Find My Stuff: A Search Engine for Everyday Objects

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

ABSTRACT

Searching for lost keys, wallets or mobile phones is a common nuisance. Compared to digital information, search support for physical objects is very limited. We propose Find My Stuff (FiMS) as a search engine for physical objects. We built a fully functional Arduino-based prototype. FiMS offers the users a simple search interface to locate tagged physical items in different indoor environments. A hierarchical search process ensures energy efficient and effective searches. Instead of a fixed search infrastructure, the localization system is based on SmartFurniture equipped with RFID readers and ZigBee modules. Search results provide intuitive search cues based on relative positioning to support users in the physical retrieval of their lost objects. The system requires no manual calibration and is robust against rearrangement of SmartFurniture. Safety mechanisms prevent abuse of the system and protect user privacy.

Categories and Subject Descriptors

[**Human-centered computing**]: Ubiquitous and mobile computing systems and tools

Keywords

localization, relative positioning, ZigBee, RSSI, RFID

1. INTRODUCTION

An average person misplaces up to nine items per week and spends about 15 minutes per day searching them.[1] Mobile phones, keys, and sunglasses are among the most frequently lost items. Losing those items can cost time, money, and cause headaches. While searching for digital information is a well supported everyday task, searching for common physical objects is hardly supported by current technology. Simplistic key finders[2] exist that announce an object's position by acoustic or visual signals, but do not give a precise position and require the user to be in physical proximity. Mobile phone trackers enable remote localization of lost phones with GPS, WiFi, or based on cell of origin. However, such approaches are not applicable for smaller objects, such as keys or wallets, and have difficulties with objects indoors.

In order to fill this gap, we developed *Find My Stuff (FiMS)* as a search engine and localization system for finding misplaced physical objects in common indoor environments, like homes or offices. FiMS offers a typical search engine interface to let users search for lost objects. Search results indicate the object's location and provide intuitive search guidance by leveraging relative positioning of objects and furniture (e.g., "the wallet is between couch and table"). In addition to relative positioning information, FiMS can optionally trigger acoustic and visual signals for respectively equipped items.

Instead of relying on pre-installed, dedicated, room-wide localization systems, we assume that it will be possible to equip everyday objects with inexpensive and energy efficient minimal processing and communication capabilities, in the near future. FiMS is based on respectively enabled furniture that can be arbitrarily arranged and added to the system without requiring manual calibration. Physical objects are made searchable by attaching or integrating a *Stuff* tag, which supports ZigBee and passive RFID. Search queries for an object are processed hierarchically, starting with the object's last known location (e.g., in a drawer) and extending to the room and other connected environments.

2. RELATED WORK

Similar to commercial key finders, several research approaches require the user's presence and active participation in search. Kientz et al. [1] use mobile phones and laptops to find objects tagged with bluetooth modules. Once detected, the objects start beeping. Frank et al. [2] extend this idea by also utilizing the bluetooth devices of other users in order to cover a larger search area. The position of other devices is determined by their UMTS cell information. Thus, search results can be very coarse.

Konishi et al. [3] avoid active search by users by assuming that they carry RFID readers periodically sensing tagged objects in their environment and storing those snapshots in a database. When searching a specific object, the system returns the list of surrounding objects. The user must know at least the location of one of those in order to find the desired object. Instead of user-carried RFID readers, Komatsuzaki et al. [4] use an autonomous robot to continuously scan the environment. Hallberg et al. [5] use RSSI measurement of active reference tags at known locations with fixed RFID readers to improve localization with the LANDMARC approach [6]. A drawback is the initial calibration overhead

[1] `http://www.esure.com/media_centre/archive/wcmcap_100800.html`
[2] `http://en.wikipedia.org/wiki/Key_finder`

MUM '12, Dec 04-06 2012, Ulm, Germany
ACM 978-1-4503-1815-0/12/12.

and finding optimal locations for reference tags.

Nakada et al. [7] propose a search system based on active RFID tags and ultrasonic positioning, which allows to locate uncovered objects with about 5cm accuracy. A found object is either illuminated by a movable spotlight at the ceiling or gives acoustic feedback. The main drawback of this approach is that covered objects cannot be localized by the ultrasonic system and therefore rely on the imprecise RFID detection. In order to address the problem of covered objects, Satake et al. [8] combine an ultrasonic positioning system with a ceiling camera and acceleration sensors attached to objects. When an object's signal is lost, the camera tries to track movements of a "container" (e.g., a box) to derive the new position.

Butz et al. [9] use only a ceiling camera to find objects tagged with visual markers. A searched object is illuminated by a movable spotlight. This approach can only locate objects in the camera's line of sight and requires a lengthy scan process. A camera system for finding objects inside boxes is proposed by Komatsuzaki et al. [10]. A camera above the boxes detects the opening of a box and takes a picture of it's content. The user must manually browse through pictures when searching an object.

Like FiMS, the Snoogle system [11] utilizes a ZigBee-based mesh network of sensors and objects. Objects register with the Index Point (IP) of the current room, which in turn registers to a key IP (e.g. of the building). However, the localization granularity of Snoogle is limited to the room level. The MAX system [12] also uses a hierarchy of *base-stations* on the room level, but introduces furniture-based *sub-stations*. RSSI measurements are used to determine which piece of furniture the searched object is closest to in order to report a location e.g., "sunglasses located in bedroom at desk". However, search results do not reflect ambiguities about the position, e.g., when the searched object is located between two or more sub-stations rather than near a specific one.

3. FIMS ARCHITECTURE & FEATURES

The discussion of related work shows diverse approaches for supporting physical object search. Some approaches merely augment the search process and require manual user interaction, while other approaches focus on determining the environment of the searched object rather than the specific location in that environment. A design goal of FiMS was to bridge this gap. Users should be able to use a simple search engine interface to determine the object's place and its specific location at that place regardless of their own location. Furthermore, search results should provide intuitive cues that facilitate the object's physical retrieval, even if the searched object is occluded by other objects. As another design issue, we wanted to reduce the reliance on fixed localization systems as much as possible. Homes and offices are organic environments, which are frequently rearranged and changed to match the users' needs. FiMS should be able to operate in such an environment without requiring manual recalibration or frequent maintenance. Also, registering new objects should be simple and intuitive. Last but not least, any distributed search system should be secure against eavesdropping and protect the privacy of users.

To achieve these goals, FiMS is based on a hierarchical search process and the general idea of dynamically utilizing furniture as reference points for relative positioning. In contrast to related work, we employ different localization mechanisms on different hierarchy levels to conserve energy and maintain scalability.

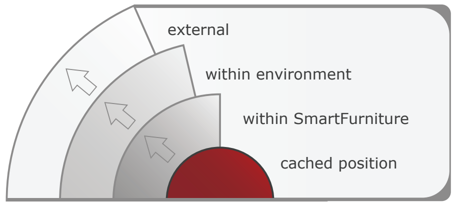

Figure 1: Hierarchical layered search model.

3.1 Search and System Architecture

Localizable objects in the FiMS system are tagged with a small module, called *Stuff*. FiMS processes search queries for a specific object according to a hierarchical layered model shown in Figure 1. Starting at the center of the model, the object's last known position is queried. If unsuccessful, the search is first extended to the *insides* of FiMS-enabled *SmartFurniture* near that position. If the Stuff cannot be located inside any SmartFurniture, SmartFurniture is used to locate Stuff inside the room. If also unsuccessful, the search is extended beyond the room to other environments associated with the user, such as other rooms in the apartment, the user's office, or authorized environments of friends. The search terminates with a search result as soon as the requested Stuff is found. We use different localization technologies on the layers to improve search speed and reduce energy consumption. Inside SmartFurniture, we use RFID localization with passive tags. On a room level we use ZigBee for wireless communication and localization.

Objects with Stuff modules can be located anywhere inside or around SmartFurniture. Stuff and SmartFurniture communicate via ZigBee and form a mesh network with Stuff modules as ZigBee end devices, SmartFurniture as ZigBee routers, and a specific furniture piece—the *StarterKit*—as the ZigBee coordinator. One or more RFID readers per SmartFurniture enable localization of Stuff inside or directly on top of it.

In addition to ZigBee and RFID, SmartFurniture also has a WiFi module in order to communicate with the search server. We opted for WiFi for the server connection because existing WiFi installations in homes and offices can be leveraged without requiring additional infrastructure. The server manages the search process and maintains the user's Stuff information. The server sends measurement requests to specific SmartFurniture and receives RSSI measurements if the object could be located, which are then used to determine the object's relative position. Although the server can reside locally, e.g., as part of the StarterKit, Internet access is required to enable users to start FiMS searches when not at home and also to enable cooperation with FiMS servers from other environments the user authorized for local search. The server provides a personalized web interface to trigger FiMS searches and manage the user's Stuff. Figure 2 gives an overview of the FiMS architecture.

If the desired object cannot be located within the user's environment, the search can be extended to other FiMS-enabled environments, e.g., the office. This requires a prior authorization process in which the remote FiMS system shares its ZigBee network information with the user's FiMS server

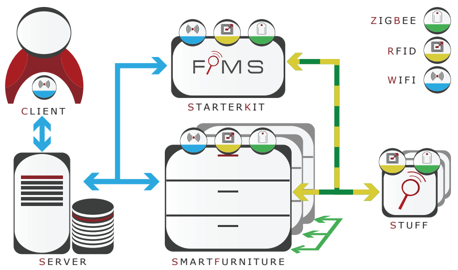

Figure 2: FiMS architecture overview.

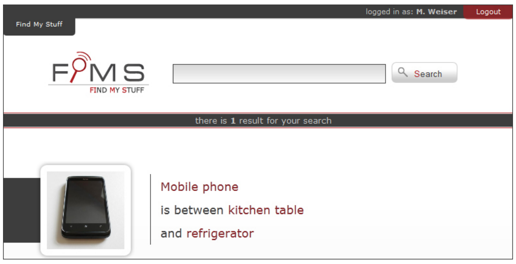

Figure 3: Search result with relative position.

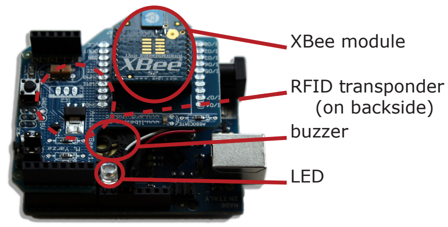

Figure 4: Arduino-based Stuff prototype.

Figure 5: SmartFurniture with integrated ZigBee and RFID modules and parabolic antennas

to enable localization of the user's Stuff in the remote environment. Note that users are always restricted to search for their own Stuff and cannot locate the Stuff owned by others.

3.2 Relative Positioning

Localization of Stuff follows the introduced layered search model. Once the desired Stuff could not be found with RFID localization, relative localization with ZigBee is triggered. The server sends a message to the Stuff via SmartFurniture. The Stuff sends beacons as broadcast messages. Each SmartFurniture that receives the beacons measures received signal strength (RSSI) for all its antennas. These values are used to calculate the direction of the Stuff. Direction estimation and RSSI values are reported to the server's result queue. Note that the signal strength is interpreted as a distance metric in this model, which can be problematic due to variations caused by external influences [13]. To compensate for this, values are checked for consistency and a weighting factor can be included reflecting the number of received beacons. The resulting position is returned as the search result together with a stored image of the searched object (Figure 3).

3.3 Seamless Configuration and Security

Two types of configuration tasks arise in our system: registering new Stuff and integrating or rearranging SmartFurniture. The configuration tasks require some user interactions in order to provide authentication and authorization of devices and to help users to form a mental model of their FiMS system. The challenge is to minimize manual configuration effort while providing sufficient security to prevent tracking of users or unauthenticated localization of their valuables. FiMS leverages spatial proximity of RFID tags and readers to reduce configuration overhead. Any items can be configured without having to cope with object IDs, passwords, or pairing keys. The StarterKit in each FiMS-enabled environment is equipped with an additional ZigBee module and RFID reader, which serve as a physical interface for registering new objects and pairing SmartFurniture. User authentication is realized with personal RFID tags. An additional supervisor tag is capable of generating user tags and pairing new SmartFurniture.

4. PROTOTYPE

We developed a fully functional prototype of FiMS based on Arduino microcontroller boards[3]. Our system currently consists of a StarterKit, a FiMS-enabled dresser, and multiple Stuff modules. We used XBee Series 2 modules for ZigBee communication. For RFID, we used readers at 125kHz and World Tag passive RFID transponders from HID Global.

Our prototype Stuff modules consist of an Arduino Uno board and an XBee module with integrated antenna (Figure 4,). A passive RFID tag is attached to the board. Additionally, the Stuff prototype is equipped with a bright LED and a buzzer, which can be used to guide users with audio-visual feedback. Energy requirements of Stuff modules are very low, because passive RFID tags do not consume any energy and the ZigBee module can be in sleep mode unless the specific Stuff is actively being searched. While not ready for everyday use due to size, our Stuff prototype serves well to demonstrate the feasibility of FiMS. We are planning to reduce the size of future Stuff designs by employing the smaller Arduino Pro Mini.

The SmartFurniture prototype (Figure 5) is a dresser with three drawers. We equipped each drawer with an RFID reader and a blue LED to highlight the Stuff's location inside the drawer. Opening the drawer to take out the searched Stuff automatically switches off the LED. In the middle drawer, we further added three directed antennas for localization, pointing left, right and ahead. In order to enhance the antennas' directional characteristics, we evaluated differ-

[3]http://arduino.cc

Figure 6: StarterKit with LEDs for visual feedback.

ent reflector designs (parabolic/plane/none, in addition to a 2 dbi antenna) in a measurement series. Parabolic reflectors exhibited the best directional shielding results thus facilitating more accurate direction estimation based on RSSI. The SmartFurniture's XBee modules, RFID readers, and LEDs are connected to two Arduino Mega boards, which handle localization tasks. From the outside, the prototype looks like a regular furniture except for the LEDs at each drawer.

The StarterKit (Figure 6) is similarly equipped as the SmartFurniture prototype. One of the three XBee modules acts as the coordinator of the ZigBee network. The Arduino Mega is directly connected to the FiMS server, also contained in the StarterKit. The Stuff registration interface is provided by a *registration plate* on top of the StarterKit, which houses the additional XBee module and RFID reader. A camera mounted above the registration plate takes pictures of new Stuff to illustrate search answers (Figure 3).

5. CONCLUSION

We proposed Find my Stuff as a usable and intuitive search engine for physical objects. FiMS leverages a hierarchical search model that employs fitting localization mechanisms for different search areas. Inside SmartFurniture, objects are located with RFID. On a room level, ZigBee RSSI measurements are used to determine the objects position relative to present SmartFurniture. FiMS can further extend search to other connected environments the user is authorized to search in. The hierarchical search approach is energy efficient because components only become active on demand and also inherently privacy-aware by not extending search queries beyond the required area. Search results from all levels are integrated into a relative positioning approach which provides relative descriptions, such as "wallet is between couch and table closer to table". FiMS can be easily extended with new objects and additional SmartFurniture. Configuration is seamless and requires only minimal user interaction while ensuring security of communication and enforcing user authentication requirements. Due to automatic reconfiguration, FiMS is also robust against rearrangement of furniture.

Currently, our Stuff modules are slightly too large to be attached to everyday objects. However, we are working on a smaller version, based on the Arduino Pro Mini. This new Stuff is soon ready for use. Also we are planning to evaluate the accuracy of SmartFurniture as well as the usability of FiMS. Ultimately, FiMS technology could already be integrated into physical objects and off the shelf furniture could support objects search and other context-aware services.

6. REFERENCES

[1] J. A. Kientz, S. N. Patel, A. Z. Tyebkhan, B. Gane, J. Wiley, and G. D. Abowd, "Where's my stuff?: Design and evaluation of a mobile system for locating lost items for the visually impaired," in *Proc. 8th Int. Conf. on Computers and Accessibility (Assets '06)*. ACM, 2006.

[2] C. Frank, P. Bolliger, C. Roduner, and W. Kellerer, "Objects calling home: Locating objects using mobile phones," in *Proc. PERVASIVE '07*. Springer, 2007.

[3] S. Konishi, Y. Kawakita, and H. Ichikawa, "Method for estimation of distance between objects and its application for finding lost objects," in *Proc. IEEE CCNC '12*. IEEE, 2012.

[4] M. Komatsuzaki, K. Tsukada, I. Siio, P. Verronen, M. Luimula, and S. Pieskä, "IteMinder: finding items in a room using passive RFID tags and an autonomous robot (poster)," in *Proc. UbiComp '11*. ACM, 2011.

[5] J. Hallberg, C. Nugent, R. Davies, and M. Donnelly, "Localisation of forgotten items using RFID technology," in *Proc. 9th Int. Conf. on Information Technology and Applications in Biomedicine*. IEEE, 2009.

[6] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "LANDMARC: indoor location sensing using active RFID," *Wireless Networks*, vol. 10, no. 6, pp. 701–710, 2004.

[7] T. Nakada, H. Kanai, and S. Kunifuji, "A support system for finding lost objects using spotlight," in *Proc. MobileHCI '05*. ACM, 2005.

[8] S. Satake, H. Kawashima, and M. Imai, "Brownie: Searching concealed real world artifacts," in *Proc. 4th Int. Conf. on Networked Sensing Systems (INSS '07)*. IEEE, 2007.

[9] A. Butz, M. Schneider, and M. Spassova, "SearchLight – a lightweight search function for pervasive environments," in *Proc. PERVASIVE '04*. Springer, 2004.

[10] M. Komatsuzaki, K. Tsukada, and I. Siio, "DrawerFinder: finding items in storage boxes using pictures and visual markers," in *Proc. 16th Int. Conf. on Intelligent User Interfaces (IUI '11)*. ACM, 2011.

[11] H. Wang, C. Tan, and Q. Li, "Snoogle: A search engine for the physical world," in *Proc. INFOCOM '08*. IEEE, 2008.

[12] K.-K. Yap, V. Srinivasan, and M. Motani, "MAX: human-centric search of the physical world," in *Proc. 3rd Int. Conf. on Embedded Networked Sensor Systems (SenSys '05)*. ACM, 2005.

[13] G. Zanca, F. Zorzi, A. Zanella, and M. Zorzi, "Experimental comparison of rssi-based localization algorithms for indoor wireless sensor networks," in *Proc. workshop on real-world wireless sensor networks (REALWSN '08)*. ACM, 2008.