sense tagged objects in the user's proximity and store those snapshots. When searching for an object, their system returns a list of surrounding objects from matching snapshots. This approach is energy and cost efficient and also supports localization of occluded and included objects. However, it relies on the assumption that lost objects are always surrounded by other tagged objects and that the user knows the position of at least one of those objects. The user must also carry an RFID reader at all times. The aforementioned *IteMinder* [9] uses an autonomous robot to continuously scan the environment. Known locations equipped with passive RFID tags serve as reference points. The system provides a map-based search cue, by showing a found object's location in a 2 m range around a reference point on a map. Drawbacks are the required time for full environment scans and installation of reference points. Hallberg et al. [5] measure RSSI of active reference tags with fixed RFID readers to improve localization with the LANDMARC approach [16]. Drawbacks are initial calibration overhead and optimal placement of reference tags. It is not clear if their system can handle occlusions and how position results are presented to users. Nakada et al. [15] combine active RFID tags for localizing occluded objects with ultrasonic positioning of uncovered objects. A found object is either illuminated by a ceiling-mounted spotlight or gives acoustic feedback. Thus, search cues are only effective if the user is in physical range. Furthermore, ultrasonic systems are costly and require precise calibration.

Brownie [17] combines ultrasonic positioning with a ceiling camera and accelerometers attached to objects in order to find covered objects. When an object's ultrasonic signal is lost, the camera tries to detect a "container" (e.g., a box) at the object's last known location and tracks the container's movements together with the object's accelerometer to derive the new position. Thus, it is assumed that containers are always in visual range of the camera. *SearchLight* [2] uses only a ceiling camera to find objects tagged with visual markers. An initial scan stores the camera's pan and tilt angle for each located object. A found object is illuminated by a movable spotlight, as used in the second user study. SearchLight can only locate objects in the camera's line of sight and requires a lengthy scan process. The use of visual markers is energy and cost efficient. *DrawerFinder* [8] uses cameras placed above boxes to take a picture of the box's content when opened. The user must manually browse through all pictures when searching an object and obviously cannot find objects in other places.

Lamming and Bohm [11] equip objects with small battery-powered devices (SPECs). Mobile SPECs have a battery lifetime of about one month; stationary SPECs of one year. SPECs broadcast their ID via an IR transmitter and log received IDs of surrounding SPECs. The history of received IDs is used to locate objects. Stationary SPECs are similar to FiMS SmartFurniture, but are not interconnected. Thus, sighting histories are distributed across all SPECss and can only be uploaded to a server when a gateway device is nearby. Furthermore, objects need to be located in visual range of another SPEC to be located. Like FiMS, Snoogle [19] combines a ZigBee mesh network of sensors and objects with a hierarchical topology. Objects register with a room's Index Point (IP), which is registered to a key IP, e.g., of the building. Room level IPs are comparable to having only one FiMS SmartFurniture per room. Thus, Snoogle's localization granularity is limited to room level, while FiMS can provide relational cues involving multiple furniture pieces, and can search inside furniture. Security and privacy is provided by an asymmetric encryption scheme and UNIX-like access control. MAX [20] uses a hierarchy of *base-stations* on the room level and furniture-based *sub-stations* similar to FiMS. RSSI measurements of RFID are used to provide relative locations of sought-for objects e.g., "sunglasses located in bedroom at desk". In contrast to FiMS, their search results cannot reflect ambiguities about the position, e.g., when the sought-for object is located between multiple sub-stations rather than near one. Also in contrast to FiMS, MAX does not address inclusion of objects and provides no further search cues once the user is looking at the stated position. Security and privacy on the object and room level are supported by MAX with asymmetric encryption, access control is enforced by base-stations.

	user-indep. localization	occlusions inclusions	intuitive search cues	seamless configura-tion	energy / cost efficiency	privacy security
FindIT [14]	○	○	◐	n/a	●	○
FETCH [6]	○	◐	◐	n/a	◐	○
Frank et al. [4]	●	◐	◐	n/a	◐	○
Konishi et al. [10]	◐	●	◐	n/a	●	○
IteMinder [9]	●	○	◐	○	●	○
Hallberg et al. [5]	●	◐	○	○	●	○
Nakada et al. [15]	●	◐	◐	◐	◐	○
Brownie [17]	●	◐	◐	◐	◐	○
SearchLight [2]	●	○	◐	◐	●	○
DrawerFinder [8]	◐	◐	○	n/a	●	○
SPECs [11]	●	○	◐	◐	◐	○
Snoogle [19]	●	◐	◐	◐	◐	●
MAX [20]	●	◐	◐	◐	◐	●
FiMS	●	●	●	◐	◐	●

not supported (○), partially supported (◐), supported (●)

Table 1: Comparison of existing object search approaches.

DISCUSSION AND CONCLUSIONS

In summary, FiMS is a system for physical objects search that provides user-independent localization by employing a hierarchical search model, which ranges from inside furniture to remote locations. FiMS SmartFurniture is equipped with RFID readers and directed ZigBee antennas to support localization of objects inside and in relation to furniture. An active Stuff module is attached to objects to make them searchable. Our current Stuff prototype is small enough to be attached to a key chain and can operate for at least 2 months with two coin batteries.

FiMS is based on relative positioning of objects in relation to multiple SmartFurniture rather than exact localization of objects. While MAX [20] also uses signal strength to determine proximity between an object and one furniture, our approach supports positioning in relation to multiple SmartFurniture and can provide advanced search cues in which furniture serve as landmarks, e.g., "wallet is between dresser and couch, near couch." Directed antennas in SmartFurniture not only sense nearby Stuff, but also automatically determine relative orientations of furniture pieces to each other, without requiring user calibration or knowledge of the furniture's actual position. Our position accuracy experiments with two SmartFurniture showed that our approach is sufficiently robust in realistic settings (85–87% accuracy) and not impacted by present persons. User evaluation showed that our relative position cues lead to faster object retrieval than manual search, even for coarse results. In a Wizard of Oz setup, our cues performed also significantly better than map-based and spotlight-based cues, in most tasks.

SmartFurniture contains multiple active components that incur costs, which can be traded off against search granularity. While replacing all furniture in a home with SmartFurniture will remain prohibitive in the near future, 1-2 SmartFurniture per room already provide a significant advantage in the retrieval of occluded objects. A limitation in such a scenario would be localization of objects in un-equipped furniture. While retrofitting furniture could also be possible, it requires additional configuration and calibration effort. Therefore, we plan to investigate another approach, more in line with the goal of seamless configuration. A single, low-cost FiMS component with only one antenna could be placed in a room's light fixture or power outlet, to provide the information in which room an object is located. Such components could be easily deployed in multiple rooms in the user's home and connected environments, e.g., her office.

We are further planning a longitudinal study in actual homes, in order to gain insights on how relative position cues aid users in locating objects they misplaced themselves. We also plan to investigate search cues in relation to three or more SmartFurniture and further miniaturization of Stuff tags, in order to facilitate direct integration into physical objects.

ACKNOWLEDGMENTS

The authors would like to thank all study participants, the anonymous reviewers for their valuable feedback, Enrico Rukzio for extensive comments on earlier drafts, and Frank Bögelsack for assistance with antenna evaluation.

REFERENCES

1. Ahmed, S., and Menzel, W. A novel planar four-quad antenna. In *Proc. EUCAP '12*, IEEE (2012).
2. Butz, A., Schneider, M., and Spassova, M. SearchLight – a lightweight search function for pervasive environments. In *Proc. PERVASIVE '04* (2004).
3. Esure Home Insurance. We're a bunch of 'losers'. `www.esure.com/media_centre/archive/wcmcap_100800`, 2012.
4. Frank, C., Bolliger, P., Roduner, C., and Kellerer, W. Objects calling home: Locating objects using mobile phones. In *Proc. PERVASIVE '07*, Springer (2007).
5. Hallberg, J., Nugent, C., Davies, R., and Donnelly, M. Localisation of forgotten items using RFID technology. In *Proc. ITAB '09*, IEEE (2009).
6. Kientz, J. A., Patel, S. N., Tyebkhan, A. Z., Gane, B., Wiley, J., and Abowd, G. D. Where's my stuff?: Design and evaluation of a mobile system for locating lost items for the visually impaired. In *Proc. Conf. on Computers and Accessibility (Assets '06)*, ACM (2006).
7. Knierim, P., Nickels, J., Musiol, S., Könings, B., Schaub, F., Wiedersheim, B., and Weber, M. Find my stuff: a search engine for everyday objects (demo). In *Proc. MUM '12*, ACM (2012).
8. Komatsuzaki, M., Tsukada, K., and Siio, I. DrawerFinder: finding items in storage boxes using pictures and visual markers. In *IUI '11*, ACM (2011).
9. Komatsuzaki, M., Tsukada, K., Siio, I., Verronen, P., Luimula, M., and Pieskä, S. IteMinder: finding items in a room using passive RFID tags and an autonomous robot (poster). In *Proc. UbiComp '11*, ACM (2011).
10. Konishi, S., Kawakita, Y., and Ichikawa, H. Method for estimation of distance between objects and its application for finding lost objects. In *Proc. IEEE CCNC '12*, IEEE (2012).
11. Lamming, M., and Bohm, D. SPECs: another approach to human context and activity sensing research, using tiny peer-to-peer wireless computers.
12. Lewis, J. R. IBM computer usability satisfaction questionnaires: psychometric evaluation and instructions for use. *Int. J. HCI 7*, 1 (1995), 57–78.
13. Liang, C.-J. M., Priyantha, N. B., Liu, J., and Terzis, A. Surviving wi-fi interference in low power ZigBee networks. In *Proc. SenSys '10*, ACM (2010).
14. Ma, H., and Paradiso, J. The FindIT flashlight: Responsive tagging based on optically triggered microprocessor wakeup. In *UbiComp '02* (2002).
15. Nakada, T., Kanai, H., and Kunifuji, S. A support system for finding lost objects using spotlight. In *Proc. MobileHCI '05*, ACM (2005).
16. Ni, L. M., Liu, Y., Lau, Y. C., and Patil, A. P. LANDMARC: indoor location sensing using active RFID. *Wireless Networks 10*, 6 (2004), 701–710.
17. Satake, S., Kawashima, H., and Imai, M. Brownie: Searching concealed real world artifacts. In *Proc. Conf. on Networked Sensing Systems (INSS '07)*, IEEE (2007).
18. StickNFind Technologies. StickNFind, 2013. `http://www.sticknfind.com/`.
19. Wang, H., Tan, C., and Li, Q. Snoogle: A search engine for the physical world. In *INFOCOM '08*, IEEE (2008).
20. Yap, K.-K., Srinivasan, V., and Motani, M. MAX: human-centric search of the physical world. In *Proc. SenSys '05*, ACM (2005).
21. Zanca, G., Zorzi, F., Zanella, A., and Zorzi, M. Experimental comparison of RSSI-based localization algorithms for indoor wireless sensor networks. In *Proc. REALWSN '08 workshop*, ACM (2008).