

Commenced Publication in 1973

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison

Lancaster University, UK

Takeo Kanade

Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler

University of Surrey, Guildford, UK

Jon M. Kleinberg

Cornell University, Ithaca, NY, USA

Alfred Kobsa

University of California, Irvine, CA, USA

Friedemann Mattern

ETH Zurich, Switzerland

John C. Mitchell

Stanford University, CA, USA

Moni Naor

Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz

University of Bern, Switzerland

C. Pandu Rangan

Indian Institute of Technology, Madras, India

Bernhard Steffen

University of Dortmund, Germany

Madhu Sudan

Massachusetts Institute of Technology, MA, USA

Demetri Terzopoulos

University of California, Los Angeles, CA, USA

Doug Tygar

University of California, Berkeley, CA, USA

Gerhard Weikum

Max-Planck Institute of Computer Science, Saarbruecken, Germany

Tanzeem Choudhury Aaron Quigley
Thomas Strang Koji Suginuma (Eds.)

Location and Context Awareness

4th International Symposium, LoCA 2009
Tokyo, Japan, May 7-8, 2009
Proceedings



Springer

Volume Editors

Tanzeem Choudhury
Dartmouth College
Computer Science Department
Hanover, NH 03755, USA
E-mail: tanzeem.choudhury@dartmouth.edu

Aaron Quigley
University College Dublin
School of Computer Science & Informatics
Complex & Adaptive Systems Laboratory
Belfield Office Park, Dublin 4, Ireland
E-mail: aaron.quigley@ucd.ie

Thomas Strang
German Aerospace Center
Institute of Communications and Navigation
82234 Wessling/Oberpfaffenhofen, Germany
E-mail: thomas.strang@dlr.de
and
University of Innsbruck
Semantic Technology Institute Innsbruck (STI2)
6020 Innsbruck, Austria

Koji Suginuma
Sony Corporation
Semiconductor Group R&D Division
1-11-1 Osaki, Shinagawa-ku, Tokyo 141-0032, Japan
E-mail: koji.suginuma@jp.sony.com

Library of Congress Control Number: Applied for

CR Subject Classification (1998): H.3, H.4, C.2, H.2.8, H.5, K.8

LNCS Sublibrary: SL 3 – Information Systems and Application,
incl. Internet/Web and HCI

ISSN 0302-9743
ISBN-10 3-642-01720-7 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-01720-9 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

springer.com

© Springer-Verlag Berlin Heidelberg 2009
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12678716 06/3180 5 4 3 2 1 0

Preface

These proceedings contain the papers presented at the 4th International Symposium on Location and Context Awareness (LoCA) during May 7–8, 2009 in Tokyo, Japan. Location and context awareness are fundaments to next-generation mobile and pervasive computing systems. Pervasive computing is a model of computing in which computation is everywhere and computer functions are integrated into everything. The ultimate aim is to make information, applications and services available anywhere and at anytime in the human environment in a fluid manner appropriate to our current *context*.

Once away from the desktop, we find ourselves in a wide variety of contexts and hence situations. For computing to be relevant and useful in these emerging situations we must rely on a range of contextual cues. Context includes physiological, environmental, and computational data, whether sensed or inferred. In addition, context includes details of a user's activities, goals, abilities, preferences, affordances, and surroundings. With location and context awareness we can expect computers to deliver information, services, and entertainment in a way that maximizes convenience and minimizes intrusion.

LoCA 2009 presented new and significant research on systems, services, and applications for detecting, interpreting, and using location and other contextual information. Developing this awareness involves research in sensing, systems, machine learning, human–computer interaction, and design. We sought technical papers describing original, previously unpublished research results including:

- New hardware platforms for sensing location and context
- Machine learning techniques for inferring user location and context from low-level sensor data
- Location and context representation, management, and distribution
- Privacy policies and communication protocols for location and context information

Our call for papers resulted in 77 abstracts, with 54 full papers being submitted to LoCA 2009. After careful peer review by our international Program Committee, 18 papers were selected for publication in these proceedings to form what we feel is a high-quality program. Three papers were nominated for the best paper award. We extend a sincere thank you to all the authors who submitted papers, to the 27 hard-working members of our Program Committee, and to all our external reviewers.

May 2009

Tanzeem Choudhury
Aaron Quigley
Thomas Strang
Koji Suginuma

Organization

Organizing Committee

Program Chairs	Tanzeem Choudhury (Dartmouth College, USA)
General Chair	Aaron Quigley (University College Dublin, Ireland)
Local Chair	Thomas Strang (DLR and UIBK, Germany)
Publicity Chairs	Koji Suginuma (Sony Corporation, Japan)
Local Publicity Chairs	Ashish Kapoor (Microsoft Research, USA) Alex Varshavsky (AT&T Research Labs, USA) Takeshi Iwamoto (KDDI Labs, Japan)

Program Committee

Hamid Aghajan	Stanford University, USA
Christian Becker	Universität Mannheim, Germany
Hung Bui	SRI International, USA
Andrew T. Campbell	Dartmouth College, USA
Lorcan Coyle	University College Dublin, Ireland
Anind Dey	Carnegie Mellon University, USA
Simon Dobson	University College Dublin, Ireland
Dieter Fox	University of Washington, USA
Mike Hazas	Lancaster University, UK
Jeffrey Hightower	Intel Research Seattle, USA
Johan Hjelm	Ericsson Research Tokyo, Japan
Ashish Kapoor	Microsoft Research, Redmond USA
Minkyong Kim	IBM Research Watson, USA
John Krumm	Microsoft Research, Redmond, USA
Marc Langheinrich	ETH Zurich, Switzerland
Rene Mayrhofer	University of Vienna, Austria
Nuria Oliver	Telefonica Research, Spain
Shwetak Patel	University of Washington, USA
Matthai Philipose	Intel Research, USA
Matthew Reynolds	Duke University, USA
Bernt Schiele	TU Darmstadt, Germany
Chris Schmandt	MIT Media Lab, USA
James Scott	Microsoft Research Cambridge, UK
Thomas Strang	German Aerospace Center (DLR)
Alex Varshavsky	AT&T Research Labs, USA
Chris Wren	Google, USA
Danny Wyatt	University of Washington, USA

Table of Contents

Multi Activity Recognition Based on Bodymodel-Derived Primitives	1
<i>Andreas Zinnen, Christian Wojek, and Bernt Schiele</i>	
Activity Inference through Sequence Alignment	19
<i>Driss Choujaa and Naranker Dulay</i>	
Improving Location Fingerprinting through Motion Detection and Asynchronous Interval Labeling	37
<i>Philipp Bolliger, Kurt Partridge, Maurice Chu, and Marc Langheinrich</i>	
Using Context Annotated Mobility Profiles to Recruit Data Collectors in Participatory Sensing	52
<i>Sasank Reddy, Katie Shilton, Jeff Burke, Deborah Estrin, Mark Hansen, and Mani Srivastava</i>	
Location Diversity: Enhanced Privacy Protection in Location Based Services	70
<i>Mingqiang Xue, Panos Kalnis, and Hung Keng Pung</i>	
Nonvisual, Distal Tracking of Mobile Remote Agents in Geosocial Interaction	88
<i>Steven Strachan and Roderick Murray-Smith</i>	
Directional Beacons: A Robust WiFi Positioning Method Using Angle-of-Emission Information	103
<i>Kensaku Kawauchi, Takashi Miyaki, and Jun Rekimoto</i>	
Bluetooth Tracking without Discoverability	120
<i>Simon Hay and Robert Harle</i>	
Error Estimation for Indoor 802.11 Location Fingerprinting	138
<i>Hendrik Lemelson, Mikkel Baun Kjærgaard, Rene Hansen, and Thomas King</i>	
Activity Recognition from Sparsely Labeled Data Using Multi-Instance Learning	156
<i>Maja Stikic and Bernt Schiele</i>	
A Framework for Creating and Using Maps of Privately Owned Spaces	174
<i>Steven Shafer</i>	
Daily Routine Recognition through Activity Spotting	192
<i>Ulf Blanke and Bernt Schiele</i>	

VIII Table of Contents

SofTOA: Software Ranging for TOA-Based Positioning of WLAN Terminals	207
<i>Marc Ciurana, David López, and Francisco Barceló-Arroyo</i>	
Position Estimation from UWB Pseudorange and Angle-of-Arrival: A Comparison of Non-linear Regression and Kalman Filtering	222
<i>Kavitha Muthukrishnan and Mike Hazas</i>	
An Ultrasonic 3D Positioning System Using a Single Compact Receiver Unit	240
<i>Masanori Sugimoto, Kan Tulathimutte, Toshio Ito, Tetsuya Sato, and Hiromichi Hashizume</i>	
LOCK: A Highly Accurate, Easy-to-Use Location-Based Access Control System	254
<i>Yongcai Wang, Junhui Zhao, and Toshikazu Fukushima</i>	
Characterizing the Space by Presenting Non-visual Information	271
<i>Tomohiro Akagawa, Takuji Narumi, Young Ah Seong, and Takashi Kiriyama</i>	
Author Index	283

Multi Activity Recognition Based on Bodymodel-Derived Primitives

Andreas Zinnen^{1,2}, Christian Wojek², and Bernt Schiele²

¹ SAP Research, CEC Darmstadt, Germany

² Computer Science Department, TU Darmstadt, Germany

{andreas.zinnen,wojek,schiele}@cs.tu-darmstadt.de

Abstract. We propose a novel model-based approach to activity recognition using high-level primitives that are derived from a human body model estimated from sensor data. Using short but fixed positions of the hands and turning points of hand movements, a continuous data stream is segmented in short segments of interest. Within these segments, joint boosting enables the automatic discovery of important and distinctive features ranging from motion over posture to location. To demonstrate the feasibility of our approach we present the user-dependent and across-user results of a study with 8 participants. The specific scenario that we study is composed of 20 activities in quality inspection of a car production process.

Keywords: Activity Recognition, Boosting, Human-Body Model.

1 Introduction

Human activity recognition is an important part of contextual information due to its potential in application domains ranging from medical diagnosis over human behavior modeling to industrial applications. Most previous approaches to activity recognition [1,14,7,10,3,5,19] rely on the use of *signal-oriented features* (such as mean, variance, and FFT-coefficients) and state-of-the-art machine learning techniques (such as HMM and SVM). However, the success of activity recognition in general is still far from being satisfactory in realistic and challenging real-world scenarios. Even state-of-the-art approaches are typically challenged by the recognition of short and non-repetitive activities, by the recognition of a large number of activities in a user-independent manner, and by spotting activities in large and continuous data streams. While previous work has aimed to address some of these challenges at least individually [4,17,6,8] we argue in this paper that the sole use of *signal-oriented features* limits the applicability of most state-of-the-art approaches to realistic scenarios.

The first main contribution of this work is that we follow a model-based approach where high-level primitives are derived from a human-body model. This is in contrast to most previous work that typically relies on signal-oriented features only. As human activities are composed of various sub-actions we derive various motion primitives such as *move the hands up*, *turn the torso* or *turn the arm*. Since these primitives are based on a human body model they are more

robust or can be even invariant to the variability of performing various activities. Besides motion primitives, we also calculate posture features and use location information. The second main contribution of the paper is the simultaneous recognition of multiple activities. For this we employ the joint boosting framework [11]. This framework allows to select discriminant features from a large pool of features that are shared across different activity classes. This makes the classification scheme not only more efficient but also allows to recognize multiple activities in a user-independent manner. The third contribution of the paper is a novel segmentation procedure. The main goal is the segmentation of activities in a continuous data stream thereby reducing the search space considerably. As we will show in the experiments the segmentation procedure enables efficient as well as effective recognition of multiple activities. Finally, we apply the proposed method to a car-quality control data set provided by [6].

The paper is structured as follows. In section 2 we describe the segmentation procedure. Section 3 describes the various motion primitives and posture features derived from a human body model. Section 4 explains how joint boosting can be used for multiple activity recognition. Section 5 summarizes the experimental results. Finally section 6 concludes and discusses the contributions of the paper.

2 Activity Segmentation in Continuous Data Streams

The following sections describe our novel approach for activity recognition. We first introduce our human body model based on inertial sensors placed at several positions of the subject's body (section 2.1). Based on this body model, we derive our novel segmentation procedure generalizing the work of [17] and [18]. As we will see in the experiments, this segmentation procedure results in a significant data reduction. The next step of the algorithm calculates multiple features and primitives ranging from motion over posture to location (section 3). Section 4 then describes how we apply the joint boosting framework [11] to reliably detect 20 different activities. Section 5 experimentally analyzes the performance of this novel algorithm on a previously published data set [6].

2.1 Body Model

Human activity recognition is directly linked to human motion and movement analysis. In this paper we are interested in the recognition of activities such as *open trunk* or *check hood gaps*. It is important to note that the exact execution of these activities may vary greatly between subjects and will often vary substantially even for the same individual due to personal preferences, fatigue and other reasons. As a result, signal-oriented features such as FFT-coefficients of body-worn sensors will also vary substantially. Any subsequent classification using such signal-oriented features only will be difficult and might be even impossible. By using a human body model we can calculate more general motion primitives such as *moving hands up* irrespective of the exact orientation of the hands and the exact execution of the movement. As a result, the subsequent classification becomes much easier. In the following we describe the calculation of

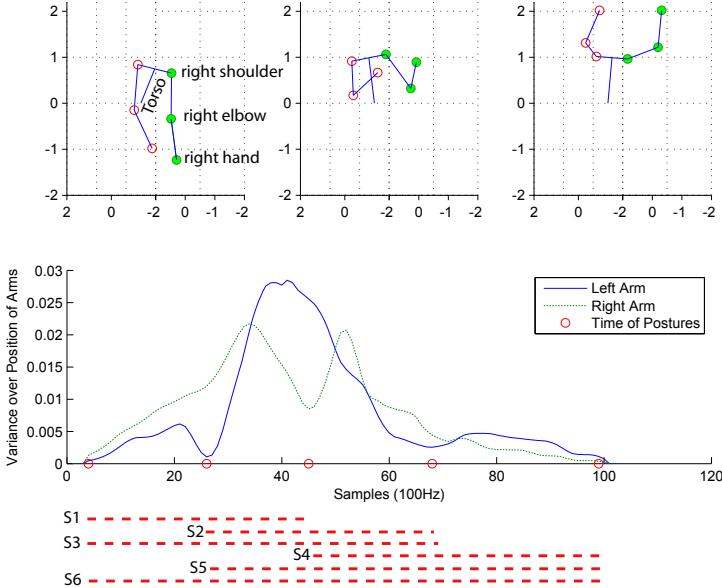


Fig. 1. Upper body model (top) and variance over hand positions (bottom) including segmentation while opening a hood based on 5 XSens inertial sensors

various such movement primitives as well as posture features. Our proposed upper body model concatenates the orientation information of five inertial sensors [15] located at the user's upper and lower arms and the torso. Figure 1 illustrates three snapshots of the resulting 3D body model (depicted as stick figures) while opening the hood of a car from a back perspective. For illustration, the left figure is labeled with the corresponding body extremities where the user moves down both hands to grasp the hood. Lifting both arms in the middle figure, the user reaches the final hand positions above his head illustrated in the right figure. While body models have been proposed for motion capturing by [16] and have been used for activity recognition in computer vision [20], we are not aware of similar work in the area of activity recognition using body worn sensors.

2.2 Segmentation

Segmentation of the continuous data stream may be seen as a filter for subsequent classification. Ideally the result of this filtering step are segments that are easier to classify than the continuous data stream. While performing activities, the movements (trajectory) can be described by a sequence of upper body postures as described in section 2.1. We observed previously [18] that many activities in a continuous data stream are enclosed by short but fixed positions of the hands (abbreviated SFP in the following). SFPs often occur at the beginning and end of activities. We also observed [17] that turning points (TP in the following) in hand movements can help to segment activities. For both SFPs and TPs the

variance in the hand position over a small time interval will be lower since the user slows down the speed of movement.

In the following, we illustrate this novel segmentation procedure using *Open Hood* as sample activity. Figure 1 displays snapshots of the trajectory while opening the hood. We calculate the variance over the hand positions in the 3D body model. As previously mentioned, the variance of the hand positions will be lower for SFPs and TPs in arm movements. Between SFPs and TPs, the variance will increase. The lower picture illustrates the variance of the user’s right hand positions as a dotted green and the corresponding left hand positions as a solid blue line in the course of the activity.

Note that calculating the variance over the lower arms’ orientation as applied in [17] is often not enough to identify all SFPs and TPs, e.g. when pulling the arm. Here, the orientation of the lower arm does not change and therefore does not indicate a SFP or TP of the hand. Therefore this novel segmentation procedure effectively generalizes both previous works [17,18].

For the segmentation we therefore detect local minima within the variance of the hand positions separately for both hands. Each local minimum gives us a potential SFP or TPs enclosing an activity (*cf.* red circles in lower figure 1). In the subsequent activity classification (section 4), we will have to test on segments of a specific minimum and maximum length enclosed by pairwise combinations of the detected local minima. As can be seen in the figure (red dashed lines), the displayed time interval contains six segments (S1-S6) complying with the time constraints that are evaluated in the later classification step.

To quantify the quality of the segmentation algorithm we evaluated the performance on the entire activity set as well as on the background set. Overall, the data set ([6], described in section 5.1) contains more than 280 minutes. The mean length of the analyzed activities is 1.4 seconds with a standard deviation of about 0.95 seconds. As the segmentation procedure is used as a filter for the subsequent activity recognition step, it is important that all activities are contained within these segments. This is the case as the average distance between the annotated start and end times of activities and a detected SFP or TP is less than 0.15 seconds. The distance is actually lower than the expected variance in annotation accuracy.

These numbers show that the algorithm is capable of finding the relevant SFPs or TPs and segments. Furthermore, the segmentation procedure results in a significant data reduction. Within the data (1,691,418 samples), less than 34,000 segments have to be evaluated. For each of these segmentations, the algorithm extracts various primitives and features to enable reliable activity recognition. The following section introduces three types of features for this purpose.

3 Calculation of Primitives and Features

The previous section introduced a method to segment a continuous data stream considering SFPs and TPs. This section describes how we extract discriminant

features from the data to enable reliable activity recognition. The analyzed activities can be performed in various ways. Not only the speed has an impact on the execution of an activity. A user who for instance opens a car door can use both hands in various ways. Considerable variability also occurs when opening or closing the hood or the trunk, where the user is not constrained to use a specific arm. In addition, a change of the user's position to the object will influence the motion trajectory. The main objective of section 3.1 is the identification of general movement primitives, such as moving the arms up or down (e.g. while opening the hood) or pulling the arm toward the body or turning the torso (e.g. while opening a car door). These primitives are based on the 3D body model as presented in section 2.1. The second feature set considers postures within the segments of interest (Posture Features). The last features are calculated on location data using an ultra-wide band (UWB) system from Ubisense [12] attached to the user's chest (Location Features).

3.1 Motion Primitives

Taking into account the 3D body model from section 2.1, we are now looking for primitives characterizing basic arm movements like up or down (Height Primitives), push and pull (Push-Pull Primitives), the bending of the body forward and back (Bending Primitives) and arm twisting (Twist Primitives). Additionally, a fixed car position leads to similar arm movement directions for different people and instances of the same activity. Therefore, we also calculate histograms on the movement direction of both hands (Direction Histograms).

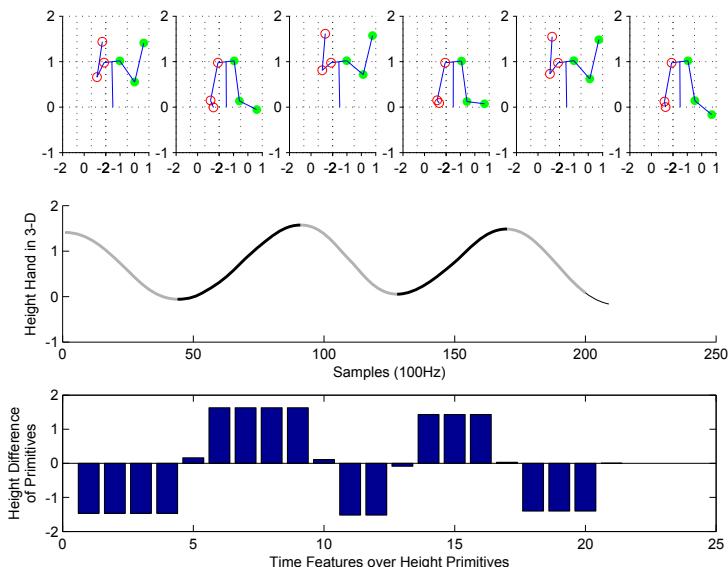


Fig. 2. Illustration of height features

Height Primitives. Figure 2 illustrates a series of snapshots of the activity *Check Trunk* where both hands are recurrently moved up and down in order to test the trunk’s hinges. As an example, the middle plot displays the right hand’s height calculated using the 3D model (left hand analog). By applying a sequential minimum-maximum search, we divide the segment into areas of up (black) and down (gray) movements as color-coded in the figure. In this example, we detect five height primitives.

In the next step, features are calculated on the detected primitives. A temporal coding (time features over primitives) of a fixed length is illustrated in the lower part of figure 2. The segment is divided into twenty equally spaced bins. Each bin is assigned the height difference of the associated height primitive (20 features). Furthermore, we add the number, average, maximum and minimum of up and down primitives for the analyzed segment (8 features). Finally, a histogram of the primitives’ length, normalized by the segment’s overall length, is included (5 features). All together, 66 height features for both hands are included in the subsequent training and classification steps.

Push-Pull Primitives. The calculation of push-pull features is similar to the height features. To enable the calculation of these features we introduce the following preprocessing step. We project the data points of the 3D body model onto the plane defined by the gravity vector as the normal.

Figure 3 illustrates a resulting sequence of snapshots for the recurrent pushing and pulling of the right arm from the bird’s-eye view, as it e.g. occurs for the activity *Check Right Door Lock*. The black arrows emphasize the movement’s direction of the right hand at each point in time. As previously mentioned, various motions can indicate the illustrated hand movement. First, the positions of the hand and elbow towards the torso change. Similarly, a turning of the torso is a sign for the push and pull movements.

Figure 4 depicts the start (dotted black) and end (solid blue) postures while pulling the arm towards the body. The first push-pull primitive is calculated on the angle included by the connection lines of the torso to the right hand at the beginning of the segment and the current point in time (*cf.* left figure 4). Obviously, this angle will increase when pulling the arm and decrease again while pushing.

The upper figure 5 represents the angle at each point in time in the push-pull segment (*cf.* figure 3) including three times pulling and two times pushing. Analogously to the height primitives, we apply minimum-maximum search to segment

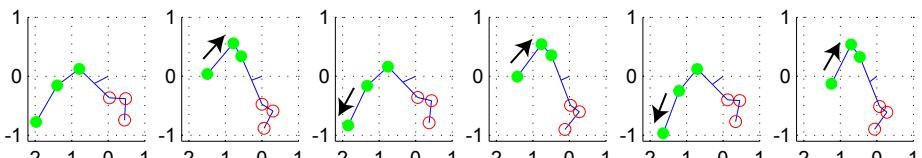


Fig. 3. Illustration of recurring push pull

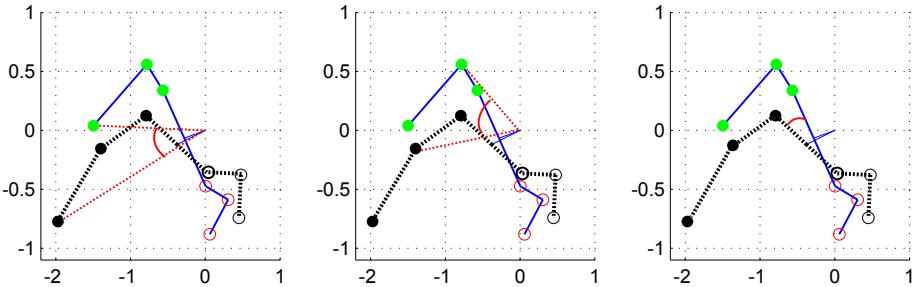


Fig. 4. Primitives extracted by torso turn, hand and elbow movements

five primitives as color-coded in figure 5. Similarly, we approach the second kind of push-pull primitives. This time, we calculate the angle included by the connection lines of the torso to the right elbow at the beginning of the segment and the current point in time (see middle plot of figure 4). The resulting primitives can be seen in the middle of figure 5. As motivated before, the last push-pull primitive is the turning of the torso as given by the angle between the body axis at the beginning of the segment and the current time (see figure 4 right).

As before, several features are calculated on the detected primitives (compare to paragraph 3.1). Beyond a temporal coding (time features over primitives) of a fixed length and a normalized histogram of the primitives' length, we add the number, mean, maximum and minimum value for each push-pull primitive. 165 push-pull features (66 hands, 66 elbows, 33 torso) are added to the feature set.

Bending Primitives. In addition to the use of arms and the turning of the torso, one can observe that users often bend the torso to support the activity, e.g. when opening the hood or opening two doors at once. To calculate primitives describing the forward and backward bending of the torso, the minimum-maximum search is applied to the angle included by the torso's direction and the gravity. Clearly, this angle increases/decreases when bending forward/backward. The features are calculated analogically to the height and push-pull features (33 features).

Twist Primitives. Twist primitives are motivated by activities that cause the user to twist one or two arms, e.g. when turning a knob. They are not calculated using the body model like the previous primitives, but directly on the integrated gyroscope data of the lower arms. A rotation of a lower arm in a clockwise direction leads to an increasing sum, whereas a counterclockwise turn causes a decrease. Minimum-maximum search over the sum yields the twist primitives. The features are calculated analogically to the primitives before (66 features for left and right arm).

Direction Histograms. A fixed car position leads to similar arm movement directions for different persons and instances while performing an activity.

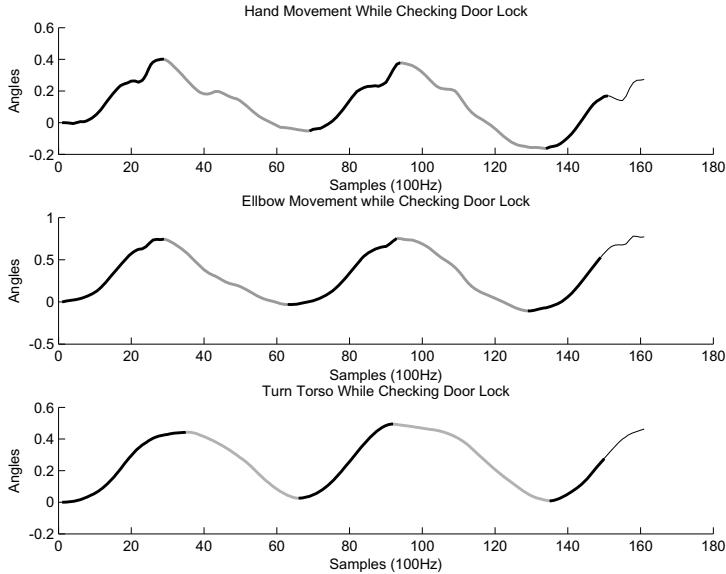


Fig. 5. Segmented push-pull primitives by torso turn, hand and elbow movements

Therefore, we finally calculate histograms on the movement direction of both hands (Direction Histograms). For illustration, we will again use the previous example of a pull movement. In a first step, we calculate the directional vectors of succeeding hand positions. The directional vectors for the pull activity are displayed as arrows in figure 6 (left). We define eight directions in the two dimensional plane as illustrated in the right picture. The directional vectors of the segment are then assigned to eight bins of a histogram represented by eight directions. This assignment leads to the direction histogram illustrated in the middle picture of figure 6. We observe high values for bin two and three. Visually this distribution is a good representation for the performed movement in the left picture of figure 6.

To make the feature robust to speed-difference the histogram is normalized by the overall length of the current segment. Also, we use soft assignment to the two neighboring bins where the assignment is proportional to the angle of the directional vectors. An example is illustrated in figure 6 by the red short arrow in the right figure. Here the direction lies between the directions of *bin 1* and *bin 2*. Therefore, it is assigned with a percentage of $\frac{\alpha}{45}$ to *bin 1* and with a percentage of $\frac{\beta}{45}$ to *bin 2*. The resulting 16 direction features for both hands are added to the feature set.

3.2 Posture Features

As seen in the last paragraph, activities can be decomposed into motion primitives. One can now ask the question whether discriminant features over body

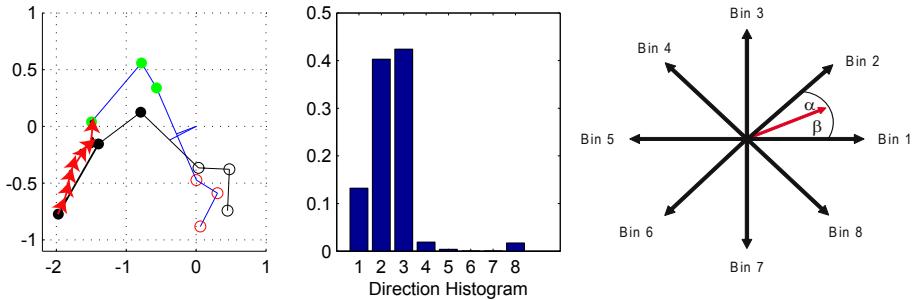


Fig. 6. Pull movement (left), direction histogram of the right hand movement (middle), direction definition for the histograms (right)

postures can help improve the recognition results. Looking at the activity *Fuel Lid*, it is clear that all users have to keep the right hand in a similar height to contact the fuel knob. It should also be noted that this height level will not change much while performing the activity. This observation is different from activity *Open Hood* where the arm height is changing very quickly. While writing, both hands remain in a similar height and orientation towards gravity. Here, one hand holds the notepad and the other writes with a pen. In the following we will consider postures that can help to distinguish between activities and the background. In the experiments described below, we consider the following postures: the arms' orientation towards gravity (6 dimensions), the distance between the two hands (1 dimension) and the hands' height (2 dimensions), as well as the torso's relative direction to the car (2 dimensions). We add the minimum, maximum, mean and variance over the postures (44 features) to the feature set.

3.3 Location Features

The worker's relative position to the car body is estimated using an ultra-wide band (UWB) system by Ubisense. Sensor data of four tags is smoothed with a Kalman filter before the four sensor sources are combined. Despite the use of multiple sensors, the data becomes patchy at certain times. In a preprocessing step, we estimate location values for these gaps by linearly interpolating between known values in time. Based on these location values, two different location representations are calculated and added to the feature space. First, the median of the enclosed x and y-values of a specific segment are added (2 features). Second, we calculate the angle included by the first basis vector and the connection of a specific location point to the car center M. The median of this angle for a segment is also added to the features (1 feature).

Figure 7 illustrates the accuracy of the preprocessed location data for the twenty annotated activities. While location is a good indicator which subset of activities might be performed, it is clear that location alone will not be sufficient to distinguish all activities. Especially behind and on the sides of the car are various overlapping areas for different activities. In the case of the activity *Writing*

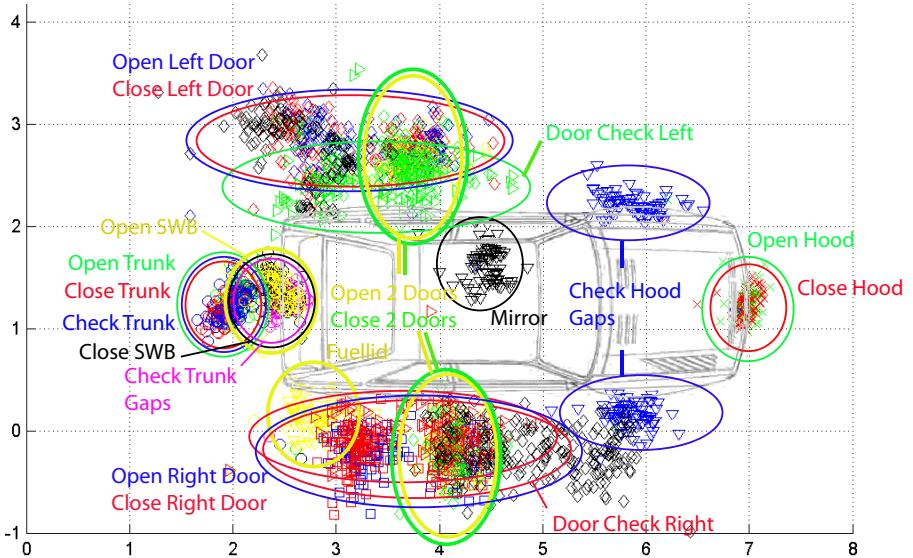


Fig. 7. Location of the 20 activities after preprocessing

(see black diamonds), location will not be helpful as the activity is spread over the entire area.

4 Multi-Activity Recognition Using Joint Boosting

The final step of our algorithm classifies the different segments into multiple activities using the motion, posture and location features introduced in the previous section. We employ joint boosting as it allows to learn a runtime efficient classifier by sharing features across multiple activities.

4.1 Joint Boosting

Boosting [2] is a widely used state-of-the-art machine learning technique for classification that has been applied to many different tasks such as camera-based object detection [13]. The basic idea is to combine multiple weighted weak classifiers $h_m(x)$ in order to form a strong classifier H . Often, decision tree stumps and regression stumps on a single dimension d of the feature vector x_m^d are used as weak classifiers. In this work we employ regression stumps defined as:

$$h_m(x) = \begin{cases} a_m & \text{if } x_m^d > \theta_m \\ b_m & \text{if } x_m^d \leq \theta_m \end{cases}$$

with θ_m being an automatically determined threshold. Torralba *et al.* [11] extend the idea of boosting to multi-class problems in a way that weak classifiers can be shared across multiple classes while the strong classifiers are learned jointly.

Thus, contrary to binary boosting the weighted error is not only sought to be reduced for a single strong classifier in each round, but for a whole set of strong classifiers. However, the number of possible sharing sets is 2^C where C is the number of overall classes. Clearly, an exhaustive search over all combinations even for a moderate number of classes is infeasible. As proposed by [11] we adopt a greedy best first search strategy to obtain an approximation to the best sharing set, which reduces complexity to $O(C^2)$. Torralba *et al.* show that surprisingly few weak classifiers are required for joint boosting to obtain the same classification performance compared to independently learned classifiers. Hence, classification runtime is reduced, in particular, when the number of classes becomes large.

More formally, in each boosting round m joint boosting selects the weak classifier which reduces the weighted squared error most for a set of classes S_m . The final strong classifier for a class c given a feature vector x is:

$$H_c(x) = \sum_m h_m(x) \delta(c \in S_m)$$

Our experiments (section 5.2) indicate that groups of similar activities are separated during the first rounds and are disambiguated among each other in later boosting rounds.

To further reduce the number of required rounds we counted mis-classification on background samples as error for the activity classifiers but did not learn a classifier for the background class. In order to handle the highly unbalanced training sample distribution among classes we adapted the weight initialization such that for each activity the class specific instance weights for positive samples as well as for negative samples sum up to 0.5.

4.2 Calculation of Test and Training Vectors

Activities in the training data are annotated. Therefore, we can directly calculate all features (section 3) on the positive training segments. The resulting feature vectors can be used as direct input for the training phase of the joint boosting algorithm.

Calculation of the features for the negative training segments and the test data is more complex. As previously discussed in section 2, we can automatically detect SFPs and TPs in the continuous data stream. However, the number that we detect is larger than the number needed to segment the activities from the background stream (see S1-S6 in Figure 1).

From the training data we extract one minimal and one maximal length over all considered activities and allow for $\pm 30\%$ variance of these values. When calculating the features in the test data and the training data (negative instances), all possible combinations of segments with length between the minimum and maximum length are analyzed. For all resulting segments, the features are calculated. Note that a negative instance on the training data is only used as input for the training phase of the joint boosting algorithm if it does not overlap with any activity instance. All test segments are classified by the boosting algorithm. As the writing

activity is considerably longer than all other activities, we split the annotation of activity *Writing* into sequences shorter than four seconds. The advantage is that the overall maximal length of all activities becomes shorter and the overall number of segments that need to be classified drops significantly. The only drawback is that we obtain multiple segments classified as writing for a single continuous writing activity which have to be merged in a post-processing step.

5 Experiments and Results

The specific scenario that we study is composed of 20 activities in quality inspection of a car production process. The next paragraph shortly introduces the previously published data set [6]. Subsequently, we evaluate the distinctiveness of each activity separately with respect to the collected background data and the remaining activities.

5.1 Car-Quality Control Data Set Provided by [6]

The data were collected in a large industrial project on the use of wearable technology in production environments [9]. During the experiments, data of a wearable system composed of 7 motion sensors [15], 16 force sensing resistors (FSR) for lower arm muscle monitoring and 4 ultra-wide band (UWB) [12] tags for tracking user position were collected. In this paper, we use the data of 5 motion sensors (located at the user's upper and lower arms and torso) and the 4 UWB tags for activity classification. A list of the twenty activities can be seen in Figure 8.

5.2 Evaluation of the Model-Based Approach to Activity Recognition

In the following, we evaluate the distinctiveness of each activity separately with respect to the collected background data and the remaining activities. In each validation round, we calculate the probability for all detected segments. As proposed in [6], we evaluate each segment as follows: if a considered segment overlaps more than 50% with the annotation of a specific activity, it will be counted as a true positive. Clearly the number of false positives grows while increasing the recall for the actual activity. Ideally both precision and recall are 100%. The following paragraphs report on two types of experiments. In the *User-Dependent Experiments*, we analyze the performance of the recognition system for user-dependent training and evaluation. This is the same setting as used in [6]. In *Across-User Experiments*, we report for user-independent training and testing. In *General Results*, we summarize general conclusions and analyze in more detail how joint boosting works for our activity classes.

User-Dependent Experiments. As mentioned before, each user performed the activity sequence 10 times. We perform leave-one-out cross-validation for

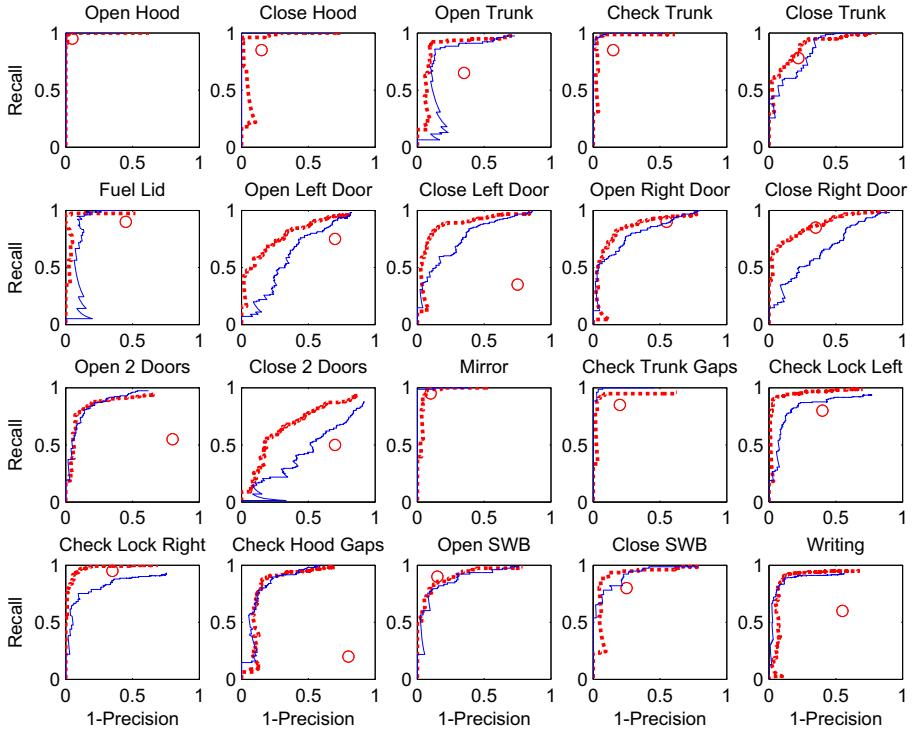


Fig. 8. ROC curves for the 20 activities (dotted red: user-dependent, solid blue: across-user). The red circles are the user-dependent results of Ogris et al. [6] (SWB = spare wheel box)

each user. The red dotted plot in Figure 8 summarizes the results for each activity.

For ten out of twenty activities, the system achieves very good performance with a recall and precision greater than 0.9. For six of those activities, an almost optimal performance can be observed. The activities *Check Hood Gaps* and *Writing* follow closely with a recall of 0.85 and a precision of 0.85. Most door related activities and *Open Spare Wheel Box* still obtain a recall of 0.75 for a precision of 0.75. *Close Two Doors* performs worst with a recall 0.6 for a precision of 0.75.

As our experiments show, the new recognition method yields significant performance improvements compared to the user dependent approach introduced by [6] (see red circle in the figure) in 14 out of 20 activities. For five activities our system performs similarly. Only for activity *Open Spare Wheel Box* (Open SWB), our system is marginal worse as can be seen in the dotted red plot.

Across-User Experiments. Beyond user dependent experiments, we perform leave-one-user-out cross-validation to analyze the system performance across users. As we have data from eight different users for our activity set we

effectively perform eight-fold cross validation across the eight users. Interestingly, the novel approach proposed in this paper still achieves good detection rates across users. Whereas the system achieves similar results for ten out of twenty activities, the precision-recall ratio drops for door related activities. Considering the difficulty of the dataset, the results show that the proposed features and primitives derived from a human body model make activity recognition robust even across user. In three cases, the results of the across-user experiments are better than the user-dependent (*Close Hood, Mirror, and Check Trunk Gaps*) because of more representative data and a larger training set. Interestingly, the across-user results of our system still outperform the user-dependent results introduced by [6] for 13 of the 20 activities while performing similarly for three.

General Results. Many of the activities with lower recognition performance are door related. Visual inspection of the corresponding body models indicates that magnetic disturbances occur for these activities. This might be a reason for the decreased performance. Additionally, opening and closing car doors allows for a greater degree of variability during execution than most other activities.

To gain insight into how joint boosting works on this data set, it is helpful to examine which features are selected and when. As previously mentioned, joint boosting shares weak classifiers across multiple classes.

Figure 9 shows the final set of features selected in the first 25 rounds and the sharing matrix that specifies how the different features are shared across the 20 activity classes. Each row corresponds to one feature and each column shows the features used for each activity class. An entry (gray box) in cell (j, i) means that activity i uses feature j . The darker the entry, the stronger does the corresponding feature vote for the class. Lighter entries are weighted less.

In the first round, the algorithm selects height primitives to separate activity classes 1 to 5 and 13 from the remaining classes. The selected activities share distinctive height primitives while interacting with the hood, trunk or mirror. The second round characterizes activities with low height primitives. In rounds number three and four, boosting considers more features shared by many classes. Whereas the first location feature is shared by all activities in the back of the car, the torso direction helps to characterize activities in the front and on the front left side of the car.

The figure illustrates that the features are shared between classes in a way that is not tree-structured. A hierarchical partitioning is therefore not possible. However, the figure also shows how joint boosting reduces the computational complexity, by finding common features that are shared across several classes. Already after less than five rounds, features shared by all classes are considered. Furthermore, all features used in the first ten rounds are shared by at least four classes.

In the last paragraph we described how boosting is able to reduce the weighted squared error on the training data within a few rounds by finding common features shared across several classes. The features selected by joint training in the first rounds are generic location and height features, whereas the features chosen in the later rounds tend to be more activity specific.

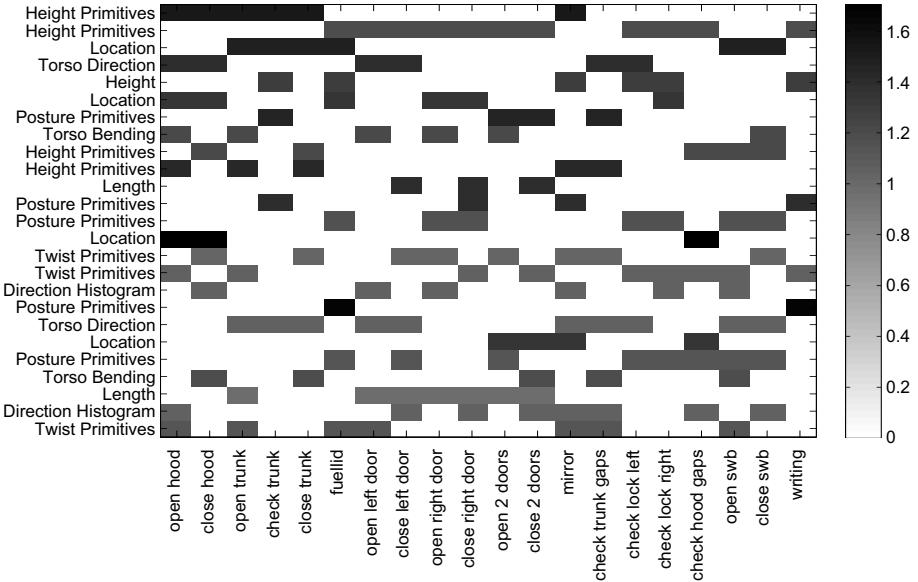


Fig. 9. Matrix that relates features to classifiers from top to bottom, which shows which features are shared among the different activity classes. These features were chosen from a pool of 393 features in the first 25 rounds of boosting.

To gain a little more insight into the obtained classifier we discuss several classes in more detail. Quite interestingly, most introduced features and all primitives are used. The activities *Open/Close Hood*, *Open/Check/Close Trunk*, *Mirror*, *Open SWB*, *Close SWB* and *Check Trunk Gaps* can be characterized by height primitives. In addition, boosting finds bending primitives to be discriminant for those activities. Furthermore, it is not surprising that twist primitives strongly describe activity *Fuel Lid*. Characteristic twist primitives over the entire activity's duration can also be observed for *Open Hood/Trunk*, *Mirror* and *Check Trunk Gaps*. For most door-related activities and *Check Hood Gaps*, the push-pull primitives and the direction histograms become prominent. These activities are mainly characterized by a planar movement of the hand. To support the activities, the users bend the body forward and backward. Posture primitives are mainly used for activities *Check Trunk*, *Check Trunk Gaps*, *Writing*, *Fuel Lid*, *Open Two Doors* and *Close Two Doors*. Beyond the mean and variance of the hands' height (distinctive against background), the distance between the two hands is characteristic for the same activities. Whereas *Open/Close Two Doors*, *Check Trunk* and *Check Trunk Gaps* necessitate a broad distance between the hands, the user keeps the hands close to each other for *Writing* and *Fuel Lid*. Finally, the orientation of the left and right arm towards gravity is specific for the activity *Writing*.

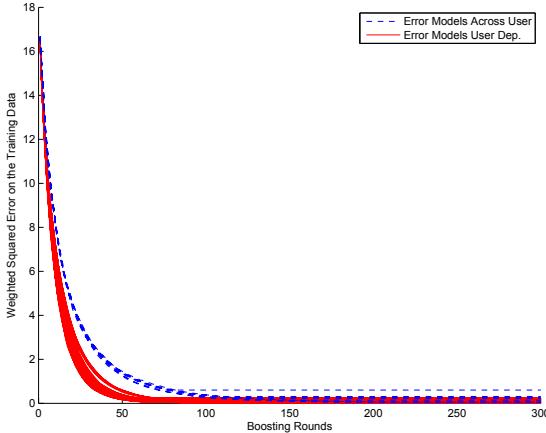


Fig. 10. Weighted squared error on the training data

As previously mentioned, most of the features introduced in Section 3 are used by the boosting framework. One can wonder how many classifiers or rounds are needed by boosting to obtain reasonable results for the twenty activities. Figure 10 plots the weighted squared error on the training data (y-axis) as a function of the boosting rounds (x-axis). The solid red curves correspond to the eight user-dependent models. As can be seen, the error on the trained models converges after 40-60 rounds. Not surprisingly, in case of across-user training, boosting requires more classifiers (100-120) to converge. This fact shows that activity recognition across user is a more complex task as the variability of activity performances is higher, and therefore boosting has to find more weak classifiers to obtain good results.

6 Discussion and Conclusion

We present a novel model-based approach to activity recognition using high-level primitives that are derived from a 3D human-body model. Using short but fixed positions of the hands and turning points of hand movements, the continuous data stream is segmented in short segments of interest. The method is successfully employed for a car-quality control setting of 20 different activities in a continuous data stream. In the experiments our approach shows superior performance with published results on the same data set (based on string matching).

Our representation of motion and posture primitives on segments has proven effective in both user dependent and across-user activity recognition. The introduced high-level features such as *move the hands up* and *down* or *turn the torso* or *arm* allow recognition even in the presence of large variability while performing activities.

Our results confirm the strength of the introduced features. Unlike signal-oriented features, body-model features are closely linked to motions that are required when performing an activity, for instance raising one or two arms when opening the hood. The exact execution is not of great importance but rather features like the end position of the hands described by the body-model. First results comparing signal-oriented versus body-model features confirm a superior performance of body-model derived features. In 12 out of 20 activities, we obtain better results. For 6 activities, the results of both approaches are similar. For the remaining 2 activities, our approach is marginal worse than the approach based on signal-oriented features.

Beyond the modeling aspects, we pay particular attention to computational feasibility. Based on the segmentation, the evaluation can be calculated in real-time. The feature calculation using non-optimized MATLAB code for the data set of 280 minutes takes 230 minutes. The succeeding evaluation is performed in 7 minutes. Indeed, the runtime will increase when recognizing more activity classes. However since joint boosting is able to share features among the classes efficiently, the runtime will still be real-time.

References

1. Deng, J., Tsui, H.: An HMM-based approach for gesture segmentation and recognition. In: 15th Int. Conf. on Pattern Recognition, vol. 2, pp. 679–682 (2000)
2. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: a statistical view of boosting. *The Annals of Statistics* 38(2), 337–374 (2000)
3. Kallio, S., Kela, J., Korppiä, P., Mäntyjärvi, J.: User independent gesture interaction for small handheld devices. *IJPRAI* 20(4) (2006)
4. Lester, J., Choudhury, T., Kern, N., Borriello, G., Hannaford, B.: A Hybrid Discriminative/Generative Approach for Modeling Human Activities. In: Proc. of the International Joint Conference on Artificial Intelligence (IJCAI) (2005)
5. Mäntylä, V.-M., Mäntyjärvi, J., Seppänen, T., Tuulari, E.: Hand gesture recognition of a mobile device user. In: IEEE International Conference on Multimedia and Expo. (2000)
6. Ogris, G., Stiefmeier, T., Lukowicz, P., Tröster, G.: Using a complex Multi-modal On-body Sensor System for Activity Spotting. In: 12th IEEE International Symposium on Wearable Computers, Pittsburgh, USA (2008)
7. Ogris, G., Kreil, M., Lukowicz, P.: Using FSR based muscle activity monitoring to recognize manipulative arm gestures. In: Int. Symp. on Wear. Comp. (October 2007)
8. Stiefmeier, T., Roggen, D., Tröster, G.: Gestures are strings: efficient online gesture spotting and classification using string matching. In: Proceedings of the ICST 2nd international conference on Body area networks (2007)
9. Stiefmeier, T., Roggen, D., Ogris, G., Lukowicz, P., Tröster, G.: Wearable activity tracking in car manufacturing. *IEEE Pervasive Computing* 7(2), 42–50 (2008)
10. Stiefmeier, T., Ogris, G., Junker, H., Lukowicz, P., Tröster, G.: Combining motion sensors and ultrasonic hands tracking for continuous activity recognition in a maintenance scenario. In: Int. Symp. on Wear. Comp. (October 2006)
11. Torralba, A., Murphy, K.P.: Sharing Visual Features for Multiclass and Multiview Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 29(5) (2007)

12. Ubisense, <http://www.ubisense.de/content/14.html>
13. Viola, P.A., Jones, M.J.: Robust real-time face detection. *Int. Journal on Comp. Vision* 57(2), 137–154 (2004)
14. Ward, J.A., Lukowicz, P., Tröster, G., Starner, T.: Activity recognition of assembly tasks using body-worn microphones and accelerometers. *IEEE Trans. Pattern Analysis and Machine Intell.* 28(10), 1553–1567 (2006)
15. XSens Motion Technologies, <http://xsens.com/>
16. XSens Moven, http://www.moven.com/en/home_moven.php
17. Zinnen, A., Schiele, B.: A new Approach to Enable Gesture Recognition in Continuous Data Streams. In: 12th IEEE International Symposium on Wearable Computers, September 28 - October 1 (2008)
18. Zinnen, A., Laerhoven, K., Schiele, B.: Toward Recognition of Short and Non-repetitive Activities from Wearable Sensors. In: Schiele, B., Dey, A.K., Gellersen, H., de Ruyter, B., Tscheligi, M., Wichert, R., Aarts, E., Buchmann, A. (eds.) *AmI 2007. LNCS*, vol. 4794, pp. 142–158. Springer, Heidelberg (2007)
19. Pylvänäinen, T.: Accelerometer based gesture recognition using continuous HMMS. In: Marques, J.S., Pérez de la Blanca, N., Pina, P. (eds.) *IbPRIA 2005. LNCS*, vol. 3522, pp. 639–646. Springer, Heidelberg (2005)
20. Ryoo, M.S., Aggarwal, J.K.: Recognition of Composite Human Activities through Context-Free Grammar Based Representation. In: *Computer Vision and Pattern Recognition*, vol. 2, pp. 1709–1718 (2006)

Activity Inference through Sequence Alignment

Driss Choujaa and Naranker Dulay

Imperial College London, London SW7 2AZ, United Kingdom
`{dfc05,nd}@doc.ic.ac.uk`

Abstract. Activity inference attempts to identify what a person is doing at a given point in time from a series of observations. Since the 1980s, the task has developed into a fruitful research field and is now considered a key step in the design of many human-centred systems. For activity inference, wearable and mobile devices are unique opportunities to sense a user’s context unobtrusively throughout the day. Unfortunately, the limited battery life of these platforms does not always allow continuous activity logging. In this paper, we present a novel technique to fill in gaps in activity logs by exploiting both short- and long-range dependencies in human behaviour. Inference is performed by sequence alignment using scoring parameters learnt from training data in a probabilistic framework. Experiments on the Reality Mining dataset show significant improvements over baseline results even with reduced training and long gaps in data.

1 Introduction

Activity inference has gained considerable interest over the past twenty years both as a goal in itself and as an intermediate step in the design of advanced systems. Important research efforts have been driven by a broad range of applications including entertainment [1] [2], logistics [3], [4], [5], healthcare [6], [7], surveillance [8], [9] and military support [10]. A great number of human-centred systems presented in the literature make use of wearable or mobile devices such as GPS receivers, PDAs and cellphones to collect context data throughout a person’s day. However, mobility and ubiquity come at the price of major downsides.

Mobile and wearable devices can stop collecting data for a variety of reasons. First, autonomous appliances are fundamentally constrained by their short battery lives. Secondly, the limited computational and memory resources available on certain devices may not be sufficient to process inputs in due time or store long histories. Thirdly, some sensors just cannot capture information in specific places – for example GPS receivers in indoor environments. Furthermore, logging applications may crash unexpectedly or be turned off intentionally. For instance, cellphone carriers are usually invited to shut down their devices in theatres.

Many services are affected by interruptions in context sensing and standard activity inference techniques perform poorly when faced with long missing intervals in input data. One reason for this is that popular models such as low-order hidden Markov models are unable to take advantage of long-range dependencies

in human behaviour. In particular, the first-order Markov assumption requires that the true state be conditionally independent of all earlier states given the immediate previous state. In contrast, a human observer can often guess what a known person has been doing from past behaviour and activities performed at remoter times of the day. For example, if someone went to bed late at night, he will probably wake up late in the morning at the same location.

In this paper, we make two main contributions. First, we quantify the relative importance of short- and long-range dependencies in human behaviour using a standard set of cellphone data. Secondly, we present a novel biology-inspired technique which fills in missing intervals in activity logs by exploiting both short- and long-range dependencies in human behaviour. Inference is performed by sequence alignment using scoring parameters learnt from training data in a probabilistic framework. Experiments on the Reality Mining dataset [11] show significant improvements over baseline results even with long missing intervals and reduced training.

The remainder of this paper is divided as follows. We first discuss related work on mobile activity inference and then investigate the relative importance of short- and long-range dependencies in human activities. In Section 4, we present our sequence alignment method for activity inference. Large-scale experiments are presented and discussed in Section 5. Finally, Section 6 summarises our work and outlines three major directions for future research.

2 Related Work

Few activity inference methods for mobile phones can operate without context data at the time of inference. One notable exception is the *eigenbehaviour* approach developed by Eagle and Pentland in [12]. The authors of this paper represent structure in daily human behaviour by eigenbehaviours, the principal components of an individual’s behavioural data in the Reality Mining dataset. Eigendecomposition proved very successful in computer vision and object recognition in particular. In [12], a user’s day is approximated by a weighted sum of his primary eigenbehaviours. Calculating weights halfway through the day of selected users, the day’s remaining behaviours are predicted with high accuracy. Eagle and Pentland also showed that users of similar demographics can be clustered into a ‘behaviour space’ and classified in their correct group affiliation almost perfectly.

Although eigenbehaviours are very powerful tools in group inference, we believe that they have serious limitations in activity inference. First, the idea that a human day can be approximated by a weighted sum of a few particular routines is rather simplistic. Many relationships between activities are lost in the process and ‘rare’ activities – potentially the most interesting ones – are almost certainly always missed by the first eigenbehaviours and considered noise. In addition, eigenbehaviours are not robust to time shifts and variations in activity durations. Last but not least, computing reliable eigenbehaviours requires a significant amount of user data which takes a long time to collect.

3 Quantifying Short- and Long-Range Dependencies

Most existing activity inference techniques exhibit *temporal short-sightedness*. In other words, they only take into account relationships between the values of certain variables at a small number of subsequent time-slices – typically, one or two. In particular, this is the case of most Hidden Markov Models and Dynamic Bayesian Networks designed for activity inference. While short-range dependencies are often very significant, they can sometimes be weaker than longer-range dependencies. In order to illustrate this statement, let us consider the case of a typical working person. If that person commutes to work in the morning, he will almost certainly commute back home in the evening, whatever activities he may have performed in the meantime.

One popular measure of the mutual dependency of two discrete random variables X and Y is the *mutual information* [13] between them:

$$I(X, Y) = \sum_{x,y} P_{XY}(x, y) \cdot \log \left(\frac{P_{XY}(x, y)}{P_X(x) \cdot P_Y(y)} \right),$$

where P_{XY} is the joint probability function of X and Y , and P_X and P_Y are the marginal probability distribution functions of X and Y , respectively.

Mutual information tells us how the knowledge of one variable reduces uncertainty about the other variable. As a measure of dependency, it has several interesting properties:

- It equals 0 if and only if the two variables are independent.
- $I(X, X)$ is the same as the entropy of X (uncertainty contained in X).
- It is symmetric: $I(X, Y) = I(Y, X)$.

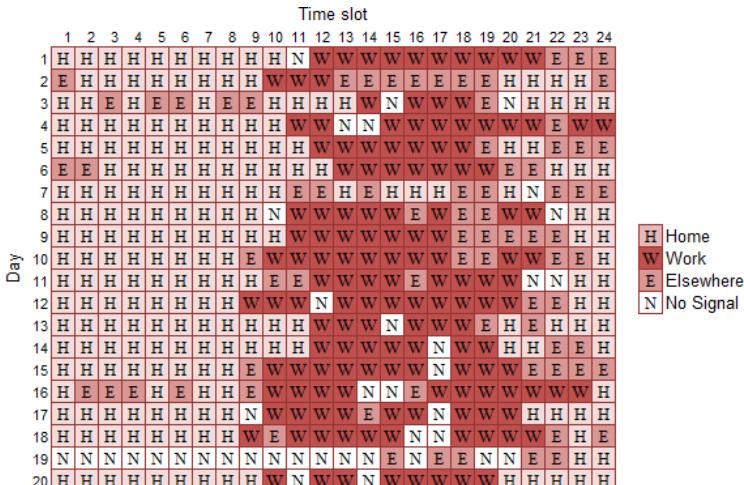


Fig. 1. Sample of a user's days

- It is positive and increasing with dependency when applied to probability distributions.
- It is independent of the actual value of the probability.

In this paper, we use a standard dataset collected during the Reality Mining project [11]. It consists of 330,000 hours of continuous data recorded over the course of nine months on 94 cellphones during the academic year 2004-2005. Participants, staff and faculty from the MIT Media Laboratory and Sloan Business School, were provided with Nokia 6600 Bluetooth-enabled cellphones running the ContextLogger. This piece of software, developed by the Context Group at the University of Helsinki [14] records different bits of information including call logs, Bluetooth devices in proximity, cell towers, application usage and phone status e.g. charging or idle. An anonymised version of the dataset collected in the Reality Mining project is the largest set of cellphone data publicly available to date and has been used extensively for activity inference.

Figure 1 shows some consecutive days of one user in the dataset. Each day is divided into 24 time slots of one hour each. For each time slot, we determined the user’s activity by computing the Cell ID of the antenna to which he was connected the longest. This Cell ID was then mapped to a one-letter symbol. H means that the user was at ‘Home’, W means that the user was at ‘Work’, E stands for ‘Elsewhere’, while N means that ‘No Signal’ was received during the time slot.

We then computed the mutual information between any two time slots in a user’s day (Figure 2). For most hours, strong short-range dependencies can

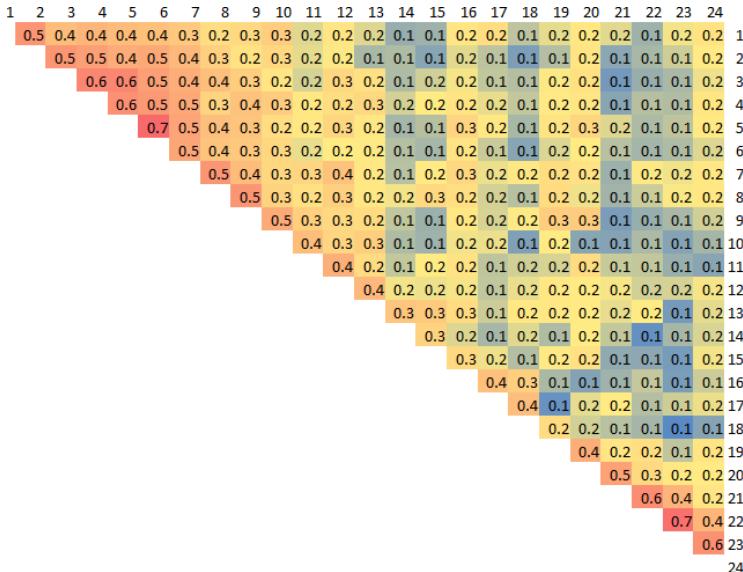


Fig. 2. Mutual information between time slots in a user’s day

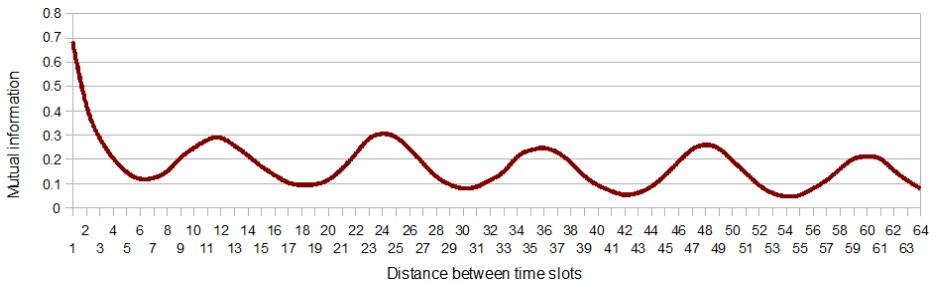


Fig. 3. Average mutual information against distance between time slots

be observed near the diagonal of the matrix. These short-range dependencies decrease around the middle of the day. In addition, certain pairs of distant time slots also exhibit high dependencies. For example, what the user does around 7 or 8 pm is significantly related to what he did in the morning around 9 am. This corresponds to the commuting routine mentioned earlier. Also, what the user will do around 6 am depends on what he was doing at 2 am. Indeed, if the user was sleeping at home at 2 am, he might go to work early in the morning. However, if he went out at night, he is unlikely to wake up early. In addition, as someone usually does not change location while sleeping, the mutual information between any two night time slots is high.

In order to highlight long-range dependencies, we plot in Figure 3 the graph of the average mutual information between two time slots against the distance between them. The importance of longer-range dependencies relative to shorter-range dependencies is striking. On average, two time slots spaced out of twelve hours have higher mutual information than two time slots distant of only four hours. We observe peaks of average mutual information between time slots spaced out of multiples of 12 hours (the values of these peaks decrease slowly with time). The twelve-hour pseudo-period corresponds to a secondary peak in the frequency domain (Figure 4).

In order to capture long-range dependencies between time slots, we designed and discuss in the following section an activity inference technique which aligns day-long sequences of activities.

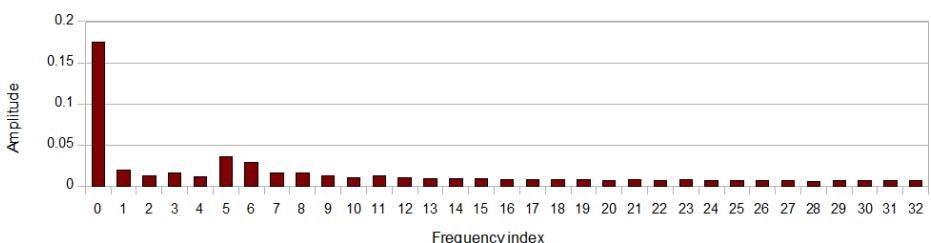


Fig. 4. Single-sided amplitude spectrum of the discrete Fourier transform

4 Aligning Activity Sequences

Alignment techniques stem from biology where they are used to compare sequences of residues. Identifying similarities between sequences often allows biologists to transfer structural or functional information from one sequence to another and determine evolutionary relationships.

4.1 Global Pairwise Alignment and Needleman-Wunsch

Pairwise alignment is '*the problem of deciding if a pair of sequences are evolutionary related or not*' [15]. Two DNA or RNA sequences which have evolved from a common ancestor usually exhibit significant similarity. However, throughout the evolutionary process, the two sequences will have accumulated certain mutations. The basic mutational process involves three types of operations. *Substitutions* change residues into other residues. *Insertions* and *deletions* respectively add or remove residues.

Because of potential insertions and deletions, biologists typically align sequences before comparing their residues. Most alignment techniques find an optimal alignment between two sequences under some additive scoring scheme e.g. +1 for a match between two residues and +0 for a mismatch.

Alignment techniques can be classified in two categories. Global alignment forces the alignment to span the entire length of the two sequences while local alignment focuses on regions of similarity within longer sequences which may differ significantly overall. In biology, local alignment is often preferred to global alignment but it can be more computationally expensive because it requires to find regions of similarity.

The dynamic programming algorithm for solving the global pairwise alignment problem is due to Needleman and Wunsch [16] and was later extended by Gotoh [17]. An optimal alignment of two sequences is built up recursively using previously computed optimal alignments for subsequences.

Let $x = x_1x_2\dots x_m$ and $y = y_1y_2\dots y_n$ be two sequences of symbols. The algorithm constructs an $(m+1) \times (n+1)$ matrix F . Scores for aligned symbols are specified by another matrix called the *scoring matrix* S . $S(x_i, y_j)$ is the score obtained by aligning symbols x_i to y_j . The algorithm uses a linear gap penalty γ . $\forall i \in \{1, 2, \dots, m\}, \forall j \in \{1, 2, \dots, n\}, F[i, j]$ is the score of the optimal alignment between prefix sequences $x_1x_2\dots x_i$ and $y_1y_2\dots y_j$, allowing gaps. The algorithm has three steps. In the initialisation step, the first row and column of F are filled using the formulae:

$$\begin{aligned} F[0, 0] &= 0, \\ F[i, 0] &= \gamma * i, \forall i \in \{1, 2, \dots, m\}, \\ F[0, j] &= \gamma * j, \forall j \in \{1, 2, \dots, n\}. \end{aligned}$$

$F[i, 0]$ represents the score obtained by aligning $x_1x_2\dots x_i$ with a sequence of i gaps whereas $F[0, j]$ represents the score obtained by aligning $y_1y_2\dots y_j$ with a sequence of j gaps.

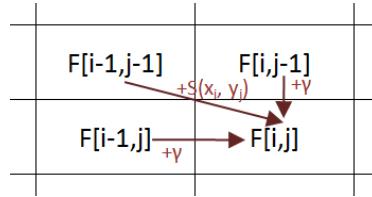


Fig. 5. The three alignment options

In the recursion step, the matrix F is filled from top left to bottom right using one of three cells $F[i - 1, j - 1]$, $F[i - 1, j]$ and $F[i, j - 1]$ to calculate $F[i, j]$:

$$F[i, j] = \max(F[i - 1, j - 1] + S(x_i, y_j), F[i - 1, j] + \gamma, F[i, j - 1] + \gamma).$$

The previous equation expresses the fact that the best score $F[i, j]$ of an alignment between $x_1 x_2 \dots x_i$ and $y_1 y_2 \dots y_j$ is the best of three options. First, x_i may be aligned to y_j , in which case $F[i, j] = F[i - 1, j - 1] + S(x_i, y_j)$. Secondly, x_i may be aligned to a gap, in which case $F[i, j] = F[i - 1, j] + \gamma$ and symmetrically, y_j may be aligned to a gap, in which case $F[i, j] = F[i, j - 1] + \gamma$ (Figure 5). As $F[i, j]$ is computed, a pointer is kept back to the cell which was used in its computation (above, left, or above-left cell).

By construction, $F[m, n]$ is the best score for an alignment between the full sequences x and y , allowing gaps. In order to retrieve an optimal alignment, a traceback step is performed. An optimal alignment is built in reverse from $F[m, n]$ to $F[0, 0]$, by following the pointers stored during the recursion step. Figure 6 shows an example of run and alignment (the symbol $-$ stands for a gap). It should be noted that several equal-scoring optimal alignments may exist. The traceback procedure can easily be modified to return all of them.

Needleman-Wunsch works because the optimal score up to a certain point in the alignment equals the optimal score up to the previous point in the alignment, plus the optimal score of the current step. F is an $(m + 1) \times (n + 1)$ matrix in which each element requires a bounded number of operations (three sums and a max in the recursion step). Therefore, the algorithm has a running time of $O(m \cdot n)$ and uses an amount of memory of $O(m \cdot n)$. (A modified version of the algorithm has a spatial complexity of $O(m + n)$, at the cost of a higher running time.) In the following, we consider sequences of equal length $m = n = 24$ having one symbol for each hour in a user's day.

4.2 Activity Inference as a Sequence Alignment Problem

It has been consistently observed that many human days were slight variations around a smaller number of routines [12], [18], [19]. Figure 7 shows two variations of the same daily routine, expressed as sequences of activity symbols. Each symbol represents the main activity of a person during a one-hour time slot. H, W, E, N stand for 'Home', 'Work', 'Elsewhere' and 'No Signal', respectively. Relative to the first sequence, the second sequence includes a time shift

	H	W	W	W	W	E
H	0	0	0	0	0	0
W	0	1	2	1	1	1
W	0	1	2	3	2	2
W	0	1	2	3	3	3
E	0	1	2	3	4	5
E	0	1	2	3	4	5

Alignment:

H	-	W	W	W	E	E
H	W	W	W	W	E	-

Fig. 6. Example of F matrix and optimal alignment

– ‘going to Work one hour later’ – and distortions such as ‘having a one-hour shorter lunch Elsewhere’. As in the case of biological sequences, similarity between activity sequences can be quantified in terms of substitutions, insertions and deletions. In the following, we use Needleman-Wunsch to align sequences of activities and compute an alignment score. However, this alignment score is not used for comparison as in the case of biological sequences, but for inference.

4.3 Alignment

In general terms, we consider a *test sequence* x of activities in which certain time slots are missing. Our objective is to infer the activity of the user during the missing time slots using a history of his past activities, formatted as a set \mathcal{H} of equal-length *training sequences*. To this end, we align the test sequence with different training sequences and compute the corresponding optimal alignment scores.

Let $x = x_1 x_2 \dots x_n$ be a test sequence defined over an alphabet of activity symbols $\Sigma = \{H, W, E, N\}$. Let $y = y_1 y_2 \dots y_n \in \mathcal{H}$ be a training sequence defined over $\Sigma \cup \{\text{?}\}$. ? is the *inference symbol* which stands for a time slot over which the user’s activity is to be inferred. Our objective is to substitute inference symbols with symbols corresponding to the user’s true activities. In order to do this, we align the test sequence with the training sequence and substitute inference symbols with aligned activity symbols in the training sequence. Intuitively, it is expected that alignments with a higher score are more likely to align inference symbols with the user’s true activities. Indeed, a high similarity score indicates that the two sequences are highly similar on non-inference symbols allowing time shifts and distortions. An example of inference is given on Figure 8.

Sequence 1: HHHHHHHHHNWWWEWWWWWWNHHHH
Sequence 2: HHHHHHHHHNWWWEWWWWWWNHHHH

Fig. 7. Two variations around the same routine

Test sequence:	HHHHHHHHHH????????WWWE-EHH
Training sequence:	HHHHHHHHHHHHWWWWWWWW-EEEHH
Inferred sequence:	HHWWWWWW

Fig. 8. Example of inference

In biology, one interesting property of alignment methods is their ability to handle insertions and deletions. This also makes alignment techniques particularly suited to sequences of human activities, which typically exhibit time shifts and distortions. In the example above, one gap was inserted in the target sequence to account for a longer stay Elsewhere and one gap was inserted in the training sequence because the user left Work earlier than in the test sequence. Sequence alignment guarantees that inferred sequences are observed in full in at least one training sequence. Therefore, the order of inferred activities and their durations are always plausible. Last but not least, only a few labelled sequences are required to align test sequences successfully (see Section 5). Collecting continuous data is difficult for the user. It is therefore essential to be able to infer activities from few training examples.

4.4 Activity Inference

As mentioned in Section 4.2, people typically have important daily routines. We can therefore perform global pairwise alignment on daily sequences. In the following, we make sequences start at midnight and end at midnight the next day. Because of the biological need to sleep at night, these boundaries are relatively stable in users' days.

Our activity inference scheme is built around an instance-based learner. Instead of estimating a target function once for the entire dataset, instance-based learners store training examples and compare each new instance with previously stored examples to make inferences. In our case, instance-based learning allows our scheme to make inferences using few training examples, which are slightly modified (by symbol insertions, deletions and substitutions) for comparison during alignment. By application of the Needleman-Wunsch algorithm, each training sequence is aligned with the test sequence to make an inference and compute a corresponding optimal alignment score. A nearest-neighbour approach then selects the inference provided by one alignment which achieves the highest score.

4.5 Learning Scoring Parameters

We learn scoring parameters using the history \mathcal{H} of training sequences. For each pair of activity symbols, we determine scoring terms from a probabilistic model. In the following, we restrict ourselves to ungapped global pairwise alignment. Given a pair of aligned sequences, our objective is to assign a score to the alignment that gives a measure of the relative likelihood that the sequences agree on inference points as opposed to disagreeing on inference points. Following the

work presented in [15] for sequence comparison, we build a Bayesian framework to perform inference. Specifically, we consider two hypotheses that assign a probability of the alignment in each of the two cases and compute the ratio of the two probabilities.

Let us consider an *agreement hypothesis* A and a *disagreement hypothesis* \bar{A} according to which aligned pairs of activity symbols (ω_1, ω_2) occur with a joint probability $p_{\omega_1 \omega_2}$, and $\bar{p}_{\omega_1 \omega_2}$, respectively. Making the simplifying assumption that the alignment of a pair of activity symbols is independent of the alignment of other pairs of activities, we have:

$$P(x, y|A) = \prod_{i=1}^n p_{x_i, y_i} \text{ and } P(x, y|\bar{A}) = \prod_{i=1}^n \bar{p}_{x_i, y_i}.$$

One way to measure the odds of an event given a hypothesis of another is to measure the odds ratio:

$$\frac{P(x, y|A)}{P(x, y|\bar{A})} = \prod_{i=1}^n \frac{p_{x_i, y_i}}{\bar{p}_{x_i, y_i}}.$$

The odds ratio is a measure often used in Bayesian statistics. It is a sample-based estimate of the ratio of the odds of an event occurring in one group to the odds of it occurring in another group. In our case, we estimate the ratio of the odds of two sequences being aligned given A over the odds of the same event given \bar{A} . An odds ratio greater than 1 indicates that the alignment under consideration is more likely if the two sequences agree on inference points. An odds ratio of 1 implies that the likelihood is the same in both cases whereas an odds ratio less than 1 means that the alignment is less likely if the sequences agree on inference points.

An additive scoring scheme can be derived from the odds ratio by taking its logarithm (log-odds ratio):

$$\text{Score} = \sum_{i=1}^n S(x_i, y_i),$$

where

$$S(\omega_1, \omega_2) = \log \left(\frac{p_{\omega_1 \omega_2}}{\bar{p}_{\omega_1 \omega_2}} \right)$$

is the log-likelihood ratio of the activity pair (ω_1, ω_2) occurring as an aligned pair, as opposed to an unaligned pair.

The log-odds score of a full alignment equals to:

$$\text{Score} = \log \left(\frac{P(x, y|A)}{P(x, y|\bar{A})} \right).$$

We are actually interested in determining whether the two sequences x and y agree or disagree on inference points. Using Bayes' rule, we can derive:

$$P(A|x, y) = \frac{P(x, y|A)P(A)}{P(x, y)}$$

$$\begin{aligned}
&= \frac{P(x, y|A)P(A)}{P(x, y|A)P(A) + P(x, y|\bar{A})P(\bar{A})} \\
&= \frac{1}{1 + P(x, y|\bar{A})P(\bar{A})/(P(x, y|A)P(A))} \\
&= \sigma(\text{Score}') ,
\end{aligned}$$

where

$$\sigma(t) = \frac{1}{1 + e^{-t}}$$

is the well-known logistic function often used to convert sums into probabilities (e.g. in Artificial Neural Networks) and

$$\begin{aligned}
\text{Score}' &= \log \left(\frac{P(x, y|A)P(A)}{P(x, y|\bar{A})P(\bar{A})} \right) \\
&= \log \left(\frac{P(x, y|A)}{P(x, y|\bar{A})} \right) + \log \left(\frac{P(A)}{P(\bar{A})} \right) \\
&= \text{Score} + \log \left(\frac{P(A)}{P(\bar{A})} \right) \\
&= \text{Score} + \log \left(\frac{P(A)}{1 - P(A)} \right) \\
&= \text{Score} - \log \left(\frac{1}{P(A)} - 1 \right) .
\end{aligned}$$

The estimation of $p_{\omega_1\omega_2}$ and $\bar{p}_{\omega_1\omega_2}$ is performed in an unsupervised fashion from \mathcal{H} . For each pair of sequences $(y, z) \in \mathcal{H}^2$, $y = y_1y_2\dots y_n$, $z = z_1z_2\dots z_n$, $y \neq z$ are first aligned using the *identity alignment* i.e. $\forall k \in \{1, 2, \dots, n\}$, y_k is aligned with z_k . We then compute the accuracy $\text{acc}(x, y)$ of the predictions provided by the identity alignment and divide pairs of aligned sequences into two clusters according to a threshold parameter α . The agreement cluster \mathcal{A} contains all alignments of sequences (x, y) for which $\text{acc}(x, y) \geq \alpha$ whereas the disagreement cluster $\bar{\mathcal{A}}$ contains all alignments of sequences for which $\text{acc}(x, y) < \alpha$.

For each pair of activity symbols (ω_1, ω_2) , $p_{\omega_1\omega_2}$ and $\bar{p}_{\omega_1\omega_2}$ are estimated by computing the frequency at which ω_1 and ω_2 are aligned at non-inference points in \mathcal{A} and $\bar{\mathcal{A}}$, respectively. We use pseudo-counts to handle cases in which a given pair of aligned activity symbols does not occur in a cluster. Eventually, the prior probability $P(A)$ is estimated as $|\mathcal{A}|/(|\mathcal{A}| + |\bar{\mathcal{A}}|)$.

4.6 Preprocessing Activity Sequences

Needleman-Wunsch makes the simplifying assumption that the alignment of two activity symbols is independent of the alignment of other symbols. While this assumption may be justified in certain biological applications, subsequent activities are often related to each other. In order to take this into account in our model we define a preprocessing filter $\varphi_{l,r}$ as a function which transforms a sequence over $\Sigma \cup \{?\}$ into a sequence defined over the alphabet $\Sigma^{l+r+1} \cup$

Table 1. Some examples of filtering

x^{prev}	x	x^{next}	$ l $	$ r $	$\varphi_{l,r}(x)$
EHHEEE	HHHEEE	HEEEHH	0	1	HH HH HE EE EE EH
EHHEEE	HHHEEE	HEEEHH	2	0	EEH EHH HHH HHE HEE EEE
EHHEEE	HH??EE	HEEEHH	1	1	EHH HHH ? ? EEE EEH

{?} by replacing each symbol x_i with the $(l + r + 1)$ -dimensional vector $x'_i = x_{i-l}x_{i-l+1}\dots x_{i-1}x_ix_{i+1}\dots x_{i+r-1}x_{i+r}$. Intuitively, to each symbol x_i in x , we concatenate l symbols to the left and r symbols to the right. Concatenated symbols are taken in x except where their indices are lower than 1 or greater than n , in which case they are taken in the previous activity sequence x^{prev} or in the next x^{next} , respectively. Concatenated inference symbols are replaced with the nearest non-inference symbol in x . Table 1 shows three examples of output sequences using different parameters (for ease of illustration, we consider sequences of length 6).

5 Experiments and Results

Our technique was evaluated on real data extracted from the Reality Mining dataset. We considered 121 consecutive days of 5 subjects. Each day was divided in 24 one-hour time slots providing 14520 data points in total. The time range that we considered in the dataset started on 26th August 2004 and ended on 24th December 2004. Users and time periods were selected to provide a large number of valid records. Days were considered once for testing using a predefined number of immediately preceding days for training. For comparison, we computed the *most-frequent-sequence* baseline which predicts the sequence of missing activities as being the most frequently observed sequence of activities at inference points in training data. In the following, we represent inference and non-inference points using a *gap pattern*, defined as a sequence of length n in which ? stands for an inference symbol in the test sequence and * stands for any other symbol.

The similarity matrix presented in Table 2 was learnt over a training period of $\tau = 8$ weeks of data for the gap pattern *****??????*****. The gap penalty γ was empirically set to -2.0 .

Collecting continuous data for an extended period of time can be very inconvenient for the user. We therefore examined the impact of the length of the

Table 2. Example of scoring matrix

	H	W	E	N
H	0.60	0.29	0.01	-1.42
W	0.29	0.43	0.03	-2.79
E	0.01	0.03	0.12	-2.18
N	-1.42	-2.79	-2.18	1.49

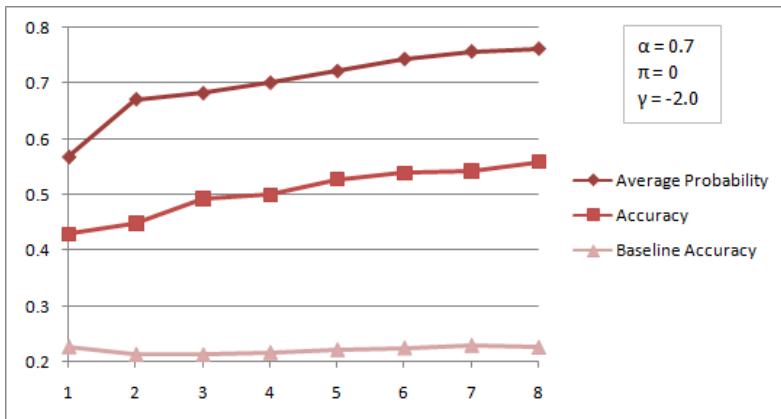


Fig. 9. Accuracy and conditional probability versus τ in weeks

training period τ on accuracy results. Figure 9 compares accuracy results for different training periods in weeks. With only two weeks of training, an accuracy of 43.1% was observed. The performance of the scheme then increases linearly with the length of the training period and always very significantly outperforms the baseline. The accuracy figures presented here must be considered carefully. Indeed, these are computed over symbol inferences. Therefore, a slight variation such as a one-hour shift in the inferred sequence relative to the true sequence of activities can affect the measure drastically. Even with this limitation, the scheme achieves almost 50% accuracy after three weeks of training. Actually, many applications may not need such fine-grained inferences over missing data. Indeed, as the device is often off when data is missing, predictions will not be used to take real-time actions. As the number of training sequences considered

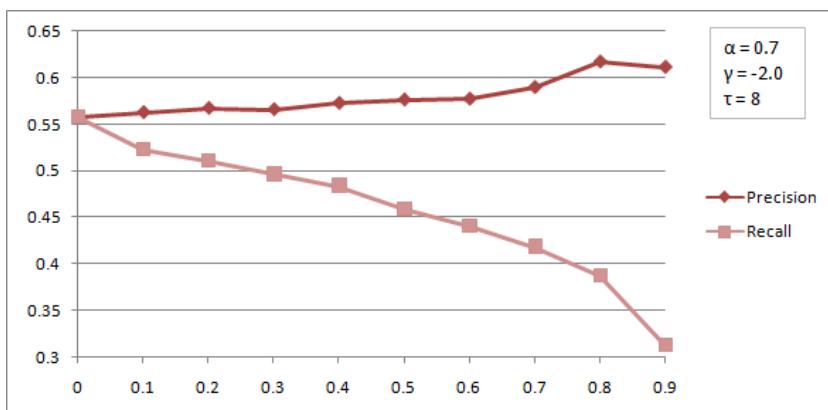


Fig. 10. Precision and recall versus π

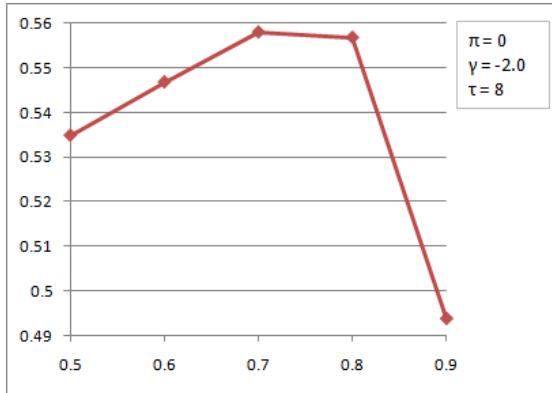


Fig. 11. Influence of α on accuracy

increases, the conditional probability of A given the optimal alignment found also increases.

This conditional probability indicates the reliability of an output. By providing no output under a certain probability threshold π , it is possible to improve the precision of the technique, at the cost of a lower recall (Figure 10).

The threshold α which separates alignments in \mathcal{A} from alignments in $\overline{\mathcal{A}}$ can be adjusted to improve accuracy. Figure 11 shows that a high value of α yields to better results. However, beyond a certain value $\alpha_0 \in (0.7, 0.8]$, the cluster \mathcal{A} does not contain enough alignments and performance decreases.

We tested the method over a variety of gap patterns. Figure 12 shows the decrease in accuracy as the length of a central gap increases. The graph is made of two linear segments with a first segment corresponding to a rapid decrease from two to six consecutive hours of missing data. In Figure 13, we are interested in

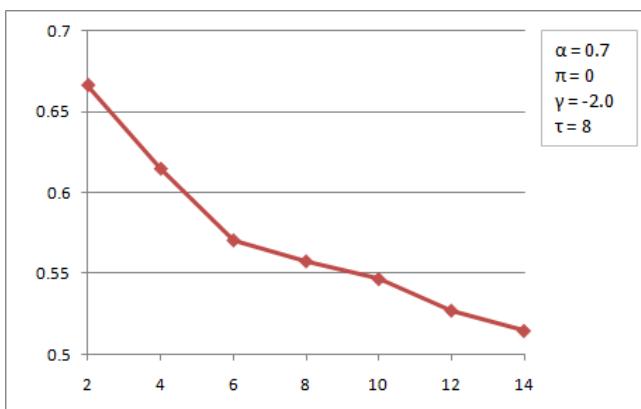


Fig. 12. Accuracy versus length of a central gap

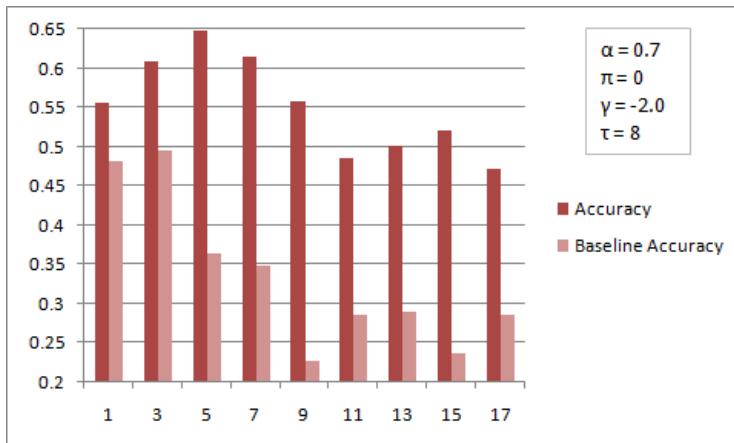


Fig. 13. Accuracy versus position of an 8-hour gap

the influence of the position of an 8-hour gap on accuracy. The x-axis is the index at which the gap begins. It can be observed that morning hours are much more predictable given the rest of the day than evening hours. Improvement over the baseline is lowest for early missing intervals. Indeed, activities in early time slots (from midnight to 8 am) are very stable, which favours the baseline approach. At the other extremity, improvement over the baseline is very significant on midday gaps for which data is available on both sides of the gap interval.

We investigated the performance of the scheme when available data was dispatched throughout the day. Figure 14 shows the performance of the scheme when half the day is missing overall but equal-length available and missing

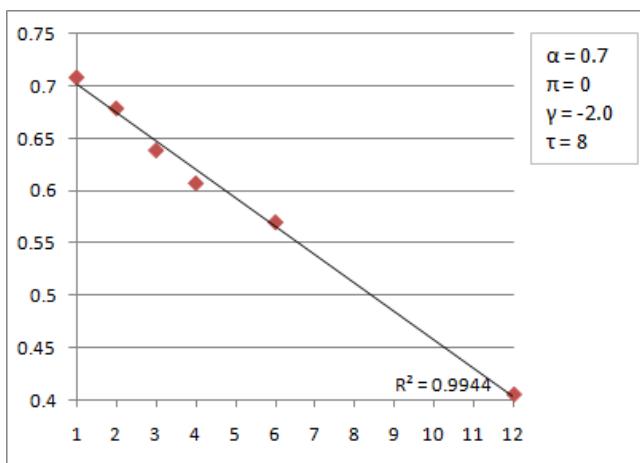


Fig. 14. Accuracy versus length of alternating missing and available intervals



Fig. 15. Improvements using the preprocessing filter

intervals alternate. The length of missing and available intervals is represented on the x-axis. The scheme provides excellent results when intervals are short and performance decreases linearly with their length.

Applying the preprocessing filter presented in Section 4.6 can further improve the accuracy of the technique. Figure 15 shows the improvements obtained for a centred twelve-hour gap. The parameters $(l, r) = (2, 0)$ provided the best results. Results for (l, r) such that $l + r > 2$ gave poor results. One explanation for this may be that the number of symbols considered in the alphabet increases exponentially with $l + r$. Training data may therefore not be sufficient to learn reliable scoring terms when $l + r > 2$.

6 Conclusion and Future Work

In this paper, we have quantified short- and long-range dependencies in human behaviour and presented a novel approach to activity inference which makes use of day-long sequences of data. Our biology-inspired technique fills in gaps in activity logs by aligning sequences through the Needleman-Wunsch dynamic programming algorithm using scoring parameters learnt from training data in a probabilistic framework. Evaluation was performed on a standard dataset collected on mobile phones. With eight weeks of training, user activities can be inferred with over 70% accuracy when every other hour is missing in the day. Also, an accuracy of almost 50% was obtained with as little as three weeks of training and eight consecutive hours missing in the day.

Filling gaps in data is a recurrent problem in pervasive computing. Alignment methods have major advantages to perform this task. They take into account large windows of context, provide output sequences which are plausible, are robust to time shifts and distortions and require little training data.

We envision three lines of future work. First, other elements of context could be integrated in aligned sequences, for example fine-grained locations, co-presence and secondary activities. Secondly, the gap penalty is currently determined empirically. Parametric alignment techniques could help find the regions of the parameter space which provide the best accuracy results on the training set. Last but not least, we are currently working on learning optimal parameters to perform activity sequence alignment using pair and profile HMMs.

Acknowledgment

This research was undertaken as part of the Cityware project (www.cityware.org.uk) and supported by UK EPSRC research grant EP/C547705/1. We would also like to thank Nathan Eagle for his kind help with the Reality Mining dataset.

References

1. Dornbush, S., English, J., Oates, T., Segall, Z., Joshi, A.: XPod: A Human Activity Aware Learning Mobile Music Player. In: Proceedings of the Workshop on Ambient Intelligence, 20th International Joint Conference on Artificial Intelligence (IJCAI 2007) (January 2007)
2. Park, H.S., Yoo, J.O., Cho, S.B.: A context-aware music recommendation system using fuzzy bayesian networks with utility theory. In: Wang, L., Jiao, L., Shi, G., Li, X., Liu, J. (eds.) FSKD 2006. LNCS(LNAI), vol. 4223, pp. 970–979. Springer, Heidelberg (2006)
3. Stiefmeier, T., Roggen, D., Ogris, G., Lukowicz, P., Tröster, G.: Wearable activity tracking in car manufacturing. *IEEE Pervasive Computing* 7(2), 42–50 (2008)
4. Ward, J.A., Lukowicz, P., Tröster, G., Starner, T.: Activity recognition of assembly tasks using body-worn microphones and accelerometers. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28, 1553–1567 (2006)
5. Antifakos, S., Michahelles, F., Schiele, B.: Proactive instructions for furniture assembly. In: Borriello, G., Holmquist, L.E. (eds.) UbiComp 2002. LNCS, vol. 2498, pp. 351–360. Springer, Heidelberg (2002)
6. Favela, J., Tentori, M., Castro, L.A., Gonzalez, V.M., Moran, E.B., Martínez-García, A.I.: Activity recognition for context-aware hospital applications: issues and opportunities for the deployment of pervasive networks. *Mob. Netw. Appl.* 12(2-3), 155–171 (2007)
7. Sánchez, D., Tentori, M., Favela, J.: Activity recognition for the smart hospital. *IEEE Intelligent Systems* 23(2), 50–57 (2008)
8. Lin, W., Sun, M.T., Poovandran, R., Zhang, Z.: Human activity recognition for video surveillance. In: IEEE International Symposium on Circuits and Systems, ISCAS 2008, May 2008, pp. 2737–2740 (2008)
9. McKenna, T.: Video surveillance and human activity recognition for anti-terrorism and force protection. In: AVSS 2003: Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance, Washington, DC, USA, p. 2. IEEE Computer Society, Los Alamitos (2003)
10. Minnen, D., Westeyn, T., Ashbrook, D., Presti, P., Starner, T.: Recognizing soldier activities in the field. In: Body Sensor Networks, Aachen, Germany (2007)
11. Eagle, N., Pentland, A.S.: Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing* 10(4), 255–268 (2006)
12. Eagle, N., Pentland, A.: Eigenbehaviors: Identifying structure in routine. *Proc. Roy. Soc. A* (2006) (in submission)
13. Shannon, C.E., Weaver, W.: The Mathematical Theory of Communication. University of Illinois Press, Urbana (1948)
14. Raento, M., Oulasvirta, A., Petit, R., Toivonen, H.: Contextphone: A prototyping platform for context-aware mobile applications. *IEEE Pervasive Computing* 4(2), 51–59 (2005)

15. Durbin, R., Eddy, S.R., Krogh, A., Mitchison, G.: Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press, Cambridge (1999)
16. Needleman, S.B., Wunsch, C.D.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* 48(3), 443–453 (1970)
17. Gotoh, O.: An improved algorithm for matching biological sequences. *J. Mol. Biol.* 162(3), 705–708 (1982)
18. Farrahi, K., Gatica-Perez, D.: Daily routine classification from mobile phone data. In: Popescu-Belis, A., Stiefelhagen, R. (eds.) MLMI 2008. LNCS, vol. 5237, pp. 173–184. Springer, Heidelberg (2008)
19. Farrahi, K., Gatica-Perez, D.: What did you do today?: discovering daily routines from large-scale mobile data. In: MM 2008: Proceeding of the 16th ACM international conference on Multimedia, pp. 849–852. ACM, New York (2008)

Improving Location Fingerprinting through Motion Detection and Asynchronous Interval Labeling

Philipp Bolliger¹, Kurt Partridge², Maurice Chu², and Marc Langheinrich³

¹ Institute for Pervasive Computing, ETH Zurich, Switzerland

² Palo Alto Research Center, Palo Alto, CA, USA

³ Faculty of Informatics, University of Lugano, Switzerland

Abstract. Wireless signal strength fingerprinting has become an increasingly popular technique for realizing indoor localization systems using existing WiFi infrastructures. However, these systems typically require a time-consuming and costly training phase to build the radio map. Moreover, since radio signals change and fluctuate over time, map maintenance requires continuous re-calibration. We introduce a new concept called “asynchronous interval labeling” that addresses these problems in the context of user-generated place labels. By using an accelerometer to detect whether a device is moving or stationary, the system can continuously and unobtrusively learn from all radio measurements during a stationary period, thus greatly increasing the number of available samples. Movement information also allows the system to improve the user experience by deferring labeling to a later, more suitable moment. Initial experiments with our system show considerable increases in data collected and improvements to inferred location likelihood, with negligible overhead reported by users.

1 Introduction

WiFi localization has shown great promise for indoor positioning, yet has not achieved ubiquitous commercial success yet. One difficulty has been the construction of an accurate mapping between signal strength patterns and physical locations. The signal strength patterns depend not only on the distances between WiFi radios, but also on other factors such as the positions of physical objects, which reflect or block signals. This complication may be partially overcome by either performing calculations with detailed models of the environment, or by collecting a dense dataset of fingerprints and their associated true locations [2]. In this paper, we focus on the latter approach, as it is generally more accurate and it is easier to collect this data.

Even so, collecting labeled fingerprint samples can be tedious. Signal readings must be collected every few meters or so, with pauses of tens of seconds at each position to get an accurate reading. This process must be repeated if the infrastructure or environment changes substantially. Commercial deployments usually

conduct such surveys as part of deployment, however in some installations, such as private homes, consumers may not have the patience for this process.

Previous work has explored end-user labeling of locations [1,4,8,5]. End-user labeling allows labels to be added as needed, in the places that users most frequently visit. However, second-by-second signal fluctuations mean that the fingerprint stored with a label may not match future measurements. Ideally, an end-user labeled fingerprint would also be collected over an interval of several tens of seconds, much as is done during formal calibration stages. Users, however, may not have the patience to comply with this restriction.

In this paper, we present PILS, an adaPtive Indoor Localization System that addresses the challenges of end-user labeling. PILS explores a technique that extends the applicability of a user-provided label from an instant to an interval over which the device is stationary. The stationary state is detected using an accelerometer, which allows PILS to detect location changes autonomously, and consequently collect stationary interval measurements without explicit user intervention.

Using intervals also enables a different kind of labeling. By detecting intervals of device immobility, the system can defer location labeling to a more appropriate time, and refer to longer time periods that are easy for users to remember (e.g., “Where were you between 9:15 am and 10:05 am today?”). This greatly improves the user experience, as users need not provide labels while at the labeled location, where they are likely engaged in some other activity. We call this technique *asynchronous interval labeling*.

The remainder of this paper is structured as follows. The next section explains how PILS relates to other indoor localization systems, in particular, those that follow a user-labeling approach. Section 3 then describes asynchronous interval labeling in detail. Sections 4 and 5, respectively, describe the prototype implementation of PILS and the results of the initial experiments. We close with discussion and conclusions in Sections 6 and 7.

2 Related Work

Research on indoor location systems has been popular for several years [2,10,17,25]. Location systems typically output Cartesian coordinates, which, for indoor settings, are often mapped to rooms based on available map data [17]. Like other systems [6,9], we output symbolic location data (such as room identifiers) directly. However, rather than using authoritative designations, we collect room labels directly from end-users during their use of the system.

WiFi fingerprinting [2] has been particularly popular for indoor positioning, because it requires no new hardware infrastructure for sites that already have WiFi. With resolution to a few meters, it can usually support room-level localization. To achieve such high accuracy from the noisy WiFi signal, however, such systems require prior manual calibration. For example, King et al. [17] were able to achieve an average error distance of less than 1.65 meters, but required prior calibration with 80 measurements every square meter (20 measurements each

at 0°, 90°, 180°, and 270° orientations). Even though a single active WiFi scan takes only 250 ms, the time needed to measure all four orientations and to move between locations quickly adds up to tens of seconds per reference point. In total, the training phase for a small 100 m² building could take well over one hour. In addition, the training may miss longer-term variations as described in Section 3. While training time can be reduced by modeling the environment [12], this approach is less accurate and requires additional information (such as floorplans) that are not always available or easy to input.

Ashbrook and Starner [1] describe how significant places may be determined by detecting stationary GPS signals, which can later be clustered and labeled. Froehlich et al. [8] also identify significant places using GSM signals. Both approaches identify places on a building-sized scale rather than a room-sized scale, and neither use an additional sensor such as an accelerometer to detect true motion stability.

Bhasker et al. previously explored collecting calibration data during use, rather than in a separate training step [4]. Their localization system employs a two stage process. First it computes geometric location. The result is shown on a map, and can be corrected if necessary. Corrections are treated as virtual access points and given higher priority when calculating locations. However, this method requires having a map and interrupting the user’s primary activity to collect input. The system also allows only one correction per location.

Our earlier work, Redpin [5], also collected calibration information from end-users. In contrast to Bhasker et al.’s geometric approach, Redpin generates symbolic location identifiers with room-level accuracy. If the current location cannot be predicted accurately, the user is prompted to enter a label for his or her current location. By allowing multiple measurements for the same location and by combining from GSM, WiFi, and Bluetooth, Redpin can provide the correct symbolic location in nine out of ten cases, as evaluated in a two-day experiment with ten users on one 30-room floor.

Other location systems also perform motion detection. Krumm and Horovitz’s LOCADIO [19] uses WiFi signal strength to both localize a device and infer whether it is moving. However, due to the natural fluctuation of signal strength readings even when standing still, this motion detection’s error rate is 12.6%, which results in a high number of false state transitions (e.g., from “stationary” to “moving”) during experimental use (24 reported when only 14 happened).

King and Kjærgaard [16] also use WiFi to detect device movement, reporting results similar to Krumm and Horovitz’s on a wider variety of hardware. They use motion data to minimize the impact of location scanning on concurrent communications: If the device is stationary, the system does not need to recompute its position (which might interfere with regular communications as both activities share the same WiFi card). In contrast, we use motion information not only for positioning, but also to aid the training: If the device is stationary, the system can collect stable WiFi measurements.

Accelerometer-based detection of stability has been a popular research topic in the context of activity recognition [3,15,20,22]. Kern [15] showed how to use

an accelerometer to distinguish a moving state (walking, running, or jumping) from a stationary state (standing or sitting). Since our use emphasizes the stability of the device, rather than the activity of the person, we do not require a worn device. Moreover, many of today’s laptops, PDAs, and mobile phones already contain accelerometers to protect system operations (by parking disk heads before possible impact) or to support the user interface (by detecting screen orientation).

3 Interval Labeling with PILS

The method of location fingerprinting using WiFi signals assumes that the pattern of *mean signal strengths* received in one location differs from the pattern observed in another location. Unfortunately, various effects, including interference from interposing static objects as well as reflections off neighboring objects, make the relationship between the signal means and location difficult to predict in practice [11,18,21,24]. Less well-documented are sources of variance in the signal, although there has been some work studying these effects over a one day period [13].

To understand the significance of those mean variations in signal strength, we performed a small experiment. We positioned five laptops in different rooms of our office building. For one week, each laptop did an active WiFi scan every minute and recorded the access points’ unique identifiers (BSSID) and received signal strengths (RSS). Figure 1(a) shows the signal strength variation for three laptops over the course of a day. Different lines correspond to the signal strengths from different access points. Rooms 2212 and 2214 are adjacent to each other, and Room 2152 is further away. Room 2212 and 2214’s patterns resemble each other much more than either of them do 2152, illustrating how these readings can be used to determine position. However, the graph also shows that there is short-term variation from minute-to-minute as well as longer-term fluctuations. The short-term fluctuations arise not only from the motion of people—average per-access point variance on low-traffic weekends was still 68% of the variance during the week. Additionally, different access points have different variances. Figure 1(b) shows the detail of the first hour, with individual scans now indicated by circles. This shows how readings can appear in scans at different rates independent of the mean received signal strengths.

The long-term variance, which is especially noticeable during the day in Room 2212, shows that for nearby locations it may not suffice to build the radio map only once. The best way to cope with long-term variance is to update the map frequently by taking measurements at different times of the day and days of the week. This addresses not only variations of unknown causes, but also infrastructure changes such as failing or replaced access points.

These signal traces also show that the best way to reduce the error caused by short-term signal variance is to average a large number of measurements taken during a short time. However, collecting measurements is tedious and not something an end-user is very eager to do. So, two challenges are: How can a

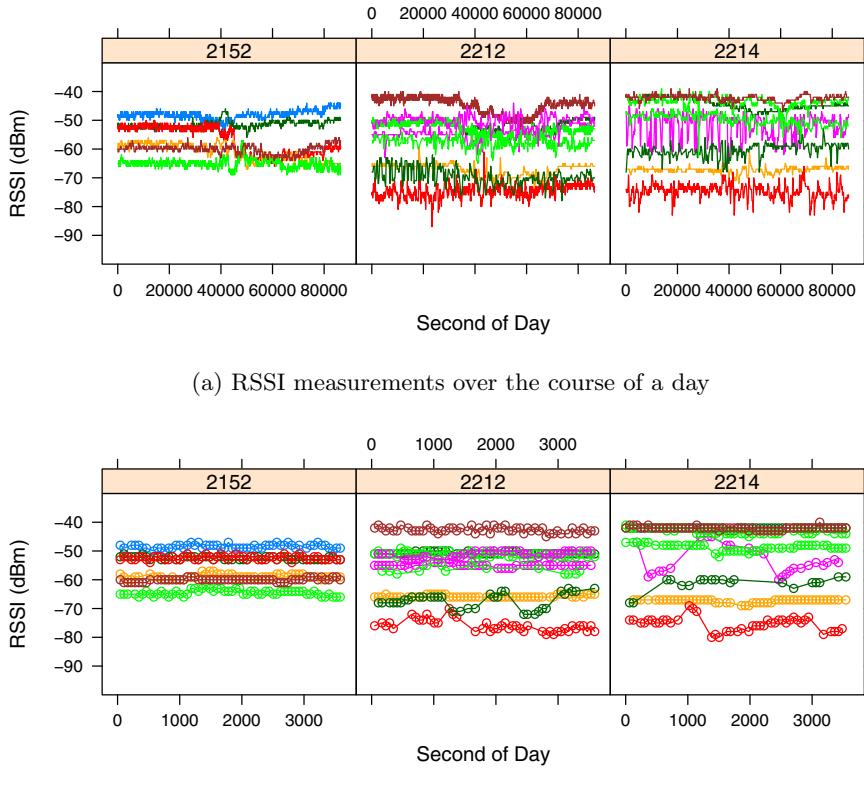


Fig. 1. Signal strength variations from three laptops. Rooms 2212 and 2214 are adjacent to each other, and Room 2152 is further away. Signal variations happen on different timescales, ranging from a few minutes to several hours.

system get users to contribute many labeled measurements to the system *without interrupting their work routine?* And how can a system continue to update the radio map over days and weeks, again *unobtrusively?*

Our method of interval labeling addresses these two challenges. Labels provided by end users are applied not only to the immediate signal strength measurement, but to all measurements taken during the interval while the device was stationary, at the same place. Figure 2 gives an example of the process of interval labeling. Using data from the accelerometer, PILS partitions time into alternating periods of “moving” and “stationary” as indicated in the second row of the figure. (The implementation of the motion detection process is described in Section 4.3.) Whenever the system is stationary, it continuously adds measurements to the interval. When it detects movement, it stops taking measurements until the device rests again, at which time a new interval begins.

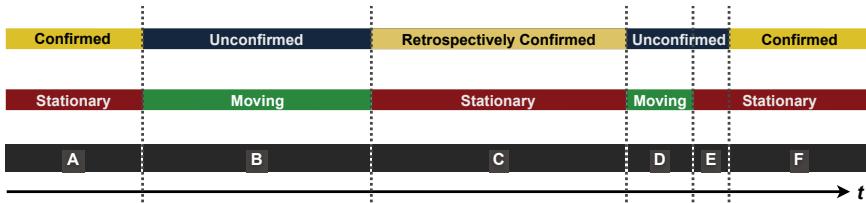


Fig. 2. Interval labeling allows the user to update the radio map with all data taken while the device is stationary. Because intervals provide more cues to users (starting time, ending time, and duration of interval), users are more likely to remember where they were during an interval than at an instant.

In addition to increasing the number of WiFi measurements that can be associated with a location label, intervals can improve the user experience of labeling. Because intervals are known to be periods of immobility, they can be more easily labeled asynchronously. A user is more likely to remember their location during the entire interval (knowing its starting time and duration) than they are likely to remember their location at a specific instant. This gives the system the freedom to postpone labeling until a more convenient time such as the start of the next stationary period, or when the user returns to their desk. This can help the system reduce the obtrusiveness of any explicit location prompts.

4 The PILS System

Figure 3 gives an overview of the three main system components: a *scanner* to listen for announce beacons, a *locator* to compare current measurements with the assembled radio map from a fingerprint database, and a *motion detector* to inform the locator about interval boundaries (i.e., changes between the moving state and stationary state).

4.1 Hardware

PILS requires a WiFi communications module and an accelerometer in the terminal—two components that are often available in today’s laptops and high-end mobile phones. We implemented our initial prototype on Mac OS X 10.5 using MacBook Pros (revision B and C).

In particular, the 15-inch machines that we used featured a WiFi network card from Atheros or Broadcom, as well as a Kionix KXM52-1050 three-axis accelerometer with a dynamic range of $+/- 2g$ and bandwidth up to 1.5KHz.

In our office environment, there were sixteen access points to cover about 70 rooms with a combined area of $1000m^2$. Access points were installed at a density of about 0.23 access points per room, or 1 access point per $62.5m^2$ of office area. Typically five access points were visible at any location.

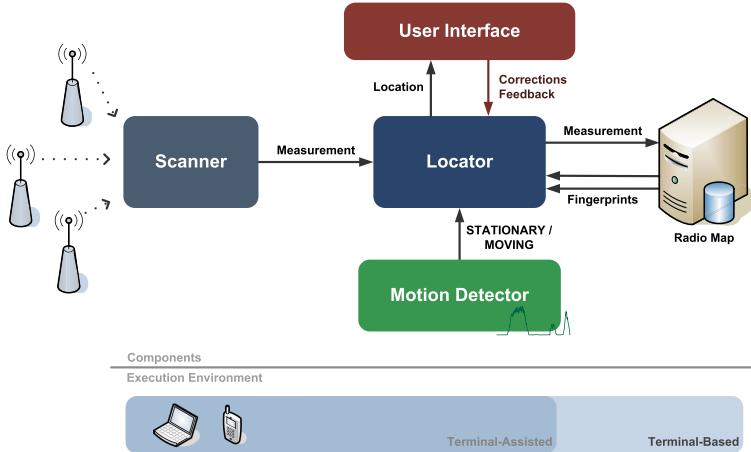


Fig. 3. Our terminal-based system has four components. The signals observed by the Scanner are sent to the Locator, which estimates the location from the radio map stored in the Radio Map. The Motion Detector informs the Locator whether the device is stationary or moving, and the User Interface collects the labels. For low-power devices, a terminal-assisted approach could outsource location estimation to a central server.

4.2 Locator

Our approach to location fingerprinting is to learn a probabilistic model of the likely readings of received signal strength (RSS) of WiFi beacons for each location we are interested in. With these learned models, we estimate the device’s location by choosing the model that gives the maximum likelihood. Our probabilistic model is similar to the approach taken by Chai and Yang [7], except that we use normal distributions for RSSI rather than quantizing RSSI values and using histograms. As long as the RSSI values are not multi-modal, such a unimodal approach still offers good performance while being computationally much simpler. By keeping only the mean and variance, updates are very fast and do not use much memory. In addition, the larger number of free parameters in a histogram approach is more susceptible to overfitting when there is not much data.

Each received signal strength reading is stored as a pair consisting of the access point’s BSSID and the measured indicator of its signal strength, i.e., $b_t = (BSSID_t, RSSI_t)$, with $RSSI_t$ being the received signal strength from the WiFi access point with unique identifier $BSSID_t$ at time t .

For each location l we learn a model of the readings received by a device in location l . For a set of n readings $\{b_1, \dots, b_n\}$ in location l , we adopt the following model for the likelihood of the set of readings:

$$P_l(b_1, \dots, b_n) = \prod_{i=1}^n p_l(BSSID_i) \cdot N(RSSI_i; \mu_l(BSSID_i), \sigma_l^2(BSSID_i)) \quad (1)$$

where N is the normal distribution and $p_l(BSSID)$ is the probability that the reading in location l comes from WiFi access point BSSID. We model each reading to be independently generated from a normal distribution with mean $\mu_l(BSSID_i)$ and variance $\sigma_l^2(BSSID_i)$, which can be different for each access point.

Given a set of n readings $\{b_1, \dots, b_n\}$ in location l , the model parameters which maximize the likelihood of the readings are given by:

$$\begin{aligned} p_l(bssid) &= \frac{R_{bssid}}{n} \\ \mu_l(bssid) &= \frac{1}{R_{bssid}} \sum_{i:BSSID_i=bssid} RSSI_i \\ \sigma_l^2(bssid) &= \frac{1}{R_{bssid} - 1} \sum_{i:BSSID_i=bssid} (RSSI_i - \mu_l(bssid))^2 \end{aligned}$$

where $R_{bssid} = |\{b_i | BSSID_i = bssid\}|$ is the number of readings that came from WiFi access point $bssid$. Note that a location l will not get readings from all access points. For those access points which were not part of the readings for learning the model, we set $p_l(bssid)$ to a very small value, e.g., 10^{-15} . The parameters $\mu_l(bssid)$ and $\sigma_l^2(bssid)$ can be chosen in any way as long as the product of p_l and the normal distribution is small.

To estimate the most likely location \hat{l} from a set of readings $\{b_1, \dots, b_n\}$, we can compute Eq. 1 and find the maximum likelihood location as follows:

$$\hat{l} = \arg_l \max P_l(b_1, \dots, b_n).$$

In practice, we compute $\log P_l(b_1, \dots, b_n)$ because it is numerically stable and the monotonic property of the logarithm guarantees the same answer for \hat{l} .

4.3 Motion Detector

The motion detector is a discrete classifier that reads the accelerometer to determine whether the device is being moved or whether it is stationary. Classification needs to be somewhat forgiving so minor movements and vibrations caused by readjusting the screen or resting the computer on one's lap are still classified as "stationary." Only significant motion such as walking or running should be classified as "moving."

To classify the device's motion state, the motion detector samples all three accelerometer axes at 5 Hz. It then calculates the acceleration magnitude and subtracts it from the previously sampled magnitude. To prevent misclassification of small movements as "moving," the signal is smoothed into a moving average of the last 20 values. Figure 4 shows that this method yields a sharp amplitude increase in the magnitude delta whenever the user is walking. The classifier includes hysteresis with different threshold values when switching between the moving and stationary states. The exact threshold values were established through a series of informal experiments. Figure 4 shows the motion magnitude trace of a user going

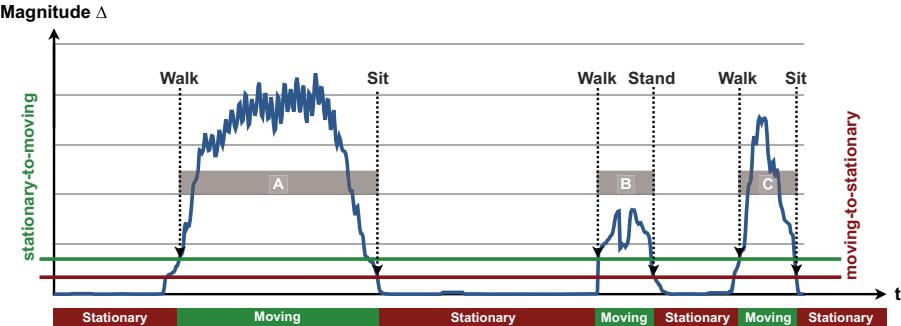


Fig. 4. Example data from the motion detector. As soon as the magnitude delta exceeds the stationary-to-moving threshold, the device is considered to be moving. This holds as long as the magnitude delta does not fall below the moving-to-stationary threshold.

from his office to a colleague’s office (A) and back (B and C), with two stationary phases in between: a longer discussion at the colleague’s office and a brief chat in the hallway. The sequence bar at the bottom of the figure shows the motion detector’s output. Due to the use of a moving average, the system imposes a small delay of 2-4 seconds before values fall below the threshold for the stationary state, however this does not appear to degrade performance.

5 Evaluation

To get a better sense of whether interval labeling would work well in practice, we conducted a user study. The study examined whether users would voluntarily correct incorrect location predictions, what the characteristics of the labeled intervals were, and whether labeling increased the system’s confidence in the user’s location.

5.1 Experimental Setup

We recruited 14 participants, all researchers at one of our institutions. Participants installed a custom application on their MacBooks. The software placed an extra menu in the right side of the menu bar, as shown in Figure 5. Users were instructed to correct the system if they saw that it incorrectly guessed the location. This was also the mechanism for adding new labels to the system. The users gained no benefit from the application other than the satisfaction of making the correction. The study ran for five weeks, which included the winter holiday period.

To remind users about the study and to provide additional feedback to the user about the system’s inferences, the user could optionally enable a voice announcement of “moving” and “stationary” when the device transitioned between moving and stationary states. Music could also optionally be played while the device was in the moving state. However, as the laptops went to sleep when their lids were closed, the music typically did not continue for the entire moving duration.



(a) The user corrects an erroneous inference through the “Correct My Location...” menu option.

(b) The user can enter any label for the current location by a simple dialog.

Fig. 5. User interface for collecting label corrections: The system’s prediction of the room is placed in the menu bar to provide ambient awareness

Location inferences were made on the users’ laptops, however all WiFi measurements and labeled data were uploaded to a server for later analysis.

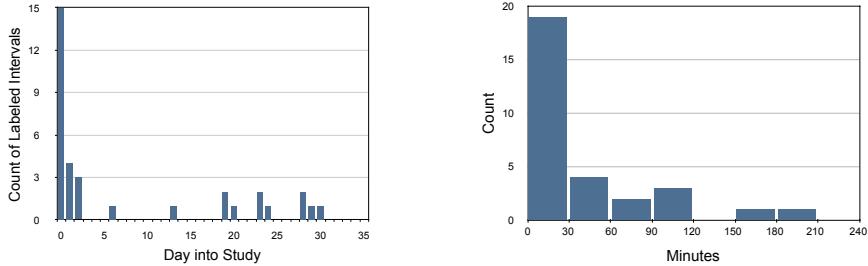
5.2 Results

WiFi Scans and Label Frequency. When running, the program conducted an active WiFi scan once every five seconds. A total of 322,089 WiFi measurements were taken. Each scan contained on average 6.6 beacons, with a standard deviation of 4.4.

Users labeled 31 intervals, with a surge on the first day, and declining frequency afterward (see Figure 6(a)). However, users continued to correct the system at a roughly constant rate until the end of the experiment, despite not receiving any reminders about the study other than the ambient awareness in the menu bar. Furthermore, continued labeling was not concentrated in a couple individuals—the contributions after the tenth day came from five different participants. All these results suggest that providing corrections is a low-overhead activity that can be sustained for at least a month.

Interval Characteristics. Figure 6(b) shows a histogram of interval durations. Most intervals were only a few minutes long. Of those under a half hour, five lasted less than a minute, and sixteen less than ten minutes.

Generally, users provided labels at the beginning of an interval. 28 intervals were labeled within the first two minutes. Of the remaining three intervals, one was labeled at the end of a half-hour interval, and two others were labeled in the middle of multi-hour intervals. From these observations we conclude that since users chose to enter corrections when arriving at a new place, this is the best opportunity for a more proactive system to query users for location data.



(a) Number of new labels added per day. Around a third of the labels were added on the first day. The decline and later uptake in labeling likely resulted from the holiday schedule.

(b) Histogram of labeled interval durations. Most intervals lasted less than a half hour. Note that there is an outlier not shown on the graph at 21.3 hours.

Fig. 6. Label Frequency and Interval Durations

Benefits of Labeling Intervals. To understand how much the system benefited from interval labeling, we examined the recorded data more closely. A sample of 1,000 WiFi measurements was drawn. Each scan was classified according to its most likely location, given the labels that the system knew about at the time the scan was taken. Two classifiers were compared, one that learned from all WiFi scans in previously labeled intervals, and another that learned only from the WiFi scan at the instant a label was assigned.

Figure 7 compares the distribution of maximum log-likelihoods for the class returned by each classifiers. The graph does not include the scans whose WiFi likelihood scores were zero, as explained in the caption. For the over 92% of

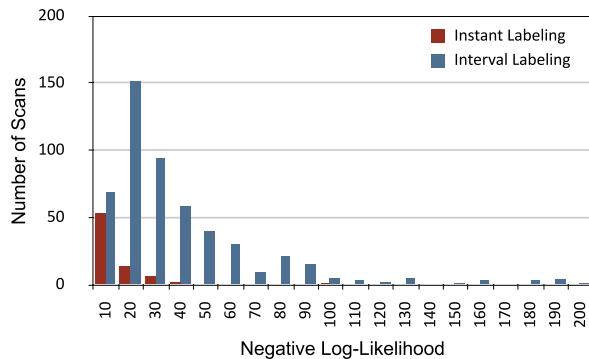


Fig. 7. Distribution of the log-likelihoods of 1,000 random WiFi scans, excluding those with zero likelihood (which include 484 for Interval Labeling, and 924 for Instant Labeling). The proportionally higher likelihood scores indicates that WiFi scans are more likely to find labels when using Interval Labeling than when using Instant Labeling.

scans in the instant labeling condition, the likelihood value gives no information about which label is best. Likelihood values can be computed, however, for over half of the scans in the interval labeling condition. Furthermore, even when a likelihood value is computed, the values are, in general, relatively higher in the interval labeling condition, which indicates greater certainty of the answers.

5.3 Survey

Following the user study, we surveyed participants to better understand the user experience. We felt that it was important to get users' perspective on both the accuracy of the system as well as the overhead involved in collecting the labels. Eleven of the participants responded to the survey.

Participants' perceptions about the system accuracy were mixed. On a Likert scale from 1–7, where 1 stands for “strongly disagree,” responses to “PILS often showed a wrong or missing label” had a mean of 3.0 and standard deviation of 1.9. But in response to “the accuracy got better over time,” responses averaged 4.3 with a standard deviation of 0.8.

In free responses, participants offered several improvement suggestions, such as reducing the latency to make an estimate and improving the autocompletion of labels. Two participants appreciated the music that played when the laptop was moving. One found it to be not only a useful form of feedback about the system's operation, but also an interesting prompt for social engagement. The other wanted to be able to choose the music from their iTunes library.

6 Discussion and Future Work

Previous work by King and Kjærgaard [16] has shown that knowing whether the user or device is in motion can be beneficial for several reasons. In our system, motion detection allows us to improve end-user data collection by supporting interval labeling instead of single measurements only. We use a very simple heuristic to differentiate between stationary and mobile intervals, yet it worked well in our prototype. In only very few cases did the motion detector report a false stationary state, while false reports of moving states never occurred.

Although these results indicate that prompting users for feedback when they arrive at a new place could minimize interruptibility, we did not focus on this aspect in this work. We plan to more thoroughly investigate this process in the future. We envision several options worth exploring, such as the active applications on the device, the time of day, or mouse and keyboard activity. We also plan to incorporate the results from interruptibility research into this process [14,23], as well as games for user labeling [26]. For example two users might give the same label for a room to win points.

Asynchronous labeling can also ensure that only “important” labels are solicited, such as the places that the user stays for long time periods or visits repeatedly. If the user stays at an unknown place for only a few minutes, PILS can omit the prompt, thus further reducing the intrusiveness of the system.

Our initial results from both the experimental study and the survey give a strong indication that the accuracy of location fingerprinting can be improved by interval labeling. However, about one third of the survey participants reported that accuracy seemed to decline over time, which could have arisen from long-term signal fluctuations or overfitting effects in the radio map. Consequently, we plan to evaluate how long to keep old measurements in the radio map, and the optimal number of measurements in a location fingerprint.

7 Conclusion

In this paper, we have presented a method to improve end-user supported location fingerprinting. By using the built-in accelerometer to detect motion, it is possible to observe and record WiFi signals during long periods. This greatly increases the number of WiFi measurements associated with a single given label. In addition, by making intervals the unit of labeling, the labeling process can be performed at a less obtrusive time, since users are more likely to recognize intervals of stability than they are to recall their locations at instants. Motion detection can also reduce the computational burden of inferring location when a label is given.

Our user study shows that labels can be collected without greatly burdening users, and that when such labels are applied to intervals, the maximum-likelihood of a new WiFi measurement is much higher than it would be if only instants were labeled.

We plan to further investigate how to improve accuracy through interval labeling. Moreover, we intend to study how to use retrospective labeling to increase the amount of labeled data while minimizing user effort.

References

1. Ashbrook, D., Starner, T.: Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing* 7(5), 275–286 (2003)
2. Bahl, P., Padmanabhan, V.: Radar: an in-building rf-based user location and tracking system. In: INFOCOM, Tel Aviv, Israel (January 2000)
3. Bao, L., Intille, S.: Activity recognition from user-annotated acceleration data. In: Ferscha, A., Mattern, F. (eds.) *PERVASIVE 2004*. LNCS, vol. 3001, pp. 1–17. Springer, Heidelberg (2004)
4. Bhasker, E., Brown, S., Griswold, W.: Employing user feedback for fast, accurate, low-maintenance geolocation. In: *Pervasive Computing and Communications (PerCom)* (January 2004)
5. Bolliger, P.: Redpin - adaptive, zero-configuration indoor localization through user collaboration. In: *Workshop on Mobile Entity Localization and Tracking in GPS-less Environment Computing and Communication Systems (MELT)*, San Francisco (2008)
6. Castro, P., Chiu, P., Kremenek, T., Muntz, R.: A probabilistic room location service for wireless networked environments. In: Abowd, G.D., Brumitt, B., Shafer, S. (eds.) *UbiComp 2001*. LNCS, vol. 2201, pp. 18–34. Springer, Heidelberg (2001)

7. Chai, X., Yang, Q.: Reducing the calibration effort for location estimation using unlabeled samples. In: *Pervasive Computing and Communications (PerCom)* (January 2005)
8. Froehlich, J., Chen, M., Smith, I., Potter, F.: Voting with your feet: An investigative study of the relationship between place visit behavior and preference. In: Dourish, P., Friday, A. (eds.) *UbiComp 2006*. LNCS, vol. 4206, pp. 333–350. Springer, Heidelberg (2006)
9. Haeberlen, A., Flannery, E., Ladd, A., Rudys, A.: Practical robust localization over large-scale 802.11 wireless networks. In: *International Conference on Mobile Computing and Networking (MobiCom)*, January 2004, pp. 70–84 (2004)
10. Harter, A., Hopper, A., Steggles, P., Ward, A., Webster, P.: The anatomy of a context-aware application. *Wireless Networks* 8(2-3), 187–197 (2002)
11. Hossain, A., Van, H., Jin, Y., Soh, W.: Indoor localization using multiple wireless technologies. In: *Mobile Adhoc and Sensor Systems (MASS)* (January 2007)
12. Ji, Y., Biaz, S., Pandey, S., Agrawal, P.: Ariadne: A dynamic indoor signal map construction and localization system. In: *International Conference On Mobile Systems, Applications And Services (MobiSys)*, April 2006, pp. 151–164 (2006)
13. Kaemarungsi, K.: Design of indoor positioning systems based on location fingerprinting technique. Dissertation, School of Information Sciences, University of Pittsburgh (January 2005)
14. Kern, N., Antifakos, S., Schiele, B., Schwaninger, A.: A model for human interruptability: experimental evaluation and automatic estimation from wearable sensors. In: *International Symposium on Wearable Computers (ISWC)* (2004)
15. Kern, N., Schiele, B., Junker, H., Lukowicz, P., Troster, G.: Wearable sensing to annotate meeting recordings. In: *Personal and Ubiquitous Computing* (January 2003)
16. King, T., Kjaergaard, M.B.: Composcan: adaptive scanning for efficient concurrent communications and positioning with 802.11. In: *International Conference On Mobile Systems, Applications And Services (MobiSys)*, January 2008, pp. 67–80 (2008)
17. King, T., Kopf, S., Haenselmann, T., Lubberger, C., Effelsberg, W.: Compass: A probabilistic indoor positioning system based on 802.11 and digital compasses. In: *Proceedings of the First ACM International Workshop on Wireless Network Testbeds, Experimental evaluation and CHaracterization (WiNTECH)* (August 2006)
18. Kotz, D., Newport, C., Elliott, C.: The mistaken axioms of wireless-network research. Technical Report TR2003-467, Dartmouth College (January 2003)
19. Krumm, J., Horvitz, E.: Locadio: Inferring motion and location from wi-fi signal strengths. In: *Mobile and Ubiquitous Systems: Networking and Services (MOBIQUITOUS)* (2004)
20. Lester, J., Choudhury, T., Borriello, G.: A practical approach to recognizing physical activities. In: Fishkin, K.P., Schiele, B., Nixon, P., Quigley, A. (eds.) *PERVASIVE 2006*. LNCS, vol. 3968, pp. 1–16. Springer, Heidelberg (2006)
21. Lim, H., Kung, L., Hou, J., Luo, H.: Zero-configuration, robust indoor localization: Theory and experimentation. In: *INFOCOM*, Barcelona, Spain (2006)
22. Mathie, M., Coster, A., Lovell, N., Celler, B.: Detection of daily physical activities using a triaxial accelerometer. *Medical and Biological Engineering and Computing* (January 2003)

23. Noy, Y., Lemoine, T., Klachan, C., Burns, P.: Task interruptability and duration as measures of visual distraction. *Applied Ergonomics* (January 2004)
24. Pan, S.J., Zheng, V.W., Yang, Q., Hu, D.H.: Transfer learning for wifi-based indoor localization. In: Association for the Advancement of Artificial Intelligence (AAAI) Workshop, May 2008, p. 6 (2008)
25. Schilit, B., Adams, N., Gold, R., Tso, M., Want, R.: The parctab mobile computing system. In: Workstation Operating Systems (January 1993)
26. von Ahn, L., Dabbish, L.: Labeling images with a computer game. In: Conference on Human factors in computing systems (CHI) (January 2004)

Using Context Annotated Mobility Profiles to Recruit Data Collectors in Participatory Sensing

Sasank Reddy, Katie Shilton, Jeff Burke,
Deborah Estrin, Mark Hansen, and Mani Srivastava

Center for Embedded Networked Sensing (CENS)
University of California, Los Angeles
`{sasank, kshilton, jburke, mbs}@ucla.edu,`
`destrin@cs.ucla.edu, cocteau@stat.ucla.edu`
<http://urban.cens.ucla.edu>

Abstract. Mobile phones and accompanying network layers provide a platform to capture and share location, image, and acoustic data. This substrate enables participatory sensing: coordinated data gathering by individuals and communities to explore the world around them. Realizing such widespread and participatory sensing poses difficult challenges. In this paper, we discuss one particular challenge: creating a recruitment service to enable sensing organizers to select well-suited participants. Our approach concentrates on finding participants based on geographic and temporal coverage, as determined by context-annotated mobility profiles that model transportation mode, location, and time. We outline a three-stage recruitment framework designed to be parsimonious so as to limit risk to participants by reducing the location and context information revealed to the system. Finally, we illustrate the utility of the framework, along with corresponding modeling technique for mobility information, by analyzing data from a pilot mobility study consisting of ten users.

Keywords: Participatory Sensing, Mobility Modeling, Location Based Services.

1 Introduction

Mobile phones and the cellular infrastructure are increasingly used for more than just communication. These platforms are being employed as tools to understand the habits and environments of individuals and communities. Many mobile phones are already equipped with acoustic, image, and location sensors in the form of microphones, cameras, and GPS, Wi-Fi, or cellular positioning, and contain a Bluetooth interface that can connect to external sensors. Phones also enable users to enter text to describe or record events. Emerging services take advantage of these diverse modalities to support active living, dietary monitoring, and analysis of environmental impact [1,2,3]. A complimentary vision, referred to as participatory or urban sensing, proposes tools to engage potentially large numbers of the general public in coordinated data gathering [4,5].

Intended to enable and encourage anyone to gather and investigate previously invisible data, participatory sensing can support advocacy - “making a case” through distributed documentation of a community need or an issue.

Mobile phones allow for easy, convenient and widespread data collection. Complimentary tools take advantage of phone networks and enable anyone to coordinate participants and initiate data collection “campaigns” that focus on social, political, or urban processes. Several challenges must be met for participatory sensing applications to flourish [6,7]. Among these is recruitment: enabling organizers, who may be community groups or simply motivated individuals, to select an interested and well-suited set of participants for a campaign based on the needs and specifications of the case they want to make. Campaign organizers might base recruitment around a number of factors. In this paper, we focus on enabling selection based upon the geographic and temporal availability of participants. Specifically, we describe a framework that enables organizers to select a group of participants based on a campaign’s contextual (further detailed in Section 3), spatial, and temporal coverage constraints. Furthermore, this recruitment system is designed to be parsimonious in its usage of context and location information to minimize disclosure risk for participants.

The paper is organized as follows: Section 2 describes a motivating application that illustrates why availability-based recruitment is useful. Section 3 discusses the recruitment problem in more detail. Section 4 overviews the design goals of the system and how the proposed recruitment stages adhere to these goals. System details are presented in Section 5, and we evaluate the recruitment framework using results from a pilot mobility study in Section 6. The paper ends with a discussion of future work items in Section 7.

2 Motivational Applications

Not all participatory sensing campaigns will need to select a limited pool of well-suited participants. Many campaigns will benefit from engaging as many people as possible, gaining the advantages of diverse backgrounds, interest, and availability [4]. Some campaigns, however, may face constraints that prevent them from incorporating all interested volunteers. If campaign organizers provide financial incentives for participation, distribute hardware to volunteers, or train individuals in specialized data collection, organizers may need to limit the number of possible participants or focus on select subset. Coverage-based recruitment can help organizers make selection decisions according to participants’ geographic and temporal availability.

An example campaign that benefits from coverage-based recruitment is inspired from the sustainability initiative at UCLA. In recent years, there has been a strong focus on raising campus awareness of environmental sustainability [8]. In coordination with the Education for Sustainable Living Program, we are undertaking a project to create a “green map” of campus sustainability resources. Volunteers will chart recycling bins and bicycle racks by taking geo-tagged photos [9]. They will also document negative impacts such as improper

waste disposal (recyclables in regular bins) and inefficient energy usage (lights continuously left on outdoors during daytime). Having images along with location information is important since it provides a visual reference of the point of interest and enables individuals to later annotate with additional data, for example noting the number of slots in a bike rack or the type of recyclable thrown in a regular waste basket. The campus sustainability campaign will take advantage of the geographic coverage that a selected group of volunteers can provide by asking volunteers to contribute information that they see as they go about their regular routines. The participants will focus on mapping “tasks” that run weeks at a time. The information that is collected will be used to improve current sustainability processes (e.g. by suggesting better placement of recycle bins or informing facilities where efficiency problems exist) as well as to help educate the UCLA community of areas for improvement.

The campus sustainability campaign will benefit from coverage-based recruitment because it depends on observations in a limited region (UCLA) over a long period (several weeks). Further, during the campaign, we will provide participants with the mobile devices and accompanying data plan necessary to perform the data collection. We will also hold training sessions to help participants understand the mapping tasks and goals. For these reasons, we must focus resources on a select number of participants. Thus, it is important to recruit participants whose availability, in terms of context (in this case transportation mode), space, and time, matches campaign needs. The campaign will remain open for anyone on campus to participate with their own mobile phones, but the hardware resources and training will be limited to a select few.

In this campaign, well-suited participants regularly walk on campus during the daytime (individuals that run, bike, or drive may be less likely to notice points of interest, and collecting clear photographs is more difficult at night); cover as much area as possible; and are consistent in their routes and routines. Recruiting participants whose availability matches the campaign coverage constraints will provide the best chance of documenting sustainability resources using a limited pool of individuals in a constrained time frame. Throughout the rest of the paper, the campus sustainability campaign is used as an example to explain the details of the recruitment framework.

Although we detail only one campaign as an example here, many other data collection initiatives could benefit from a coverage-based recruitment system. Examples include the CycleSense initiative, which tasks bike commuters to collect information about the quality of bike routes and paths, and the Walkability project, designed to gather safety issues of walking paths in local neighborhoods [10,11].

3 Problem Description and Challenge

Like many crowd-sourcing services on the web [12], a campaign seeks interested participants willing to volunteer their time to help with the data collection task. For certain campaigns it might be appropriate to focus on a specific set

of volunteers from an overall pool. In this situation, choosing volunteers wisely becomes critically important. Note that this parallels the recruitment that occurs in web services that provide a marketplace for commissioned work, such as Amazon Mechanical Turk and GURU.com, in that well-suited individuals are preferred [13,14]. Organizers of campaigns may wish to consider a number of factors. For example, organizers could request that participants have specific sensor capabilities (e.g. camera, microphone). Organizers may also wish to recruit participants who have certain performance standards based on previous involvement in campaigns [15] and are willing to be flexible to conform to sensing requests. This paper, however, focuses on a third requirement: the geographic and temporal availability of participants.

At a technical level, the recruitment problem in participatory sensing is similar to that of static sensor selection and placement [16] and robotic motion coordination for sensing [17]. The distinction is that participatory sensing must consider human mobility, which is not directly controllable. However, mobility-based recruitment does takes advantage of the fact that people's routes and locations are often regular and repetitive [18,19]. This work differs from existing systems that use mobile phones for sensing by concentrating on selecting a set of participants so that geographic coverage can be maximized. Our approach also takes into account variation of participant mobility over time. Previous work has geared task assignment for opportunistic in-situ sensing [20,21] or has focused on initiating sampling around specific location "bubbles" (regions) [22,23]. Furthermore, our work focuses on campaigns targeting phenomenon that are not easily expressed as Gaussian processes or do not have spatial and temporal distributions that are known a priori [24]. For instance, locations of sustainability assets are often non-uniformly distributed and instances of improper resource (energy, waste, water) usage are dependent on processes that are dynamic based on space, time, and other external factors (which will need to be learned through data collection).

We outline a recruitment engine that uses campaign specifications provided by an organizer to select a limited set of potential volunteers based on participants' previously-gathered mobility profiles. A mobility profile is derived from the participants' context-annotated mobility traces: streams of previously-collected location, time, and context data. Location and time are obtained via GPS receivers embedded in mobile phones or from cellular network infrastructure [25]. Context includes a number of inferences drawn from sensors available on the mobile phone, but we specifically concentrate on the transportation mode (walking, running, biking, or in motorized transport) of an individual which can be obtained by analyzing GPS traces [26].

To define mobility-based requirements for their campaign, a campaign organizer would limit a campaign to geographic regions (e.g. UCLA), temporal boundaries (e.g. between 2/1/09 and 6/1/09), and specific time period (e.g. weekdays between 8 a.m. and 6 p.m.). Organizers could also specify important contextual information, such as preferred modes of transportation. The recruitment engine would match these specifications with participants' mobility profiles. The challenge for a engine is finding a subset of volunteers whose combined

mobility profiles best fulfills the coverage requirement. Finding the appropriate set of participants requires iterating through all subset combinations of individuals considering ordering since availability could be redundant. And since participant routes and habits may differ over the course of a campaign, the recruitment system needs to analyze whether the mobility profiles of participants change as the campaign runs and alert organizers to possible coverage problems.

4 Design Goals

Recruiting volunteers for participatory sensing is somewhat analogous to recruiting volunteers or employees in non-virtual environments. Considerations include geographic and temporal availability as well as qualifications and willingness. Drawing on this similarity, we have created a coverage-based recruitment system that consists of three distinct stages: the qualifier, interview, and progress review, modeled after real-world recruitment processes. Figure 1 shows each of these three recruitment steps with the perspective of campaign actions. Furthermore, an outline of the three stages exists below.

- **The Qualifier:** To be considered for a campaign, potential participants must have a certain number of significant locations, routes, and desired transportation modes within a certain time and space.
- **The Interview:** The recruitment engine compares participants that meet initial qualification requirements to determine which limited subset of individuals maximize coverage over a specific area and time period.
- **The Progress Review:** As a campaign runs, the recruitment engine checks the coverage consistency of participants by periodically evaluating their mobility during the campaign against their qualifying profile. If similarity scores are below a defined level, the engine alerts campaign organizers so they may take action (check for coverage, provide feedback, etc.)

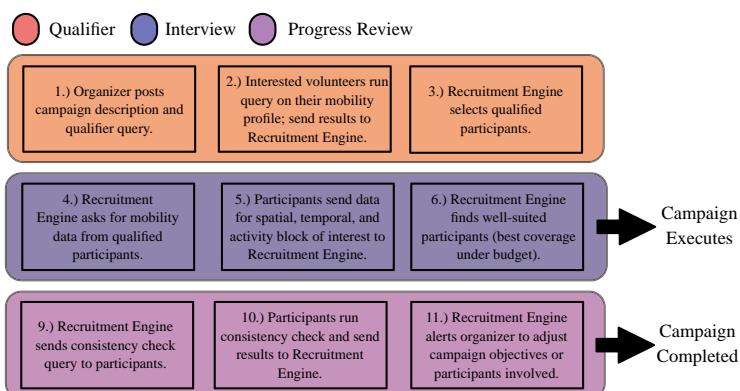


Fig. 1. Campaign Recruitment Flow with Recruitment Steps Labeled

During the design of the recruitment system, we were concerned with how best to handle querying and sharing of participant mobility information. Because this information describes an individual’s routines and habits, sharing mobility profiles presents a number of exposure risks to individuals [27,28,29]. Thus, a major design principle employed for the recruitment system was parsimony: sharing the minimal amount of information needed to meet an application’s goals [30,31]. The recruitment system uses a framework that limits the amount and granularity of mobility information shared during each step of the recruitment process. In this section, we detail how our system achieves this design goal by describing how we minimize what information is shared, with whom it is shared, and how long it is retained during each stage of the recruitment process. All of these considerations are important factors in aiding participants to understand and control the privacy of their data [32]. We assume that all mobility information used to generate a profile resides with the participant in a private data store, and participants must opt-in to allow queries to be run on this store.

If a participant is interested in a campaign, they can opt to run the qualifier query on their mobility information. These queries are created by campaign organizers and run on the participant’s data store. Only aggregate results are shared with a campaign organizer (e.g. whether geo-spatial qualifications have been met or how consistent a profile is). The campaign organizer therefore has no access to a participant’s detailed location information during the qualifier stage. Only during the interview stage does data from the mobility profile need to be shared with campaign organizers. In order to find a cover set among the participants, the recruitment engine needs access to location information. At this stage, however, not only have participants expressed interest in a campaign, they have a good chance of being chosen to participate. If a participant trusts a campaign organizer and has applied to their campaign, more information sharing may be justifiable. Parsimony, however, remains important. Participant mobility data shared with the recruitment engine is limited to a particular spatial region (e.g. UCLA) and time span (e.g. weekdays). Also, rather than sharing granular location information, the recruitment engine provides the organizer with a generalized coverage map of the chosen region (see Figure 4 in Section 6.3). If the coverage area is small, an organizer may be able to infer participants’ routes, but a targeted coverage area limits exposure of locations not relevant to the campaign. Once the interview process has ended, the system deletes the shared location data and only maintains the participant subset coverage data for the duration of the campaign. Finally, like the qualifier, the progress review is run on the participant’s store, and only aggregate measures of consistency are shared with campaign organizers. An organizer will know, for example, if a participant follows similar routines as determined by the interview process, but exact routes and locations will not be reported.

5 System Details

The steps involved in the recruitment process and the pre-processing procedures needed to build mobility profiles are detailed in this section. We place particular

emphasis on describing the underlying algorithms employed. We also explain the inputs and outputs of each stage along with parameters that need to be specified for execution of the steps in the framework.

5.1 Building Mobility Profiles

Each of the recruitment stages depends on the transformation of a participant’s raw mobility information into elements that can be used for recruitment. We assume that participants have previously used their mobile phones to collect raw data in the form of GPS traces (latitude, longitude, time). Services already exist to collect and manage this type of data [33,34], and we expect that historical location data will become increasingly available. Mobility information can also be obtained by having individuals annotate maps manually, but in this paper we focus on mobility data that is verified through actual in-field data collection. We plan to handle heterogeneous sources of mobility information along with allowing individuals to specify different granularities of data as future work (Section 7).

The recruitment engine can process raw mobility data to infer significant locations and routes along with associated transportation modes. Over time, this processed data can form participant mobility profiles. The steps required to build these profiles include:

- **Pre-Processing:** Since sampling rate may vary, the engine first normalizes the GPS logs to a set sample rate (every 30 seconds in our experiments) and fills in missing values (e.g. when an individual goes indoors and loses GPS signal). For cases where there is a gap in data and the points before and after are significantly different, the engine generates a likely route using Google Route generator, taking into account time and distance traveled [35].
- **Destination and Route Inference:** Next, the engine finds places where users spent a continuous amount of time (15 minutes) within a certain distance (50 meters) and considers this a “stay” [36]. Stays within a distance (250 meters) are clustered into “destinations” using density based clustering [37]. Destinations are then used to divide a GPS log into significant locations and routes. Time and distance thresholds were chosen based on varying the parameter combinations and analyzing which were most effective at representing participants’ notions of destinations. Routes are clustered using average minimum point-segment distance as the comparison method (threshold of 100 meters) and performing hierarchical clustering [38].
- **Route Transportation Mode Inference:** Once routes have been identified, they need to be labeled in terms of transportation mode. The routes are first segmented based on change points (speed close to zero, loss of GPS signal) [26]. Then features such as the average, maximum, and minimum speeds, and total distance for each segment are calculated, and used to classify in terms of transportation modes with likely transitions considered [26]. Also, we have explored using additional sensors on phones, such as accelerometers and GSM/WiFi radios, to help with classification [30,39].

5.2 Running the Qualifier

Campaign organizers can use participants' mobility profiles to select volunteers from as many interested, qualified candidates as possible. This is analogous to a job or volunteer hiring process where recruiters only want to see resumes from a pool of candidates who meet a set of minimum requirements. A participatory sensing organizer would therefore post a description of a campaign including its purpose, type of data needed, lifetime, and contribution expected. The organizer would also specify qualifications based on geographic and temporal coverage by using attributes such as the total number, uniqueness, transportation mode, and the minimum time needed for locations and routes in a specific spatial region and time period. The qualifier can be defined based on routes that have start/end points that are in a specific zone as well.

The recruitment engine would run the qualification filter on interested participants' mobility profiles, and only share with the organizer whether an individual meets the criteria. Interested participants who do not explicitly meet the qualification may choose to share more detailed information, such as how many more routes or locations they would need to meet the qualifier, so that the organizer has the choice to include them if needed. The consistency of a participant's mobility information can also be used as part of the qualification. We discuss this concept in more detail in the progress review stage below.

5.3 Running the Interview

Once the recruitment engine has identified a pool of participants who meet minimum qualifications, the next step is the interview. In traditional employment or volunteer recruitment, the interview phase gives employers a better understanding of the skills possessed by a set of candidates. Similarly, the participatory sensing interview evaluates the mobility qualities of a set of potential participants, and calculates which subset would maximize a coverage utility function.

In a technical sense, the coverage-based interview process is an instance of the budgeted maximum coverage problem [40]. Essentially, we have a participant pool $P = \{p_1, p_2, \dots, p_n\}$ with individual non-negative costs $\{c_i\}_{i=1}^n$ defined over a spatial and temporal block elements $E = \{e_1, e_2, \dots, e_n\}$ with associated utilities $\{u_i\}_{i=1}^n$. The goal is to find a subset of participants, $P^* \subseteq P$, for the campaign such that the utility of elements covered by subset, $U(P^*)$, is maximized while the cost of the subset, $C(P^*)$, is under a set budget, B , [24,40,41]. In summary, we are solving the following optimization problem:

$$\operatorname{argmax} U(P^*) \text{ subject to } C(P^*) \leq B$$

The budget is modeled as the resources needed to engage participants in a campaign. Each participant has a cost, which might include both compensation and a system opportunity cost for the individual's participation (i.e. if adding a participant elicits organizational or administrative costs for a campaign). The spatial and temporal blocks, E , are defined by specifying a region and time of interest (e.g. UCLA, weekdays from 8 am - 6 pm), spatial and temporal granularities

(e.g. 10000 meter² blocks, 5 minute time spans), and a set of transportation modes. Utility weights are associated with each block and are specified by the campaign organizer. To get a more intuitive sense of how utility weights can be assigned, we consider two cases: uniform weighting and priority based. In the first, all spatial and temporal blocks have equal weights, so the utility of each block could be set uniformly to 1 and the maximum aggregate utility is simply the number of blocks. The second case is priority based where certain blocks are more important to be sensed. This can be reflected by assigning higher utility values to the priority blocks and lower values to less important blocks.

The optimization problem of finding the best subset of participants is proven to be NP-hard [40,42]. Selecting a participant to be involved in a subset changes the utility for the participants not included, and one would have to search through all combinations of subsets to find the best solution. Since our utility function is both sub-modular (adding a participant helps more if fewer participants are included in the subset and less if there are more participants in the subset already) and non-decreasing (the utility of the subset is less than the set that it came from), the standard greedy algorithm is guaranteed to find a constant fraction (63%) solution when the costs of the participants are the same [42]. In the more complex setting where the costs of participants varies, a benefit-cost greedy algorithm where the ratio of utility to cost is considered when adding participants could be employed to obtain a solution [41]. Also, the dual problem of minimizing cost of the set given a particular coverage requirement can also be solved using the greedy algorithm with a log-factor approximation of the optimal value.

5.4 Running the Progress Review

The end of the interview process results in selecting a set of participants to execute the campaign. For certain campaigns, there may need to be additional checkups to ensure that the mobility of participants is consistent with their pre-established profiles and coverage over the area of interest is being maintained. This is especially true for campaigns with a long lifetime, during which people's routines may change. Thus, a "progress review" can check a participant's original mobility profile for similarity with current behavior. The campaign organizer sets the intervals for, and frequency of, reviews.

In order to check for similarity, the recruitment engine models a participant's mobility profile for a particular period of time using an $m \times n$ matrix with a set of corresponding transportation modes [19,43]. The m rows in the matrix represent spatial blocks (e.g. 10000 meter² grids, zip codes, census blocks) while the n columns model distinct time spans. An entry in the matrix would be the proportion of time spent performing a set of transportation modes within the time period of interest in a particular spatial block for a time span. Based on previous work that explores daily human location patterns, we chose a day as a representative time span for our analysis [18,19,43]. This matrix is normalized by applying the arcsine-root transformation [44]. Throughout the rest of the paper, we refer to this data construct as an "association matrix", A .

Since we are interested in dominant mobility patterns, a summarization method is necessary. An effective technique to analyze multi-dimensional data for major patterns is Principal Component Analysis (PCA). PCA finds a set of orthogonal axes (eigenvectors) that linearly transform the original data so that the largest amount of variance is explained [45]. The first coordinate axis explains the most variation in the data and the subsequent axes explain a decreasing amount of variation. The eigenvalues obtained from PCA explain the amount of variation for which each eigenvector accounts. To obtain the dominant patterns, the original data is projected onto the eigenvectors weighted by the eigenvalues. Oftentimes when analyzing location association data, the resulting projections are referred to as “eigenbehaviors” [19,43].

In practice, a more flexible technique to find eigenbehaviors is Singular Value Decomposition (SVD) [45]. Given the association matrix A , the SVD would be:

$$A = U \cdot \Sigma \cdot V^t$$

In this decomposition, U and V are the left and right singular vectors and Σ is the diagonal matrix of singular values. Thus, in the terms of PCA, the singular values are the square roots of the corresponding eigenvalues, the columns of V represent the eigenvectors, and the columns of U are the eigenbehaviors [19,43,45]. Since the eigenbehaviors represent patterns that are common across different time spans (days) and the singular values represent the importance of each pattern, one can compare consecutive time periods by taking the cosine similarity of the behavior vectors weighted by the singular value importance (percentage of variance represented by a singular value) [43]. Hence, if there exists two eigenbehaviors, U_{t1} and U_{t2} , representing different time periods, $t1$ and $t2$, with singular value importance, W_{t1} and W_{t2} , the similarity metric would be defined as:

$$\text{Similarity}(U_{t1}, U_{t2}) = \sum_{i=1}^{\text{rank}(U_{t1})} \sum_{j=1}^{\text{rank}(U_{t2})} w_{t1_i} w_{t2_j} |U_{t1_i} \cdot U_{t2_j}|$$

This measure of similarity [43] (or consistency) is indexed from 0 (least similar) to 1 (most similar) by normalizing on the base eigenbehavior similarity. Ideally, not all eigenbehavior vectors need to be used. If a large variation of the behaviors is explained by a few vectors, then the remainder of the components can be ignored. This SVD based approach has been used previously to analyze videos, documents, and cellular/WiFi association patterns for clustering purposes [19,43,46,47]. We instead focus on using this technique as a measure of an individual’s consistency of mobility patterns over time.

6 Experiments

Because we are in the preliminary stages of several campaigns, we base our evaluation of the coverage-based recruitment system on mobility profiles generated during a pilot mobility study. Running this data through our recruitment process, with the campus sustainability campaign as context, illustrates the performance and usefulness of the different stages of our approach.

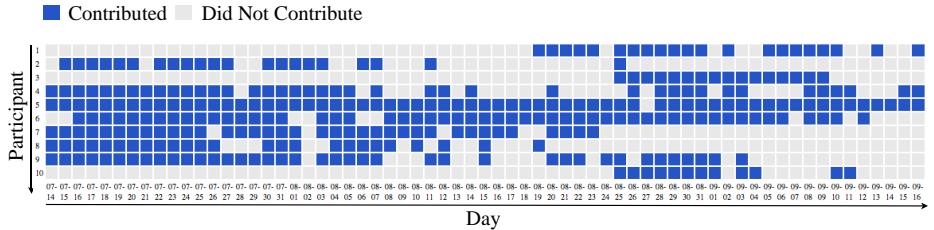


Fig. 2. Mobility Study Participation Statistics (Organized by Participants/Days)

6.1 Generating Sample Mobility Profiles

We recruited ten participants for the pilot study and asked them to carry a Nokia n95 to automatically capture and upload a GPS time series (latitude, longitude, and speed every 30 seconds). We used this information to create transportation annotated mobility profiles for the participants. The study was conducted over a time-span of sixty five days where seven users were recruited at the start and another three were phased in during the mid-point (due to equipment availability). Participants were requested to log information at all times, but had the right to turn off their devices at any time (Figure 2). Participants were all affiliated with UCLA so that a common area of coverage would be available.

6.2 Qualifier: Evaluating UCLA Destinations and Walking Routes

For the campus sustainability campaign, well-suited candidates would have walking routes to and from places on campus. Thus, a qualifier filter for this campaign could be: “Participants that have at least 4 daytime walking routes with destinations starting or ending at UCLA during a week.” Using this qualifier on pilot participants’ sample 5-weekday stretches indicates that seven participants (#3, #4, #5, #6, #7, #8, #9) would qualify for the campaign while three

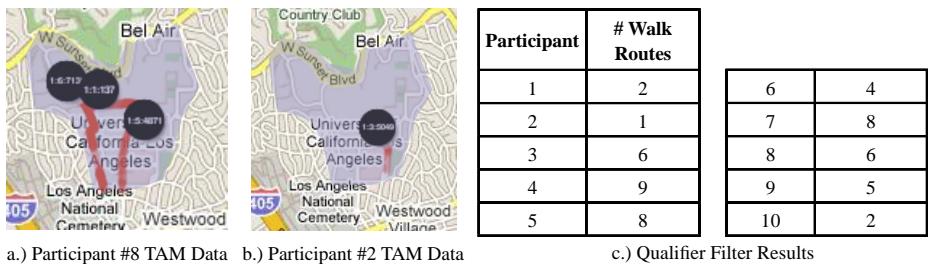
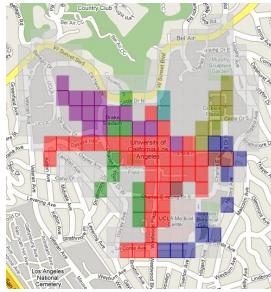


Fig. 3. Mobility Information for Participants and Qualifier Results

Participant	Coverage Score
3	101
4	88
5	51
6	44
7	74
8	87
9	96

a.) Maximum Cover Results



Participant	Color	6	Red
3	Blue	7	Green
4	Green	8	Purple
5	Teal	9	Red

b.) Map of Coverage Results

The map shows the participants that had the most coverage for that particular spatial area.

Fig. 4. Maximum Walking Coverage of UCLA During Daytime Weekday Hours

(#1, #2, #10) would not. Figure 3 shows the mobility information during a 5-weekday span for one participant that matches the qualifier (#8) and one that does not (#2). Further, the figure provides the walking route statistics for each participant. User #8 has many walking routes on campus (this individual lives on campus and prefers walking to other transport modes). User #2 has only one walking route (this participant drives to campus and parks under their office).

6.3 Interview: Evaluating Best Coverage of UCLA

To evaluate the coverage of the UCLA campus by qualified participants, we perform a set cover for a daytime hours during a 5-weekday span where the spatial granularity is set to 10000 m^2 blocks, the time block granularity is 1 hour, and the activity is walking. The results of maximizing coverage, with the budget set to infinity, and individual participants costs and block utility weights all set evenly, are shown in Figure 4.

The results indicate that participant #3 is the most valuable since he spans the most spatial/temporal blocks. In terms of spatial area alone, participants #8 and #9 provide the most coverage. These two individuals both live near campus (in close proximity) and often walk to UCLA from home. Notice that participant #9's score is much higher than #8's. This is due to the fact that #9 and #8 have common coverage, and the common areas count toward #9's score, since she spans more blocks overall. The interview process can also adapt to campaign budgets and participant costs. For instance, if the campaign had a budget of five credits, where each individual had a cost of one credit, then the interview process would eliminate participants #5 and #4.

In the above analysis, the specifications for the spatial and temporal blocks were chosen in a “generic” fashion to illustrate the capabilities of the interview process based on the mobility data available. But for the sustainability campaign, different specifications might exist for the attributes. For instance, emphasis could be placed in certain regions (around specific buildings) and time periods

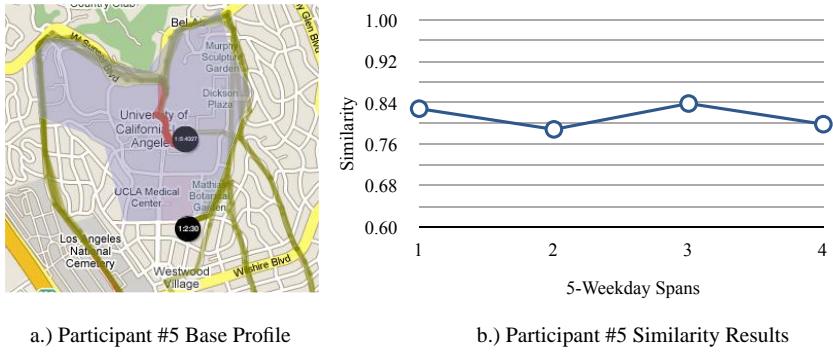


Fig. 5. Consistency of Participant #5’s Mobility Profile (Base Compared to 5 Weeks)

(during peak campus occupancy), the time granularity could be changed (“day” level indexes of time as opposed to individual hours and spatial blocks contoured to walking paths on campus).

6.4 Progress Review: Comparing Similarity of Profiles over Weeks

If participants’ availability deviates from their established mobility profiles during a campaign, the organizer should be alerted. The inconsistent participant could be checked to see if their coverage still meets coverage requirements by running the interview process or the organizer might need to recruit additional participants to help with the campaign. The progress review consistency “check up” is especially important for long running data collections, such as the campus sustainability campaign, since participants’ behavior can change over time.

To demonstrate the utility of the performance review, the availability changes of two participants (#5 and #9) in the pilot mobility study is analyzed. Note that since the spatial granularities are set again to 10000 m² blocks and a 5-weekday span is analyzed, the association matrix is 208 (representing the blocks at UCLA) by 5 (number of days) in size. Participant #5 provided GPS information for the longest period, and an interview determined that the individual was very consistent in his routes and habits over the time span. To test whether the progress review technique would corroborate this consistent behavior, we employed SVD to obtain eigenbehaviors based on daytime walking instances for the first 5-weekday span, and then compared them with the eigenbehaviors from the next four 5-weekday spans at the UCLA location. The top three eigenbehaviors were used, in this comparison and the one below, since they represented over 90% of the variance (using all five components results in similar consistency measures as considering the top three). Figure 5 (a.) illustrates the mobility information for the base week, and Figure 5 (b.) depicts the similarity results for the progress review. The average similarity over the time span was 0.82.

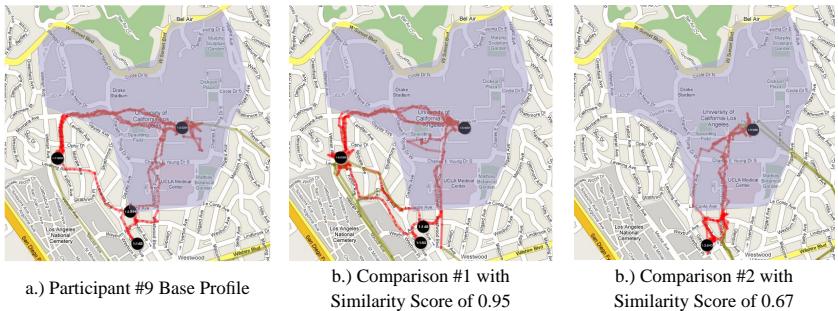


Fig. 6. Consistency of Participant #9’s Mobility Profile (Base Compared to 2 Weeks)

Not all users are as consistent as Participant #5. For instance, Participant #9’s mobility drastically changed over the campaign period. An interview revealed that the participant moved her home address during the time period. Thus, her daily routes and locations changed significantly. We performed a consistency check comparing two consecutive 5-weekday spans, between which the move occurred, to her base profile (Figure 6). As expected, the first week is fairly consistent with the base profile, achieving a similarity score of 0.95. The second week is significantly different, achieving a similarity of only 0.67.

As shown above, it is not always necessary to use all the eigenbehaviors when performing a consistency check since dominant components could exist that explain the majority of the variance. To further illustrate this point, we analyzed all participant mobility data in regards to the UCLA campus and daytime walking patterns with 10000 m^2 spatial blocks which results in a $208 \times n$ coverage days association matrix. Table 1 shows the number of days in which there was a coverage match for each participant along with the fraction of variance represented by the top three eigenbehaviors. Based on the analysis, the top three components can represent an average of 0.76 fraction of the variance of the participants’

Table 1. Participant UCLA Walking Totals and Variance Fraction Represented by Top 3 Components

Participant	# of Days	Variance Fraction
1	5	0.90
2	8	0.70
3	4	0.94
4	13	0.62
5	28	0.64
6	8	0.74
7	8	0.70
8	12	0.69
9	15	0.78
10	4	0.89

behavior. These dominant patterns represent walking behavior influenced by location choices for work and going to different places for dining.

7 Future Work

We are currently building the overall system to manage campaigns: enabling organizers to easily create, manage, organize data collections, and enabling participants to find and participate in campaigns in which they are interested in. This final section details future work on campaign recruitment. This includes exploration of making the recruitment process clear for participants involved, as well as enhancements to make the recruitment more flexible and effective.

7.1 Designing for System Legibility

Because a coverage-based recruitment system demands that participants share information about their locations, habits and routines, privacy is an important concern. We define privacy as a negotiation between participants and organizers of what information to share or withhold. Privacy decision-making is contextual and individually variable [48,49,50].

To negotiate information disclosure, individuals should understand the benefits and repercussions of sharing their data. To make an informed decision, they must understand who is asking for the data (identity); what the data will reveal about them (granularity); what the organizer wants to use the data for (purpose); and how long data will be retained by a requesting organizer (retention). Through each stage, the system communicates to participants the nature of the query they are running, what information the query shares and with whom, and how long the information can be retained by the campaign organizer. The system should also communicate the identity and reputation of a campaign organizer to potential participants, so that participants can decide whether to trust an organizer with their mobility information. Communication between the system and the participant is essential to the system's legibility: the ways in which a system enables people of all technical backgrounds to make informed disclosure decisions. Our ongoing work explores methods to represent disclosure to participants to aid informed decision-making.

7.2 Recruitment Process Enhancements

In this paper, we focused on coverage-based recruitment, but there are other elements that future work will add to the recruitment process as well. In addition to a participant's sensing capabilities and their past campaign performance and participation, we plan to integrate social network membership or external credentials as factors in recruitment [15]. Furthermore, we will consider a participant's responsiveness, which includes both their willingness and flexibility to perform specific sensing requests that might deviate from their own objectives when tasked by the campaign organizer.

Another area to investigate is how the level of parsimony affects the uncertainty of a participant's qualification for a campaign. This will most affect the interview stage of the recruitment process where mobility profile information needs to be shared with the recruitment engine as opposed to results of a query. If a campaign organizer has coverage constraints based on coarse level mobility information, it will be less of a disclosure risk for participants but it might make figuring out which set of participants to recruit much more difficult. On the other hand, creating coverage requirements that require fine-grained information might result in obtaining a more well-suited participant set, but has the downside of a higher privacy risk. We are interested in designing tools so that organizers can balance participant disclosure risk and qualification uncertainty.

The current recruiting system has a fairly rigid structure: a campaign is defined and participants are compared against organizer specifications based on profile information. To make the system more adaptable, a negotiation step, where participants can specify a level of service to which they are willing to commit, could be incorporated as well. Also, the system could allow participants to specify different incentive criteria including not having any specified at all, being rewarded based on performance, or just obtaining a flat reward for participation as a whole. Additionally, coverage-based recruitment relies on having access to historical mobility information, but this data might not always be available. Thus, the framework should be able to incorporate manually-specified availability as well. An approach we will explore is placing confidence weights on different sources of coverage information. Also, the mobility information that is specified might range in terms of granularity, so the system would need to represent availability uncertainty based on the resolution of data provided. Finally, we are interested in exploring how the recruitment system can operate during the campaign as a run-time tool for the designer. Thus, exploring the robustness and accuracy of the mobility analysis technique (employing eigenbehaviors) when faced with daily or sub-weekly updates will be a point of analysis along with exploring how feedback and participant changes affect reaching the campaign coverage goals.

Acknowledgments

This work is supported by NSF Grant CNS-0627084 and IIS-0832873. Hardware is provided by Nokia Research Labs under the Sensor Planet initiative.

References

1. Consolvo, S., McDonald, D., Toscos, T., Chen, M., et al.: Activity sensing in the wild: a field trial of ubifit garden. In: Computer Human Interaction (2008)
2. MycaNutrition: MyFoodPhone Nutrition (2008), <http://mycanutrition.com>
3. CENS: Personal Environmental Impact Report (2008),
<http://peir.cens.ucla.edu>
4. Burke, J., Estrin, D., Hansen, M., Parker, A., Ramanathan, N., Reddy, S., Srivastava, M.: Participatory Sensing. In: World-Sensor-Web, Sensys (2006)

5. Eisenman, S., Lane, N., Miluzzo, E., Peterson, R., Ahn, G., Campbell, A.: MetroSense Project: People-Centric Sensing at Scale. In: World-Sensor-Web, Sensys
6. Campbell, A., Eisenman, S., Lane, N., Miluzzo, E., et al.: The Rise of People-Centric Sensing. IEEE Internet Computing 12(4), 12–21 (2008)
7. Parker, A., Reddy, S., Schmid, T., et al.: Network System Challenges in Selective Sharing and Verification for Personal, Social, and Urban-Scale Sensing Applications. Hotmobile (2006)
8. UCLA Institute of the Environment: Sustainability (2008), <http://sustain.ucla.edu/>
9. GreenMap: Green map - think global, map local (2008), <http://greenmap.org/>
10. Shilton, K., Ramanathan, N., Reddy, S., Samanta, V., Burke, J., Estrin, D., Hansen, M., Srivastava, M.: Participatory Design of Sensing Networks: Strengths and Challenges. In: Participatory Design Conference (2008)
11. LiveablePlaces: Liveable Places, Making the Connections Community Workshop (2008), <http://liveableplaces.org>
12. Brabham, D.: Crowdsourcing as a Model for Problem Solving: An Introduction and Cases. Convergence 14(1), 75 (2008)
13. Amazon: Amazon mechanical turk (2008), <http://mturk.com>
14. GURU.com: Freelancers at online service marketplace (2008), <http://guru.com>
15. Reddy, S., Shilton, K., Burke, J., et al.: Evaluating Participation and Performance in Participatory Sensing. In: UrbanSense, Sensys (2008)
16. Ganesan, D., Cristescu, R., Beferull-Lozano, B.: Power-efficient sensor placement and transmission structure for data gathering under distortion constraints. ACM Transactions on Sensor Networks (TOSN) 2(2), 155–181 (2006)
17. Kansal, A., Kaiser, W., Pottie, G., Srivastava, M., Sukhatme, G.: Reconfiguration methods for mobile sensor networks. Transactions on Sensor Networks (2007)
18. Gonz, M., et al.: Understanding individual human mobility patterns. Nature 453(7196), 779 (2008)
19. Eagle, N., Pentland, A.: Reality mining: sensing complex social systems. Personal and Ubiquitous Computing 10(4), 255–268 (2006)
20. Miluzzo, E., Lane, N., Campbell, A.: Virtual Sensing Range (Poster Abstract). In: Proc. of ACM 4th Intl. Conf. on Embedded Networked Sensor Systems, pp. 1–3 (2006)
21. Eisenman, S., Lane, N., Campbell, A.: Techniques for Improving Opportunistic Sensor Networking Performance. In: Nikoletseas, S.E., Chlebus, B.S., Johnson, D.B., Krishnamachari, B. (eds.) DCOSS 2008. LNCS, vol. 5067, pp. 157–175. Springer, Heidelberg (2008)
22. Lu, H., Lane, N., Eisenman, S., Campbell, A.: Bubble-Sensing: A New Paradigm for Binding a Sensing Task to the Physical World using Mobile Phones. In: Workshop on Mobile Devices and Urban Sensing, IPSN (2008)
23. Gaonkar, S., Li, J., Choudhury, R., Cox, L.: Micro-Blog: sharing and querying content through mobile phones and social participation. In: Mobicom (2008)
24. Krause, A., Horvitz, E., Kansal, A., Zhao, F.: Toward Community Sensing. In: ACM Conference on Information Processing in Sensor Networks (IPSN) (2008)
25. Hightower, J., Borriello, G.: A survey and taxonomy of location systems for ubiquitous computing. IEEE Computer 34(8), 57–66 (2001)
26. Zheng, Y., Liu, L., Wang, L., Xie, X.: Learning transportation mode from raw gps data for geographic applications on the web. In: WWW Conference (2008)
27. Anthony, D., Henderson, T., Kotz, D.: Privacy in Location-Aware Computing Environments. Pervasive Computing, IEEE 6(4), 64–72 (2007)

28. He, Q., Wu, D., Khosla, P.: The Quest for Personal Control over Mobile Location Privacy. *IEEE Communications Magazine* (2004)
29. Shahabi, C., Khoshgozaran, A.: Location Privacy in Geospatial Decision-Making. In: Bhalla, S. (ed.) *DNIS 2007. LNCS*, vol. 4777, pp. 1–15. Springer, Heidelberg (2007)
30. Mun, M., Estrin, D., Burke, J., Hansen, M.: Parsimonious Mobility Classification using GSM and WiFi Traces. In: *IEEE EmNets* (2008)
31. Shilton, K., Burke, J., Estrin, D., Hansen, M., Srivastava, M.: Participatory Privacy in Urban Sensing. In: *Workshop on Mobile Device and Urban Sensing, IPSN* (2008)
32. Waldo, J., Lin, H., Millett, L.: Engaging Privacy and Information Technology in a Digital Age. National Academies Press, Washington (2007)
33. Yahoo: FireEagle - Take Location Data to the Web (2008), <http://fireeagle.com>
34. MapMyRide: Map Cycling and Biking Routes (2008), <http://mapmyride.com>
35. Google: Google Maps API (2008), <http://maps.google.com>
36. Kang, J., Welbourne, W., Stewart, B., Borriello, G.: Extracting places from traces of locations. *Mobile Computing and Communications Review* 9(3), 58–68 (2005)
37. Zhou, C., Frankowski, D., Ludford, P., Shekhar, S., Terveen, L.: Discovering personal gazetteers: an interactive clustering approach. In: *ACM GIS*, pp. 266–273. ACM, New York (2004)
38. Froehlich, J., Krumm, J.: Route Prediction from Trip Observations. *Society of Automotive Engineers* 2193, 53 (2008)
39. Reddy, S., Burke, J., Estrin, D., Hansen, M., Srivastava, M.: Determining Transportation Modes on Mobile Phones. In: *ISWC* (2008)
40. Khuller, S., Moss, A., Naor, J.: The budgeted maximum coverage problem. *Information Processing Letters* 70(1), 39–45 (1999)
41. Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., Van Briesen, J., Glance, N.: Cost-effective outbreak detection in networks. In: *ACM Knowledge discovery and data mining*, pp. 420–429. ACM Press, New York (2007)
42. Nemhauser, G., Wolsey, L., Fisher, M.: An analysis of approximations for maximizing submodular set functions I. *Mathematical Programming* 14(1), 265–294 (1978)
43. Hsu, W., Dutta, D., Helmy, A.: Mining Behavioral Groups in Large Wireless LANs. In: *ACM Mobicom* (2007)
44. Osborne, J.: Notes on the use of data transformations. *Practical Assessment, Research & Evaluation* 8(6) (2002)
45. Shlens, J.: A tutorial on principal component analysis. Systems Neurobiology Laboratory, University of California at San Diego (December 2005)
46. Deerwester, S., Dumais, S., Furnas, G., Landauer, T., Harshman, R.: Indexing by latent semantic analysis. *American Society for Info. Science* 41(6), 391–407 (1990)
47. Gong, Y., Liu, X.: Video summarization using singular value decomposition. In: *CVPR*, vol. 2. IEEE Computer Society, Los Alamitos; 1999 (2000)
48. Cohen, J.: Privacy, Visibility, Transparency, and Exposure. *University of Chicago Law Review* 75(1) (2008)
49. Nissenbaum, H.: Privacy as Contextual Integrity. *Wash. Law Review* 79(1) (2004)
50. Zimmer, M.: Privacy and Surveillance in Web 2.0: A study in Contextual Integrity and the Emergence of Netveillance. *Society for Social studies of Science* (2007)

Location Diversity: Enhanced Privacy Protection in Location Based Services

Mingqiang Xue, Panos Kalnis, and Hung Keng Pung

Department of Computer Science,
National University of Singapore
`{xuemingq,kalnis,punghk}@comp.nus.edu.sg`

Abstract. Location-based Services are emerging as popular applications in pervasive computing. Spatial k -anonymity is used in Location-based Services to protect privacy, by hiding the association of a specific query with a specific user. Unfortunately, this approach fails in many practical cases such as: (i) personalized services, where the user identity is required, or (ii) applications involving groups of users (e.g., employees of the same company); in this case, associating a query to any member of the group, violates privacy.

In this paper, we introduce the concept of *Location Diversity*, which solves the above-mentioned problems. Location Diversity improves Spatial k -anonymity by ensuring that each query can be associated with at least ℓ different semantic locations (e.g., school, shop, hospital, etc). We present an attack model that maps each observed query to a linear equation involving semantic locations, and we show that a necessary condition to preserve privacy is the existence of infinite solutions in the resulting system of linear equations. Based on this observation, we develop algorithms that generate groups of semantic locations, which preserve privacy and minimize the expected query processing and communication cost. The experimental evaluation demonstrates that our approach reduces significantly the privacy threats, while incurring minimal overhead.

1 Introduction

Nowadays, mobile devices are able to access a variety of Location-based Services (*LBS*) in the pervasive environment. As an example of LBS, a car driver may send query about the route to the nearest gas station when she is driving on the expressway.

A characteristic of such services is that they require the user location to answer queries. This raises serious privacy concerns, since the location can reveal to an attacker (which may be the LBS itself) sensitive information about the user, such as her interests, alternative life style, health conditions, etc. The problem is not solved by simply replacing the user ID with a pseudonym, since the location information can be joined with other public data (e.g., white pages service) to re-identify the user. Therefore, most of the existing work combines pseudonyms with Spatial k -anonymity [5,10,16]. Specifically, between the users and the LBS

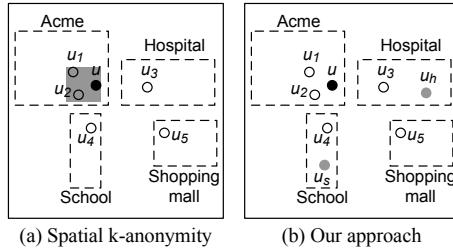


Fig. 1. *k*-anonymity vs. Location Diversity

exists a trusted Anonymizer service that keeps the current locations of all users. When user u wants to send a query, she contacts the Anonymizer, which removes the ID of u and constructs an Anonymizing Spatial Region (*ASR*) that includes u and $k - 1$ other users near u ; the *ASR* is sent to the LBS. The intuition is that by observing the *ASR*, the probability of identifying u as the querying user is at most $1/k$. The LBS answers the query for the entire *ASR* and sends the candidate results to the Anonymizer, which filters the false positives and returns the actual answer to u .

Nevertheless, there are cases where k -anonymity is inadequate. The problem exists because each user location is implicitly associated with semantic information. For example:

Example 1 (Group privacy violation). Acme is an insurance company with a large client list; the list is a valuable business asset and must be kept secret. Acme's employees visit frequently their clients. To plan their trip, they use an LBS (e.g., Google maps) which suggests the fastest routes; due to varying traffic conditions the routes may change over time. Obviously, if the LBS is malicious, it can reconstruct with high probability the entire client list, by observing frequent queries that originate at Acme. To avoid this, queries are issued through an Anonymizer, which implements Spatial k -anonymity. Figure 1.a shows a map of the current users; u is the querying user. The Anonymizer attempts to generate a small *ASR*¹. Assuming $k = 3$ the *ASR* is the gray rectangle that contains $\{u, u_1, u_2\}$. Unfortunately, the *ASR* contains only Acme's employees; therefore, the malicious LBS is sure that the query originated at Acme.

The previous example demonstrates that although k -anonymity is satisfied, the group privacy is violated. This happens because all users inside Acme's headquarters are (with high probability) Acme's employees, therefore they belong semantically to the same group. Motivated by this, we propose the concept of *Location Diversity*, which guarantees that each query can be associated with at least ℓ semantically different locations. This is shown in Figure 1.b, where the Anonymizer sends to the LBS a packet of $\ell = 3$ queries originating at u , u_h and u_s . The semantic locations of these users are Acme, Hospital and School,

¹ Small *ASRs* are desirable for efficient query processing [10].

respectively. Therefore, an attacker cannot identify with probability larger than $1/\ell$ that the query comes from Acme.

A second case involves LBS applications where the user's true identity is required. Such applications (e.g., location-based social networking²) provide the user with personalized services according to her profile. Spatial k -anonymity is not applicable in this case, because it requires pseudonyms. On the other hand, our approach can effectively protect user's location privacy, as shown in the next example.

Example 2 (Personalized services). Assume that u_3 (Figure 1.b) visits frequently a hospital because of her chronic disease. u_3 subscribes to an LBS, which provides a social networking service that allows u_3 to see her friends who are currently near her. However, u_3 believes that hospital is a sensitive location and she does not want the LBS (attacker) to learn her semantic location when she is using the service in the hospital. Obviously, Spatial k -anonymity cannot be used because the true identity of u_3 is needed in order to identify her friends. On the other hand, Location Diversity can be successfully employed. Assuming $\ell = 3$, the Anonymizer sends to the LBS a packet of 3 queries, containing u_3 plus two possible locations from two semantically different sites (e.g., $\{u_3, u_5, u_s\}$). From the query, the LBS thinks that possibly u_3 is in shopping mall, or school, or hospital. Therefore, the LBS can only assume that u_3 is in a hospital with probability at most 1/3.

Location Diversity is similar to ℓ -diversity [15] which is used to anonymize relational data. However, similar idea has never been used in Location-based services. In summary, our contributions are:

- We define the concept of semantic locations and identify cases where Spatial k -anonymity fails to protect privacy. We propose Location Diversity, which protects against such threats by grouping together sets of ℓ semantically different locations.
- We model the attacker's knowledge as a system of linear equations that represent the observed queries. We show that a necessary condition to preserve privacy, is to have infinite solutions for all variables of the equation system.
- Based on our attack model, we develop algorithms that implement Location Diversity while minimizing the query processing cost. Our experimental results show that Location Diversity provides superior privacy protection compared to Spatial k -anonymity without significant computational and communication overhead.

The rest of this paper is organized as follows: Section 2 overviews the related work. Section 3 formulates Location Diversity, whereas Section 4 discusses the attack model. Section 5 presents our anonymization algorithms. The experimental results are presented in Section 6. Finally Section 7 concludes the paper with directions for future work.

² For example, MyLoki: <http://my.loki.com/>

2 Related Work

k -anonymity [17] was initially studied in the context of relational databases, where data must be published without revealing the identity of individual records. Even if the identifying attributes (e.g., *name*) are removed, an attacker may identify specific persons by using combinations of other attributes (e.g., $\langle \text{zip}, \text{sex}, \text{DoB} \rangle$), called quasi-identifiers (*QI*). A table is anonymized if each record is indistinguishable from at least $k - 1$ other records with respect to the *QI*. A common form of anonymization is generalization, which involves replacing specific *QI* values with more general ones (e.g., *city* \rightarrow *state*). Several generalization algorithms have been proposed [13][18] [8].

In some cases k -anonymity is not sufficient. Consider an anonymized database where all tuples in a group have the same value for a sensitive attribute (e.g., *disease*). By associating a person with that group, an attacker will know the disease of that person, even though k -anonymity prevents the association of a specific tuple with the person. ℓ -diversity [15] solves this problem by requiring that each anonymized group contains at least ℓ “well-represented” sensitive attribute values.

Spatial k -anonymity is an adaptation of the relational methods in the context of Location-based Services (*LBS*). The aim in this case is to hide the association between a specific user and a query submitted to the *LBS*. Most approaches follow the architecture of Figure 2, which requires a *trusted* Anonymizer between the user and the *LBS*. The Anonymizer maintains the current location of all users and is responsible for (*i*) hiding the ID of the querying user u (i.e., by using a pseudonym), and (*ii*) replacing the original query with an anonymizing spatial region (*ASR*) that contains u and at least $k - 1$ other users near u ; this process is called *spatial cloaking*.

Several cloaking methods have been proposed. Clique Cloak [5] combines spatial with temporal cloaking. Each query q specifies a temporal interval δt that the corresponding user u is willing to wait. If within δt , the Anonymizer finds $k - 1$ more clients in the vicinity of u that also issue queries, all these queries are combined in a single *ASR*; otherwise, q is rejected. In Interval Cloak [9], on the other hand, the Anonymizer maintains the current locations of all users in a Quad-tree. Once it receives a query from u , it traverses the tree (top-down) until it finds the quadrant that contains u and fewer than $k - 1$ users. Then, it selects the parent of that quadrant as the *ASR*. Casper [16] is also based on Quad-trees. The Anonymizer builds a hash table on the user ID pointing to the lowest-level quadrant where the user lies. Thus, each user is retrieved directly, without having to access the tree top-down. Casper generates smaller *ASRs* compared to Interval Cloak, by considering the quadrants at the same level of the tree before ascending to the parent node. Due to the smaller *ASRs*, the computational and communication cost during query processing are lower.

Interval Cloak and Casper do not guarantee privacy when the distribution of user locations is skewed, since the outlier users are easily exposed [10]. To solve this problem, Kalnis et al. [10] formulated the reciprocity property, which states that, if a user u_j belongs to the *ASR* of u_i , then u_i belongs to the *ASR* of u_j .

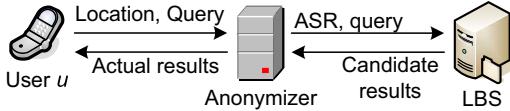


Fig. 2. Architecture for Spatial k -anonymity

They also developed the Hilbert Cloak algorithm, which uses the Hilbert space filling curve [1] to map the 2-D space into 1-D values. The algorithm partitions the 1-D list into groups of k users (the last group may have up to $2k - 1$ users). For a querying user u the algorithm finds the group where u belongs, and returns the minimum bounding rectangle of that group as the ASR. The same ASR is returned for any user in a given group; therefore Hilbert Cloak is reciprocal. A similar method has also been implemented on a Peer-to-Peer system [7]. Chow and Mokbel [4] extend reciprocity to continuous queries involving moving users.

The above-mentioned methods assume that the Anonymizer maintains the current locations of numerous users. However, in practice this is problematic because: (i) Privacy can be compromised if some users are not trustworthy (e.g., the attacker deploys decoys at known locations). (ii) The cost of location updates is high and (iii) it is unlikely that a large number of mobile users will accept such cost in constant basis, for the sporadic benefit of issuing an anonymous query. For these reasons, recent methods do not require an Anonymizer to maintain the user locations. In Probabilistic Cloaking [2] the ASR is a region around the query point. Given an ASR, the LBS returns the probability that each candidate result satisfies the query based on its location with respect to the ASR. Space Twist [18] is another approach where the user sends a fake query, called anchor, to the LBS. The LBS progressively returns the nearest-neighbors of the anchor to the user, who computes her own nearest neighbor from those results. Conceptually, the ASR in this case corresponds to an irregular shape that models the information leak during query processing. Similar to these methods, we also avoid storing the current user locations at the Anonymizer and prefer to send to the LBS fake user locations when necessary.

None of the previous methods considers the semantic locations that are included in the ASR, so they cannot be used in our problem. A possible solution is to encrypt the location of the user, before sending the query to the LBS. This approach is followed by Khoshgozaran and Shahabi [11], who employ the Hilbert transformation together with encryption to conceal both the spatial data and the queries from the LBS; their system supports approximate nearest-neighbor queries. A serious drawback is that all users must be trustworthy. Ghinita et al. [6] provide much stronger privacy by employing Private Information Retrieval [12]. The user does not need to trust anybody (e.g., Anonymizer or other users) and the LBS does not gain any information about the query. Unfortunately, the system supports only 1-Nearest-Neighbor queries; moreover the computational and communication cost are very high.

3 Location Diversity

In Section 1 we mentioned that the location of a user carries contextual information. For example, a user who is currently at a shopping mall is most probably in the process of shopping; a user at a girl’s school is a young female student with high probability. Below is the definition of semantic location, which will be used throughout the rest of the paper:

Definition 1 (Semantic Location). *A semantic location is a region on the map that typically gathers user population that has similar contextual information, such as age, gender, activities, etc. Examples of semantic locations are schools, hospitals, a golf course, the office of a specific company, etc.*

Let SL be the set of all semantic locations (e.g., $SL = \{school, hospital, Acme\}$). Each semantic location may have many instances. We denote by $T(L_i)$, $L_i \in SL$ the set of all instances of semantic locations that belong to type L_i . For example, if $L_i \equiv \text{‘school’}$ and there are three schools on the map, then $T(\text{‘school’}) = \{school_1, school_2, school_3\}$. Obviously, $T(L_i) \neq \emptyset$.

One question is when should we classify two semantic locations into the same category. For example, German and English schools could both be classified under ‘school’, but they could also form two different categories. Such classification is application dependent and is outside the scope of this paper. We assume that the categories of semantic locations are given as a-priori knowledge to both the attacker and the Anonymizer.

Let SQ be the set of all spatial queries submitted to the LBS and let $Q_i \in SQ$. Each Q_i is associated with the semantic location of the querying user. For example, if a user asks Q_i from Acme’s headquarters, then the query is associated with ‘Acme’. The attacker observes all queries; therefore, he can construct estimations about the distribution of each query for each semantic location. Formally, $\forall L_i \in SL$ the attacker estimates the distribution $D_{L_i}(q)$, where q is a random variable taking values from SQ . Through these estimations the attacker may breach privacy; this is demonstrated in the following example:

Example 3. Let $SL = \{L_1, L_2, L_3\}$, $SQ = \{Q_1, Q_2, Q_3\}$ and assume that the attacker observes 100 queries whose distribution is shown in Table 1. The attacker can generate association rules in the form $L_j \Rightarrow Q_i$. Assume $L_1 \equiv \text{‘Acme’}$ and $Q_1 \equiv \text{‘Find the fastest route to 107 River Ave’}$. Since this query is sent from ‘Acme’ with high probability (i.e., 89%) the attacker can assume that the ‘107 River Ave’ address corresponds to an Acme’s client; the name of the client can be found through a white pages service.

Observe that the same query Q_i can be asked from several semantic locations; we denote the set of these locations by QL_i . In the previous example, $QL_1 = \{L_1, L_3\}$ corresponds to query Q_1 . Formally, a location L_j belongs to QL_i if the probability that the query Q_i was asked from L_j is greater than zero. Formally:

Definition 2 (Weak Location Diversity). *A query Q_i exhibits Weak Location Diversity if $|QL_i| \geq \ell$ (i.e., Q_i is associated with at least ℓ semantic locations). ℓ is a system-wide privacy parameter.*

Refer to the example of Figure 1.b: To implement Weak Location Diversity, the querying user u sends her query Q_u to the Anonymizer. As mentioned above, the Anonymizer has a-priori knowledge of the set $\bigcup_{L_i \in SL} T(L_i)$ (i.e., all instances of semantic locations). The Anonymizer selects a set QL_u such that QL_u contains the location of u as well as $\ell - 1$ other semantic locations; in the example, $QL_u = \{Acme, Hospital, School\}$. Each semantic location represents a large spatial region. From each region, the Anonymizer selects a user who represents a query point. Obviously, u is selected from the ‘Acme’ region. For the other two regions, the Anonymizer selects two random points which correspond to fake users³ u_h and u_s . The fake users are selected because the Anonymizer does not maintain the exact user locations (see Section 2). The three query points (i.e., u, u_h, u_s) are packed together and sent to the LBS, which executes three separate queries and returns all results to the Anonymizer. Next, the Anonymizer filters out the false positives and returns the actual result to u .

Weak Location Diversity guarantees privacy only if each semantic location has a single instance (i.e., $\forall L_i \in SL \Rightarrow |T(L_i)| = 1$). In Section 4 we will show that, if the previous condition is not satisfied, the Cross Group Inference Attack can compromise privacy. Motivated by this, we propose *Strong Location Diversity* which is not vulnerable to the Cross Group Inference Attack.

Definition 3 (Strong Location Diversity). *An anonymization method exhibits Strong Location Diversity if, for every query $Q_i \in SQ$ and every semantic location $L_j \in SL$, the probability of associating Q_i with L_j is at most $1/\ell$.*

If a method satisfies the Strong Location Diversity property, then every query anonymized by that method satisfies Weak Location Diversity; the opposite is not true. As a consequence, it is easy to find a solution for the Weak version; the only requirement is that there exist at least ℓ different semantic locations (i.e., $|QL_i| \geq \ell$). In contrast, depending on the number of instances per semantic location (i.e., $|T(L_i)|$), there are cases where no solution exists for Strong Location Diversity. In Section 5 we develop algorithms that generate solutions in between the two extremes.

Table 1. Queries collected by the attacker

Location	Query	Count	Rule	Conf.	Sup.
L_1	Q_1	40	$L_1 \Rightarrow Q_1$	89%	40%
L_1	Q_2	10	$L_1 \Rightarrow Q_2$	50%	10%
L_2	Q_2	10	$L_2 \Rightarrow Q_2$	50%	10%
L_2	Q_3	10	$L_2 \Rightarrow Q_3$	29%	10%
L_3	Q_1	5	$L_3 \Rightarrow Q_1$	11%	5%
L_3	Q_3	25	$L_3 \Rightarrow Q_3$	71%	25%

³ In general, this does not raise any privacy issue, unless the attacker knows the actual locations of all users; however, we believe that such an assumption is unrealistic.

4 Cross Group Inference Attack

In this section, we show an attack (*CGIA*) which demonstrates that the Weak Location Diversity may not always guarantee privacy.

Remember that in our definition of semantic location, there can be multiple instances of the same semantic location. For example, several branches of the insurance company Acme (as in example 1) located at different places in the city are all instances of the Acme. Since these branches belong to the same company, it is possible that they share the same client list. Compare to other instances of other semantic locations, these branches tend to send similar queries because of the same client list. In the worst case, they produce very similar query patterns in long term. Remember that the client list is the secret business asset to Acme, and the query patterns at any branches would help the attacker to reconstruct the client list. In the following, we present a worst case analysis on the information gain of the attacker under the worst case assumption that instances of the same semantic location produce very similar query patterns.

Consider the example of Figure 3.a, where $SL = \{A, B, C, D\}$ is the set of semantic locations. $T('A') = \{a_1, a_2, a_3\}$, $T('B') = \{b_1, b_2\}$, $T('C') = \{c_1, c_2\}$ and $T('D') = \{d_1\}$ are the instances of the corresponding semantic locations. The figure shows the groups that are used for anonymization. For example group $G_1 = \{a_1, b_1\}$ indicates that, if the querying user is in the region of a_1 or b_1 , then the anonymizer will construct a query packet that includes fake users from a_1 and b_1 . Observe that all groups contain two distinct semantic locations, therefore they satisfy Weak Location Diversity for $\ell = 2$. The intuition of CGIA is that, although the attacker cannot infer any information from a single group, he is able to reconstruct the query patterns of individual semantic locations by examining the query history.

Let $Q \in SQ$ be a particular query of the attacker's interest. By the definition of semantic locations, we expect that the query distribution in all instances of a semantic location is similar. Under the worst case assumption, the number of Q asked from the instances of the same semantic location are the same. For example, a_1, a_2, a_3 each asks the same number of Q in the worst case. Specifically,

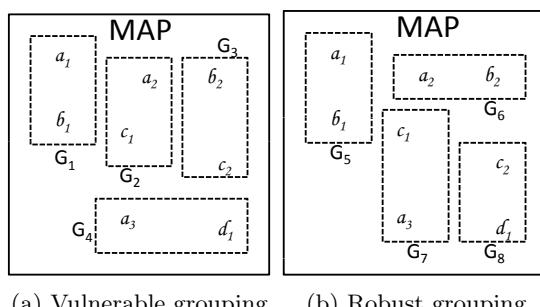


Fig. 3. Cross group inference attack

let A^Q be the number of times that Q was asked from an instance of semantic location A (i.e. a_1 or a_2 or a_3). Similarly, we define B^Q , C^Q and D^Q for the rest of the semantic locations. Moreover, let G_i^Q be the total number of times that Q was asked from all semantic locations of group G_i . Then, from Figure 3.a we form the following system of linear equations:

$$\begin{cases} A^Q + B^Q = G_1^Q = z_1 \\ A^Q + C^Q = G_2^Q = z_2 \\ B^Q + C^Q = G_3^Q = z_3 \\ A^Q + D^Q = G_4^Q = z_4 \end{cases} \quad (1)$$

The system has a unique solution:

$$\begin{cases} A^Q = \frac{z_1+z_2-z_3}{2} \\ B^Q = \frac{z_1-z_2+z_3}{2} \\ C^Q = \frac{z_2-z_1+z_3}{2} \\ D^Q = \frac{2z_4-z_1-z_2+z_3}{2} \end{cases} \quad (2)$$

Since the attacker knows $z_{1\dots 4}$ (by observing the query history), he can calculate A^Q , B^Q , C^Q , and D^Q . Thus, he can identify the probability density function of query Q for all semantic locations. By repeating the process for other queries in SQ , the attacker can construct a table similar to the one of Example 3 (see Section 3). Therefore, the grouping of Figure 3.a is vulnerable under CGIA, although Weak Location Diversity is satisfied.

A different grouping is shown in Figure 3.b. Using the same methodology, we obtain the following system:

$$\begin{cases} A^Q + B^Q = G_5^Q = z'_1 \\ A^Q + B^Q = G_6^Q = z'_1 \\ A^Q + C^Q = G_7^Q = z'_2 \\ C^Q + D^Q = G_8^Q = z'_3 \end{cases} \quad (3)$$

The solution is:

$$\begin{cases} A^Q = x, x \in \mathbb{N} \\ B^Q = z'_1 - x \\ C^Q = z'_2 - x \\ D^Q = z'_3 - z'_2 + x \end{cases} \quad (4)$$

where x is a free variable. Since A^Q , B^Q , C^Q and D^Q all depend on x , the attacker cannot calculate the number of times that Q is associated with a specific location. Therefore, the grouping of Figure 3.b is robust against CGIA.

In general, consider a map with N groups G_1, G_2, \dots, G_N and let $L_i \in SL$, where SL is the set of semantic locations. The attacker can construct a system of N linear equations for every query $Q \in SQ$. The form of the j -th equation that corresponds to a query Q is:

$$\begin{aligned} \sum_{i=1}^{|SL|} h \cdot L_i^Q &= G_j^Q, 1 \leq j \leq N \\ h &= \begin{cases} 1 & \text{if } L_i \in G_j \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (5)$$

In a consistent system of linear equations, if the rank of the system is the same as the number of variables, then there is a unique solution. In this case, the grouping is vulnerable to CGIA. On the other hand, if the rank is smaller than the number of variables, then the system has infinite number of solutions. However, there are cases where the attacker can still compromise privacy, because some of the variables may have unique solutions. Consider the following example:

$$\left(\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{array} \right) \quad (6)$$

The above is the reduced row-echelon form of a linear system after Gauss-Jordan elimination. Although the system has an infinite number of solutions, the second row has only one non-zero entry; therefore the second variable has a unique solution, rendering the grouping vulnerable to CGIA. Based on this observation, we propose the following definition:

Definition 4 (Robust system). *A system of linear equations is robust if every equation in the reduced row-echelon form has two or more non-zero entries.*

If a grouping generates a robust system, then it is guaranteed that the attacker cannot calculate the exact number of queries for any semantic location. Nevertheless, the following example shows that, even with a robust system, the attacker can gain *probabilistic* information about the query distribution:

5 Implementing Location Diversity

5.1 Assumptions about the Attacker

We assume that in the worst case the attacker is the LBS; obviously, in this case the attacker has complete knowledge about the map. The attacker will also keep the history of all user queries and observe the incoming queries. We assume that the attacker is semi-honest, which means that he is only curious in gaining privacy information from the history of user queries, but not cheat the users in query answering. The attacker has the following goals: 1) Infer the semantic location where a query is generated from. 2) Learn the query pattern of a semantic location, i.e., $D_{L_i}(q)$ for a semantic location under category L_i .

5.2 System Architecture

The system utilizes the same architecture as in spatial k -anonymity (see Figure 2). Instead of generating ASRs the Anonymizer injects Location Diversity

Algorithm 1 Grouping Algorithm

```

1: sortCatList(catList)
2: length  $\leftarrow 0$ , width  $\leftarrow 0$ 
3: for  $i \leftarrow L - 1$  to 0 do
4:   if catList[ $i$ ].size = 0 then
5:     continue
6:   end if
7:   length  $\leftarrow i + 1$ , width  $\leftarrow \text{catList}[i].size$ 
8:   break
9: end for
10: if width = 0 and length = 0 then
11:   return
12: else
13:   for  $i \leftarrow 0$  to width − 1 do
14:     new eqn
15:     for  $j \leftarrow 0$  to length − 1 do
16:       eqn.add(catList[j].remove(0))
17:     end for
18:     eqnList.add(eqn)
19:   end for
20:   go to line 1
21: end if

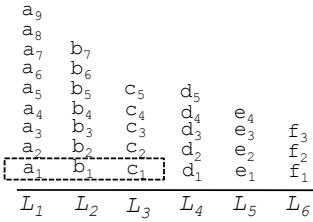
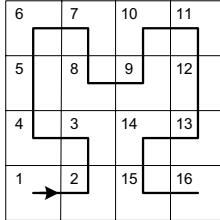
```

into the queries. The Anonymizer will send ℓ or more point queries to the LBS server instead of a single query with an ASR region. The Anonymizer needs to keep the semantic information of the map. Before it can provide Anonymization service to the users, it runs the grouping algorithm so that groups of semantic locations can be formed.

5.3 Grouping Algorithms

The grouping algorithms form groups among semantic locations so that each group contains at least ℓ semantic locations from different categories. As we have shown in Section 4, certain groups are vulnerable to CGIA. A good grouping algorithm should take the attacker’s capability into consideration, and form groups that result to robust system of linear equations. Our algorithm is doing so by finding maximum number of common groups in the map. Common groups refer to the groups that have members from common categories. Since each common group is a repetition of another, it gives no extra information to the attacker. In terms of forming system of linear equations, this methodology finds common equations that can be canceled during the reduction (e.g., Gauss-Jordan elimination), so that the rank of the system can be minimized.

Algorithm 1 depicts how the grouping algorithm works. We use $T(L_i)$ to represent the vector structure that enumerates all the instances under category L_i . We store all $T(L_i)$ in an array *CatList*. In each loop, we first sort *CatList* in descending order according to the number of instances in each $T(L_i)$. Then

**Fig. 4.** Grouping algorithm**Fig. 5.** $2^2 \times 2^2$ and $2^3 \times 2^3$ Hilbert curves

we take one instance from each of the ℓ first $T(L_i)$ in the array and form a group. Once the group is formed, we remove the corresponding semantic location instances from $T(L_i)$. We label this group as the leading group. We continue to check if the instances from the first ℓ $T(L_i)$ could form a common group with the leading group. If this is true, we form a new group. We continue adding groups until no common group with the leading group can be found; then we go back to the beginning of the loop. The loop stops when all $T(L_i)$ are empty. It is possible that the *CatList* has less than ℓ non-empty $T(L_i)$ left, meaning that we cannot form a group with semantic locations from ℓ categories. If it this case, we form a group with as many different semantic locations as possible, and we combine it with another group that already has ℓ distinct semantic locations. In this way all groups satisfy the Location Diversity property. Figure 4 is the visualization of *CatList* at the beginning. a_i to f_i are instances of categories L_1 to L_6 respectively. The *CatList* is in sorted order. With $\ell=3$, $\{a_1, b_1, c_1\}$ in the box of dotted line forms the leading group. Next, four more groups are generated: i.e., $\{a_2, b_2, c_2\}$, $\{a_3, b_3, c_3\}$, $\{a_4, b_4, c_4\}$ and $\{a_5, b_5, c_5\}$; these groups contain instances from the same semantic locations as the leading group. After these four groups have been found, the algorithm sorts *CatList* again and enters another loop.

5.4 Minimization of Group Area

We prefer to form groups whose members are closer to each other, as geographically closer semantic locations tend to return common answers. Finding the optimal grouping is difficult, as it is a typical combinatorial problem. In the following, we introduce two heuristics to the grouping algorithm that aim to minimize the distance between group members by swapping group memberships of semantic locations that are in the same category while preserving Location Diversity. We use the area of the Minimum Bounding Region (*MBR*) that contains all members of a group as the metric of the relative distances between members, i.e., smaller MBR implies better grouping.

5.5 Greedy Optimization

Suppose *CatList* is in the sorted order, and *CatList*[j] refers to the j -th ($j \geq 0$) element of *CatList*. The greedy optimization can be applied when the grouping

Algorithm 2 Greedy Optimization

```

1: function greedySelect(catList, width, length)
2: for  $i \leftarrow 0$  to  $width - 1$  do
3:   new eqn
4:   for  $j \leftarrow 1$  to  $length - 1$  do
5:      $minArea \leftarrow +\infty$ ,  $minIndx \leftarrow -1$ 
6:     for  $k \leftarrow 0$  to  $catList.size - 1$  do
7:        $area = computeArea(catList, i, j, k)$ 
8:       if  $area < minArea$  then
9:          $minArea \leftarrow area$ ,  $minIndx \leftarrow k$ 
10:      end if
11:    end for
12:    eqn.add(catList[j].get(minK))
13:    catList[j].remove(minK)
14:  end for
15:  eqnList.add(eqn)
16: end for

```

algorithm is finding a common group with the leading group. In greedy optimization, the first member of the new group is chosen to be the first instance in *CatList[0]*. Subsequently, the $(j + 1)$ -th member of the new group is chosen from *CatList[j]* which forms smallest area of MBR with the 1-st to the j -th members in the group. For example, in Figure 4, a_1 is selected as the first member of the new group, whereas b_1 is selected only if it forms smallest MBR with a_1 among all the b_i . Similarly, c_1 is selected only if it forms smallest MBR with a_1 and b_1 among all the c_i . The pseudo code of the greedy optimization is given in Algorithm 2. The time complexity of the grouping algorithm with Greedy optimization is $O(N^2)$, where N is the number of instances of semantic locations.

5.6 Hilbert Optimization

Another algorithm with better time complexity is Hilbert optimization. The Hilbert optimization is based on 2-dimensional Hilbert curve (see Figure 5). One of the properties of Hilbert curve is that, if two points are close to each other in multi-dimensional space, they also tend to be close in the one dimensional space generated by the Hilbert curve. With Hilbert curve, we can divide a square map into $2^n \times 2^n$ number of squares, where n is called the level of Hilbert curve. For example, in Figure 5(left), a square map with the level 4 Hilbert curve is divided into $2^2 \times 2^2$ squares. Following the Hilbert walk, each of the square is assigned an ID with increasing number. By making use of Hilbert curve, we can sort all the semantic locations under the same category in increasing order of their ID before running the grouping algorithm. For example, in Figure 4 each column is sorted according to Hilbert order. Large Hilbert level will result in smaller squares in the map. Imagine when the Hilbert level is infinitely large each of the

square becomes a point in the map, the accuracy will be improved. The time complexity of grouping algorithm with Hilbert optimization is $O(N)$.

6 Experimental Evaluation

In this section we evaluate the performance of our algorithms. All experiments were conducted on a Intel-Duo 2.33GHz CPU with 3.25GB RAM and Windows XP. Each experiment was run 10 times; the figures show the average values of the results. We used a variety of synthetic datasets; their parameters are explained in the corresponding experiments. We also used a real dataset consisting of points of interest (POIs) in California⁴. The dataset is classified in 63 categories (i.e., set of semantic locations SL) and contains 104,770 landmarks (i.e., instances of semantic locations). Examples of such semantic locations are airport, church, dam, park, and etc. Each semantic location is defined by a pair of longitude and latitude values. Another useful resource of classifications of locations can be found in NAICS⁵ website.

6.1 Performance of Grouping Algorithms

In the first experiment we measured the average area of the anonymization groups. Recall that we prefer groups with small area, since they incur less query processing time and the number of candidate results (which is proportional to the communication cost) is smaller. As a base-line competitor, we implement an algorithm (call *Random*) that selects randomly the semantic locations of each group.

First we used synthetic data, where we generated randomly the semantic locations in the map. In Figure 6.a we fixed $\ell = 10$ and vary the number of groups of sensitive locations between 1,000 and 8,000. Both Hilbert and Greedy create groups with much smaller areas than Random. In Figure 6.b we set $|SL| = 1,000$ and vary ℓ ; the trends are similar to the previous graph. We also performed the same experiments with the California dataset. The results are shown in Figure 6.c; the trends are similar to those of the synthetic data.

Using the random data, we measured the time to generate the groups. Figures 6.d and 6.e show that the running time for Hilbert tends to be almost unaffected by number of groups and ℓ . Greedy, on the other hand, grows roughly quadratically with number of groups and ℓ . Therefore, there is a tradeoff between the two algorithms: Greedy generates groups with slightly smaller area, but it is considerably slower than Hilbert.

6.2 Query Processing Cost at the LBS

In this set of experiments we evaluated the impact of the grouping algorithms on query processing. Again, we used the synthetic maps, where we generated

⁴ Available at: <http://www.cs.fsu.edu/~lifeifei/SpatialDataset.htm>

⁵ Available at <http://www.census.gov/eos/www/naics/>

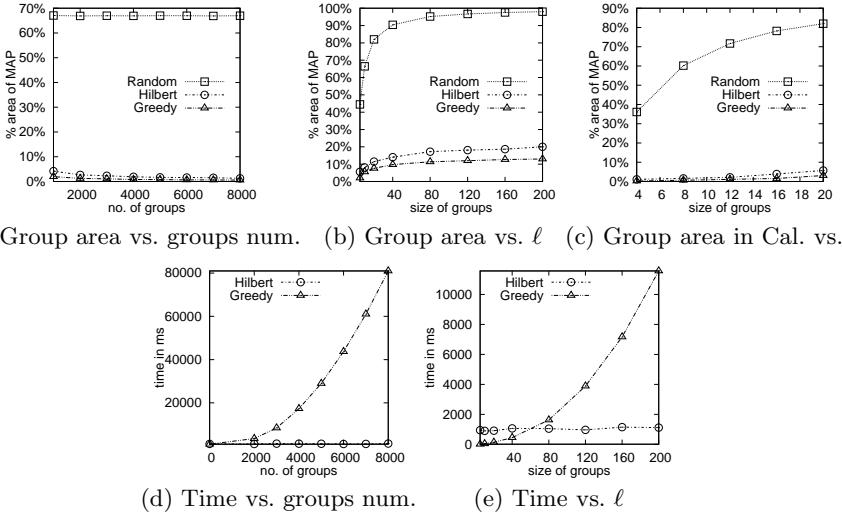


Fig. 6. Group area comparison and California POIs

2,600 POIs; the POIs are indexed by an R*-tree. The LBS implements nearest-neighbor queries; for each request, it returns the 10 nearest POIs. Figures 7.a and 7.b show the results for varying number of groups and number of POIs, respectively. We observe that the number of answers that correspond to Hilbert and Greedy groupings is significantly smaller than those for Random; the number of answers is proportional to the communication cost. A similar trend is observed in Figure 7.c, where we vary ℓ . Moreover, Figure 7.d shows the I/O cost (i.e., number of disk accesses) for varying ℓ ; again the trend among grouping algorithms is the same. This is because the number of distinct answers is typically proportional to the number of I/Os.

6.3 Evaluation of Privacy Guarantees

The following experiments evaluate the privacy guarantees of our algorithms. We focus on the rank and the number of non-zero entries (NZE) in the reduced row-echelon form of the linear system, since they characterize the ability of the attacker to find a good estimation of the real associations of queries with semantic locations. We randomly generated $|SL| = 80$ semantic locations; the number of instances for each semantic location was picked randomly and varies between 0 and 400. Using our grouping algorithms, a system of linear equations with 80 variables is generated. Thus, the maximum number of NZEs in a row and the rank, cannot exceed 80. Figure 7.e shows that by using our grouping algorithms, the rank of the linear system becomes significantly less than 80, and only increases slowly with ℓ (low rank corresponds to better privacy). Moreover, the average number of NZEs per equation is always above 2, which is the threshold to the privacy violation. In Figure 7.f we fix $\ell = 6$ and vary $|SL|$. Observe that

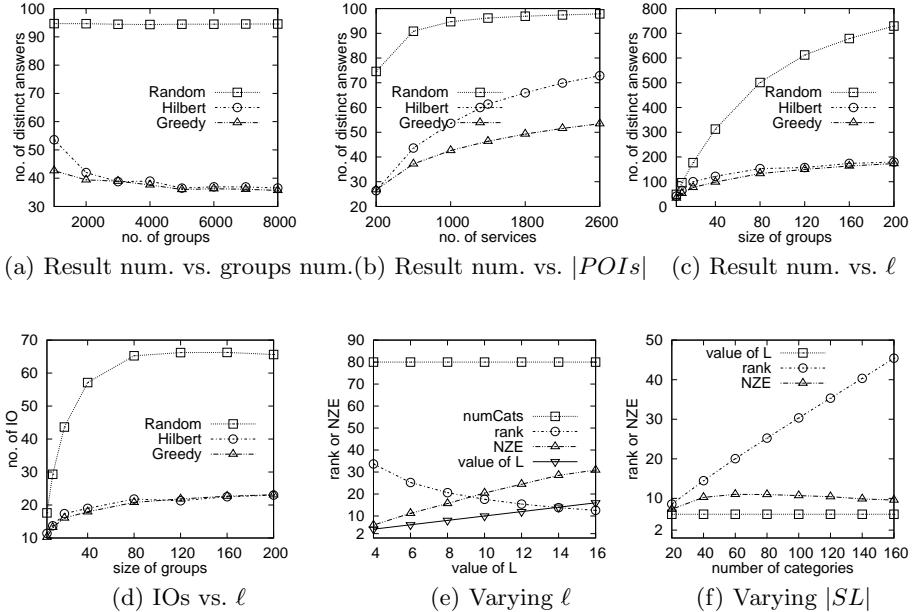


Fig. 7. Query processing cost and privacy

although the rank increases, it is still much smaller than the number of variables (by definition the number of variables is equal to $|SL|$). Also, the average number of NZEs is always above 2.

6.4 Worst-Case Privacy Evaluation

The previous graphs indicate only the average privacy guaranty of our grouping algorithms. Here, we investigate the worst case performance. In Figure 8.c we consider different combinations of ℓ and $|SL|$; for each such combination, we randomly form 10,000 systems of linear equations. The table shows the number of compromised systems (i.e., systems that contain at least one equation with a single NZE). Only 2 such cases were found; this is an indication of the robustness of our algorithms.

6.5 Comparison against k -Anonymity

In the last experiment we compare Location Diversity with Spatial k -anonymity. We implemented two recent k -anonymity algorithms, namely Hilbert Cloak (HC) and Nearest Neighbor Cloak (NNC) [10], and we used the California dataset. Users are randomly generated around each semantic location; each user belongs to the semantic location nearest to her current location. We measure diversity as the number of distinct semantic locations in a group, over the total number of semantic locations in the same group; we take the average of all groups.

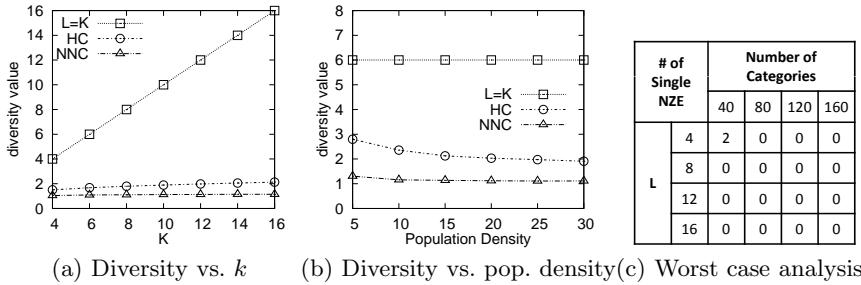
**Fig. 8.** Comparison against k -anonymity and worst case analysis

Figure 8.a depicts the diversity value for varying k . The line $\ell = k$ corresponds to the ideal case (i.e., ℓ distinct semantic locations per group). Our algorithms are always identical to this line; for clarity, they are not shown in the graph. On the other hand, the diversity value of HC and NNC is much lower than the ideal one. Similarly, in Figure 8.b we fix $k = \ell = 6$ and vary the user population density. We found that the diversity value of HC and NNC decreases. This result is expected because, as the population increases, HC and NNC can find groups with smaller cloaking areas.

7 Conclusions

In this paper, we focused on privacy in location-based services and identified two privacy threats (i.e., group privacy violation and privacy in personalized services), which are not handled by existing Spatial k -anonymity algorithms. We proposed the concept of Location Diversity, which solves these problems. We also showed that Location Diversity is not equivalent to relational ℓ -diversity, since the latter is vulnerable to the cross group inference attack. Finally, we developed algorithms for Location Diversity.

Our work is an initial step towards solving the above problems, and faces some limitations. The most important one is that it assumes a static set of sensitive locations. Therefore, it may report that a fake user is in a library at middle night which is unlikely to be true in reality. Moreover, a solution that supports location diversity without using a trusted anonymizer is also interesting for research. These issues will be targeted in our future work.

References

1. Butz, A.R.: Alternative Algorithm for Hilbert's Space-Filling Curve. In: Trans. on Computers
2. Cheng, R., Zhang, Y., Bertino, E., Prabhakar, S.: Preserving user location privacy in mobile data management infrastructures. In: Intl. Workshop on Privacy Enhancing Technologies

3. Chin, F., Ozsoyoglu, G.: Auditing and Inference Control in Statistical Databases. *Trans. on Software Engineering*
4. Chow, C.-Y., Mokbel, M.F.: Enabling Private Continuous Queries for Revealed User Locations. In: Papadias, D., Zhang, D., Kollios, G. (eds.) SSTD 2007. LNCS, vol. 4605, pp. 258–275. Springer, Heidelberg (2007)
5. Gedik, B., Liu, L.: Location Privacy in Mobile Systems: A Personalized Anonymization Model. In: Proc. of ICDCS
6. Ghinita, G., Kalnis, P., Khoshgozaran, A., Shahabi, C., Tan, K.-L.: Private Queries in Location Based Services: Anonymizers are not Necessary. In: Proc. of SIGMOD
7. Ghinita, G., Kalnis, P., Skiadopoulos, S.: PRIVE: Anonymous Location-based Queries in Distributed Mobile Systems. In: Proc. of WWW
8. Ghinita, G., Karras, P., Kalnis, P., Mamoulis, N.: Fast Data Anonymization with Low Information Loss. In: Proc. of VLDB
9. Gruteser, M., Grunwald, D.: Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In: Proc. of USENIX MobiSys
10. Kalnis, P., Ghinita, G., Mouratidis, K., Papadias, D.: Preventing Location-Based Identity Inference in Anonymous Spatial Queries. In: TKDE
11. Khoshgozaran, A., Shahabi, C.: Blind Evaluation of Nearest Neighbor Queries Using Space Transformation to Preserve Location Privacy. In: Papadias, D., Zhang, D., Kollios, G. (eds.) SSTD 2007. LNCS, vol. 4605, pp. 239–257. Springer, Heidelberg (2007)
12. Kushilevitz, E., Ostrovsky, R.: Replication is NOT needed: Single database, computationally-private information retrieval. In: Symp. on Foundations of Computer Science
13. LeFevre, K., DeWitt, D.J., Ramakrishnan, R.: Mondrian Multidimensional k-Anonymity. In: Proc. of ICDE
14. Li, N., Li, T., Venkatasubramanian, S.: t-Closeness: Privacy Beyond k-Anonymity and l-Diversity. In: Proc. of ICDE
15. Machanavajjhala, A., Gehrke, J., Kifer, D., Venkitasubramaniam, M.: l-Diversity: Privacy Beyond k-Anonymity. In: Proc. of ICDE
16. Mokbel, M.F., Chow, C.Y., Aref, W.G.: The New Casper: Query Processing for Location Services without Compromising Privacy. In: Proc. of VLDB
17. Sweeney, L.: k-Anonymity: A Model for Protecting Privacy. *Intl. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*
18. Xu, J., Wang, W., Pei, J., Wang, X., Shi, B., Fu, A.W.-C.: Utility-Based Anonymization Using Local Recoding. In: Proc. of KDD
19. Yiu, M.L., Jensen, C.S., Huang, X., Lu, H.: SpaceTwist: Managing the Trade-Offs Among Location Privacy, Query Performance, and Query Accuracy in Mobile Services. In: Proc. of ICDE

Nonvisual, Distal Tracking of Mobile Remote Agents in Geosocial Interaction

Steven Strachan and Roderick Murray-Smith

¹ Orange Labs - France Telecom
28 Chemin du Vieux Chne, 38240 Meylan, France
steven.strachan@gmail.com
² University of Glasgow,
Department of Computing Science,
Glasgow, Scotland, UK
rod@dcs.gla.ac.uk
<http://www.dcs.gla.ac.uk/~rod/>

Abstract. With the recent introduction of mass-market mobile phones with location, bearing and acceleration sensing, we are on the cusp of significant progress in location-based interaction, and highly interactive mobile social networking. We propose that such systems must work when subject to typical uncertainties in the sensed or inferred context, such as user location, bearing and motion. In order to examine the feasibility of such a system we describe an experiment with an eyes-free, mobile implementation which allows users to find a target user, engage with them by pointing and tilting actions, then have their attention directed to a specific target. Although weaknesses in the design of the tilt-distance mapping were indicated, encouragingly, users were able to track the target, and engage with the other agent.

1 Introduction

With the recent introduction of mass-market mobile phones such as the Nokia 6210 *Navigator* with location, compass bearing and acceleration sensing, we are on the cusp of significant potential progress in location-based interaction, and mobile social networking (Fröhlich *et al.* 2008). Currently this sensor group is primarily marketed for pedestrian navigation (Strachan *et al.* 2007, Jones *et al.* 2008), but it has the potential to be used for a much more exciting range of interaction styles. Strachan and Murray-Smith (2009) have described bearing-based interaction with content and services, and in linked work Robinson *et al.* (2008) describe its use for GeoBlogging, or ‘Googling the real world’, as one of their participants observed. It is also an obvious step to couple this with social networking applications, where users can probe and point at and engage with nearby friends (Strachan and Murray-Smith 2008). The richness of the sensing, and the context-sensitivity and person-specific nature of such communications suggest that designers should beware of implementing overly prescriptive mechanisms for allowing individuals to interact in such systems. We argue in this paper that representations which display the uncertainty in location, bearing and inferred context are necessary for the success of such systems, and that this allows performance to grow

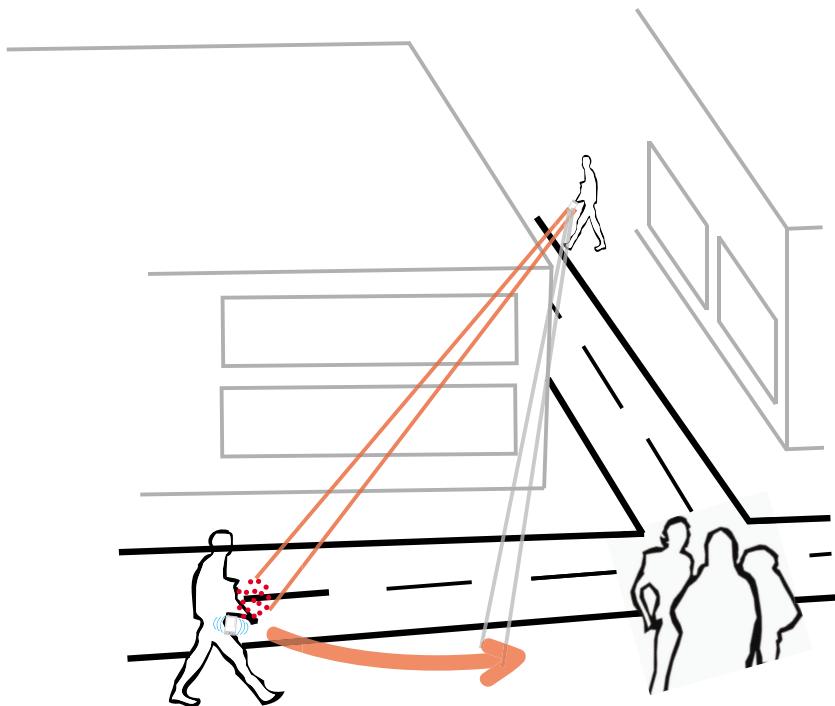


Fig. 1. A user interacts in the virtual environment with a friend, attempting to guide her to the group of friends already assembled. The interaction is primarily physical and non-visual, pointing with the phone with audio and haptic feedback as a function of the movements of the individuals.

over time as new sensing technologies and better context inference mechanisms are developed. This assumption needs to be tested empirically, and the research community needs to evolve methods for reliably comparing competing designs for such geosocial tasks. This paper introduces and describes a system, with an initial user study, which examines the interaction between a human using embodied bearing-based interaction with feedback generated by a simulated model of a human, testing whether it is in fact possible for users to track others in the virtual environment with realistic sensing conditions.

2 Mobile Spatial Interaction

Mobile Spatial Interaction (MSI) is a form of interaction that enables users to interact with a hybrid physical/virtual environment using their mobile device. Users are given the ability to interact, in an eyes-free manner, with the virtual environment by using their mobile device to focus on an object in the real world or scan the space around them to discover virtual objects using pointing motions as illustrated in figure 2. The direction in which the user is pointing is taken from a compass heading estimated using magnetometers with accelerometers used to compensate for tilt.

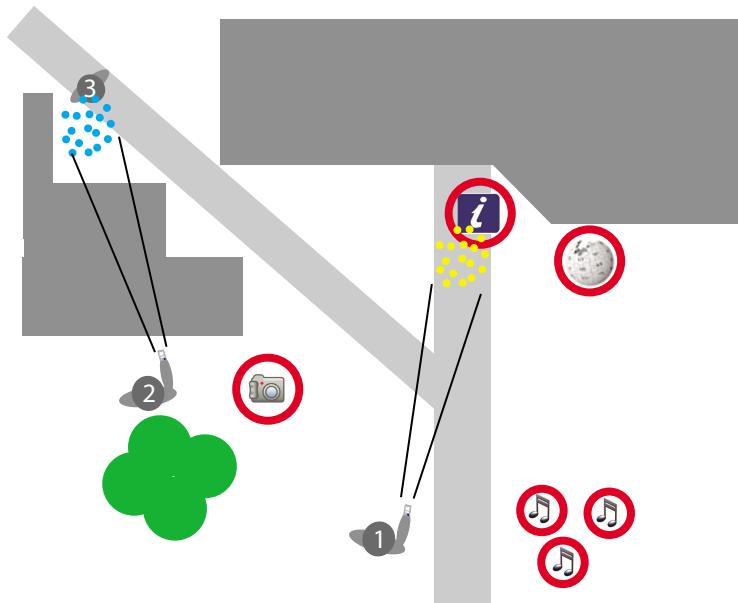


Fig. 2. User 1 is interacting with information in their combined virtual/physical environment while user 2 is interacting with user 3 out of their line of sight

2.1 Virtual Environments

A fluid and unrestricted collaboration between two or more people connected remotely via a computer has long been a goal in the fields of virtual and augmented reality. Collaborative Virtual Environments (CVEs) (Benford *et al.* 2001) enable a sense of shared space and physical presence in the virtual world. The increasing power and ubiquity of continually connected, location-aware and continuously sensing mobile devices has now enabled us to generalise down to the mobile realm with the development a Mobile Collaborative Virtual Environment (MCVE). We present in this paper an MCVE that enables the connection of two or more mobile devices to create a hybrid ‘eyes-free’ physical/virtual world in which users may interact using their mobile device as a probe for objects or for other users located in the virtual environment, while all the time receiving audio and vibrotactile feedback dependent on the nature of their probing. A key aspect to the success of this kind of interaction is the provision of a sense of embodiment or presence in the virtual environment. Greenhalgh and Benford (1995) tackle this with the DIVE and MASSIVE systems by providing a number of graphical representations of embodied participants. The major advantage that these systems have is that they are highly visual and significant emphasis is placed on the provision of visual feedback to the user. Much of the work conducted on eyes-free systems is for the visually impaired. Magnusson and Rassmus-Gröhn (2005) describe a haptic-audio system designed to guide visually impaired users through a traffic environment for exploring and learning a route. They find that most users were able to navigate a fairly complex virtual model with little trouble. Crossan and Brewster (2006) describe a system for

two-handed navigation in a haptic virtual environment designed for visually impaired users finding that participants were able to traverse a maze using haptic feedback alone.

Social Cognition in the Virtual Environment. One of the major functions of social cognition in humans is to allow the creation of a shared world in which interaction can take place. Communication between two or more people is greatly enhanced by the adoption of a shared vocabulary that enables us to share goals, so that we may then engage in joint activity. For successful interactions to take place it is necessary that the interactors achieve the same perception of the world, referred to as ‘common ground’ (Clark 1996). While this is not easy to achieve in a completely abstracted virtual environment, in a hybrid virtual/physical environment this is slightly easier to achieve. Since an MCVE is located in the real world the user is not completely immersed in the virtual world, they have access to real-world visual cues and so some of this natural intuition regarding the interaction with the physical world may be transferred into the virtual world. The augmentation of real world objects is one approach to providing a more effective interaction with the virtual world. Espinoza *et al.* (2001) describe their GeoNotes system that allows users to leave virtual messages linked to specific geographical positions. They strive here to socially enhance digital space by blurring the boundary between physical and digital space. But still little has been achieved in terms of active interaction or collaboration between two or more users in this kind of environment and it remains a challenge. The starting point for this kind of active and collaborative interaction is to align the focus of our attention in the environment, typically achieved in real life by pointing at an object or watching bodily movements that can give us some idea about the intentions of the other person (Frith and Frith 2006). Our bodies are used to provide continuous and fine-grained social cognitive signals about our psychological state, our presence, activity and our attention via gestures, facial expressions or general body posture. It is important then that this kind of information is not lost completely in the virtual environment.

3 Description of Experimental System

The system described here builds on earlier work (Strachan and Murray-Smith 2009), such that the location is provided by the GPS sensor and bearing by a combination of magnetometer and accelerometer. This means that we can point in any direction with the device at any orientation. The orientation (pitch and roll) of the device is estimated using the x , y and z components of the acceleration with respect to the gravity vector, and the pitch angle is used to control how far into the distance the user is pointing with the ‘virtual probe’, illustrated in figure 2. If the user wishes to look further ahead into the space in front they tilt the device forward. If they wish to bring their probe back to their current position they tilt the device back again, effectively pulling the device to their chest as illustrated in figure 3. The actual distance looked ahead is linearly mapped to the pitch angle of the device with a 90° pitch mapping to a 0m look-ahead and 0° pitch mapping to a 30m look-ahead. A user now has the ability to obtain information about the space around them by listening and feeling for impact events (via audio and haptic feedback from the device), when their ‘virtual probe’ effectively collides with objects in the virtual environment.

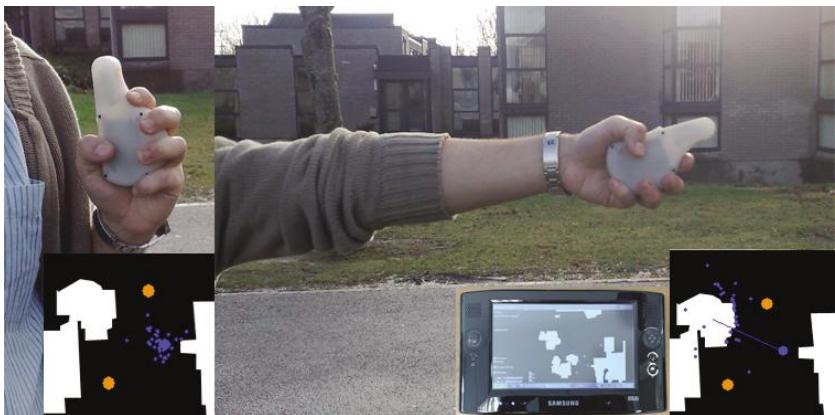


Fig. 3. Varying the orientation of the device alters the distance at which the user is probing

One further feature of this system is that it allows an agent to interact remotely with other agents in the virtual world by pointing and interacting with their mobile devices as illustrated in figure 2. For the purposes of this study interaction is limited to discrete probing of the other agent's device but it is possible in future iterations to expand this to include probing of the other user's movements giving more information about the specific areas of interest and intentions of the other user.

3.1 Uncertainty and the Virtual Probe

With this kind of continuously sensing system, uncertainty becomes an important factor to consider. We are subject to uncertainty in our GPS position estimate, uncertainty in the signals from our sensors, uncertainty in the user's motor actions, uncertainty in the intentions of the user and uncertainty in the system's belief about the user's intention that is feedback to the user. If these uncertainties are not considered and dealt with appropriately they have the potential to render a system of this kind unusable.

The system described in this paper explicitly uses the display of uncertainty to assist a user's navigation of their current location or context by acknowledging all of the uncertainties mentioned above. A straightforward propagation of our user's position through the space in front would lead to a certain point at some specified time horizon (figure 4: Left). This does not model the user's potential future positions effectively and is likely to lead to confusion from the user when uncertainty in the position estimate, for example, suddenly increases and the seemingly certain estimate is now wrong. For example, if a user is moving North at a pace of 5m/s, in 1 minute in a completely certain system he will be exactly 300m North of his current position. But if we take into account any uncertainties, in his initial position, in his heading, in his intentions and any constraints that lie in his path (such as buildings or busy roads) we are much less certain about where he will be in 1 minute (figure 4: Center, Right). In this case we know he will be roughly 300m away but there will be some distribution or uncertainty around that position which increases as we look further ahead in time. By projecting

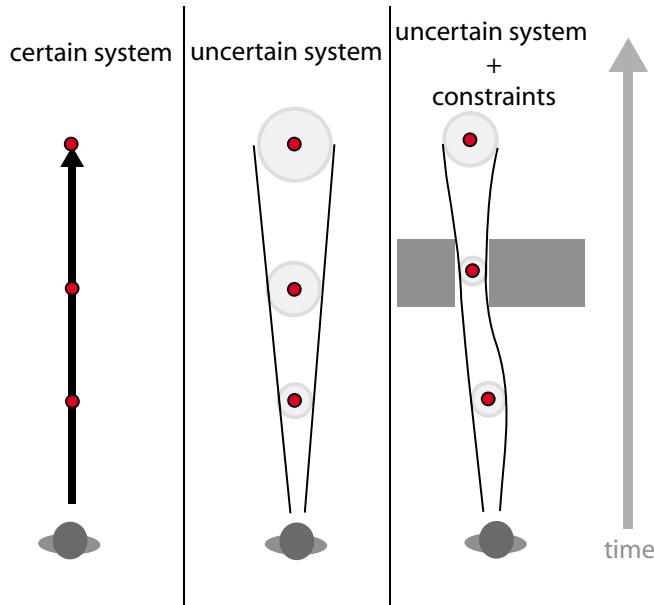


Fig. 4. Left: As we move forward in time (from bottom to top), if there was no uncertainty in our position or direction of travel, we would know exactly where we were going to be in a specified amount of time. Center: When we include uncertainty in the estimates of our position and direction we become less and less sure of our future position as we move forward in time. Right: If we include constraints, such as roads or gaps between buildings, as well as uncertainty, we can reduce the uncertainty in the constrained areas.

possible user paths into the future from some location along a given heading using Monte Carlo sampling, with uncertainty injected into both the location and heading estimates and constraints in the environment included we are presented with a distribution (represented by Monte Carlo sampled particles as in figure 2), which represent the most likely position of the user in a specified amount of time in a specific context as illustrated in figure 4.

With most currently available location-aware navigation software a single best estimate of position is made but this gives the user unreasonable confidence in the accuracy of their system and prevents the interactor from choosing an optimal strategy for dealing with the true state of the system. It has been shown that unrealistically precise feedback in the presence makes smooth, stable and accurate control difficult (Williamson *et al.* 2006) and (Strachan *et al.* 2007) and it is necessary to acknowledge that the kind of real world uncertainties we are exposed to when designing this kind of interaction place considerable limitations on the overall interaction (Strachan and Murray-Smith 2009).

A convenient side effect of this Monte Carlo propagation is that the future predicted positions can be used as a virtual probe for our virtual environment with the current future time horizon controlled with the pitch of the device and the direction taken from the compass heading as described above. By providing the user with the ability to move



Fig. 5. A Samsung Q1 UMPC with a Bluetooth connection to the WayStane inertial sensing device

this cloud anywhere in the virtual environment using this functionality and feeding back any interaction between the cloud and objects of interest using audio and haptic feedback, we enable a functional and embodied style of interaction.

3.2 Hardware

The current system runs on a Samsung Q1 Ultra Mobile PC with a Bluetooth connection to the WayStane (Murray-Smith *et al.* 2008) inertial sensing pack as shown in figure 5. This device contains the magnetometers, accelerometers and vibration devices that we require for this kind of interaction. The device also contains gyroscopes and capacitive sensing and is an adaptation of the SHAKE (Sensing Hardware Accessory for Kinaesthetic Expression) inertial sensing device.

4 Experiment

An experiment was conducted in order to quantify the users' ability to track a moving target, displaying varying behaviour, in a virtual environment using the described system with only audio and tactile (i.e. eyes-free) feedback.

Participants were first given an introduction to the system and a visual example before being allowed to practice using the functionality of the device. They were then blindfolded and placed in a specific spot within a large room which represented the center of a circle within the virtual environment, corresponding to a circle in the real world with an approximately 65m diameter. They did not move from this spot. They were then given the task of tracking a simulated human agent using the functionality of the inertial device. A screen shot is shown in Figure 7.

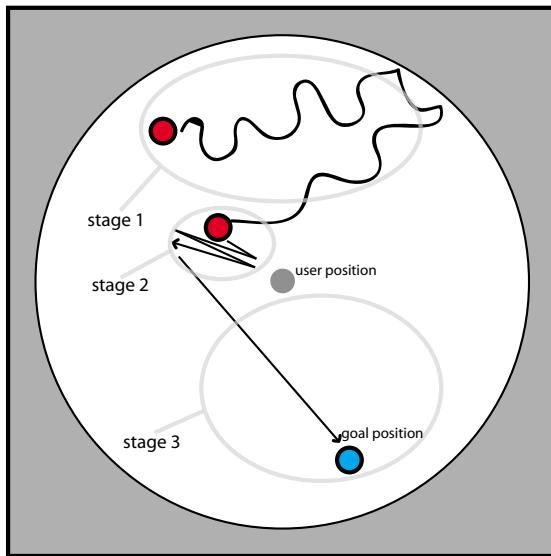


Fig. 6. The three stages of modelled human behaviour. Stage 1 shows the “random walk” phase where the agent randomly moves around the circular area until it is detected by the user. Stage 2 shows the “attention check” phase where the agent consecutively jumps from left to right in order to check if the user follows these jumps. Stage 3 shows the “goal-directed” phase where the agent moves straight towards a target point as long as the user is tracking the movement.

The modelled human agent displayed three kinds of behaviour, illustrated in Figure 6. Here we create a basic model of possible walking behaviour. In the first phase the agent is given random walk dynamics. From some initial position the agent moves by a distance drawn from a Gaussian distribution with a mean offset of 1.1m and a standard deviation of 0.5m. This distribution was observed from a GPS trace of one person walking at a normal pace outdoors in good GPS visibility. The agent was given a heading update drawn from a Gaussian distribution with mean 0 and standard deviation of 3.07°. This value was chosen to allow the agent the freedom to change direction without any implausible jumps in the heading which would lead to an erratic and unnatural walking pattern. The agent was also constrained to stay within the virtual circle.

The experiment participant was first helped to locate the agent, which is then tracked using the virtual probe for a short time until a switch to a second *attention check* mode is activated. This switch corresponded to the point where the user had sufficient contact (i.e. when the energy of impacted particles was above some threshold) with the agent for approximately 15s in total. The *attention check* part of the agent’s behaviour was designed to detect if the participant could explicitly pick up any unusual systematic hops in order to prove that the participant had really discovered the target. Such a hop might be typical of a gesture which a user might generate to indicate recognition of engagement, and which through imitation might propagate through a geosocial subculture. Immediately after the switch to this phase the agent consecutively jumps to the left, is detected by the participant, then to the right and is detected again for a total

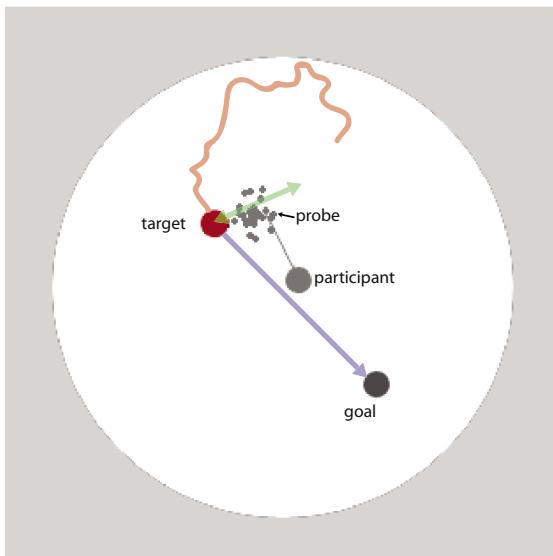


Fig. 7. Screen shot of user tracking an agent using their virtual probe. The red, green and blue superimposed lines indicate the random walk, attention detect and goal-directed stages respectively.

of 5 hops. Each hop corresponded to approximately 2.5m in the real world. This hop was designed to be large enough to be distinguishable from the movement in the first phase of agent behaviour. When the five hops were all successfully detected the agent switches to phase three, the goal-directed phase.

The *goal-directed* phase involves the movement directly to a particular place within the virtual environment. The agent is given the precise bearing of the target area and moves directly to that point only if the participant has sufficient contact with the agent. This is intended to test whether people could have their attention guided to a particular location. If the participant loses contact, the agent remains static until it is found once more. When the participant has tracked the agent to this target the experiment is complete. Participants who were performing particularly well after reaching the first goal were given up to two more goals to reach.

It is likely that if the experiment was conducted outdoors, it would lead to a more subjective analysis of user performance since we could not be aware of exact uncertainty present in the system. To enable a more objective comparison of user performances in a strictly controlled environment, we added uncertainty to the positions simulated, to create as realistic a situation as possible, testing whether this kind of interaction would be feasible in a realistic mobile spatial interaction system based on current commercially available technology. The uncertainty distribution of the participant's probe, as described in the uncertainty section was given a value in the heading estimate of 2.29° and a value of approximately 1.5m in the position estimate. Noise was also added to the participants position in order to simulate the effects of being in the real world. A value of approximately 2m was also added to the participants' virtual GPS position at

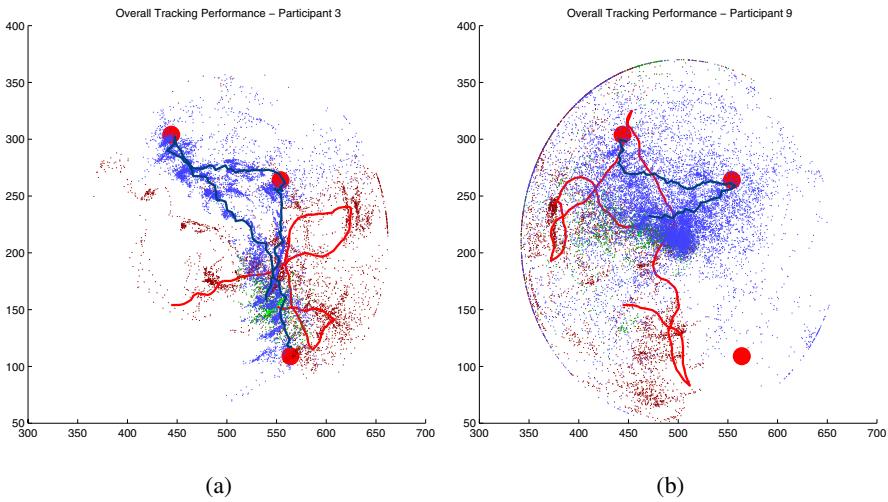


Fig. 8. (a) The full tracking run for participant 3 who showed a good overall performance. (b) The full tracking run for participant 9 who showed a bad overall performance. Red denotes the random walk phase. Green denotes the attention detection phase and blue denotes the goal-directed phase. The small dots indicate the area scanned by the user in each phase and the solid line indicates the position of the agent in each phase of the experiment. The large red circles indicate the locations of the goal areas.

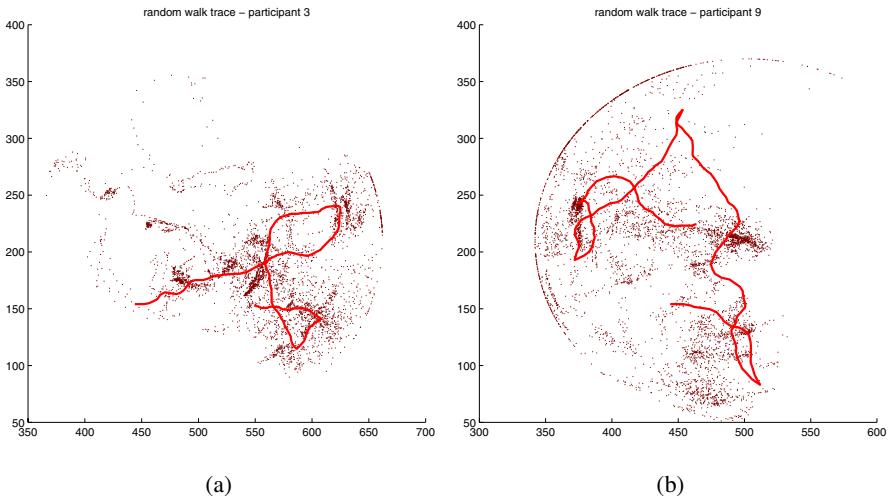


Fig. 9. Tracking performance for the random walk phase only, for a good (a) and bad (b) performance

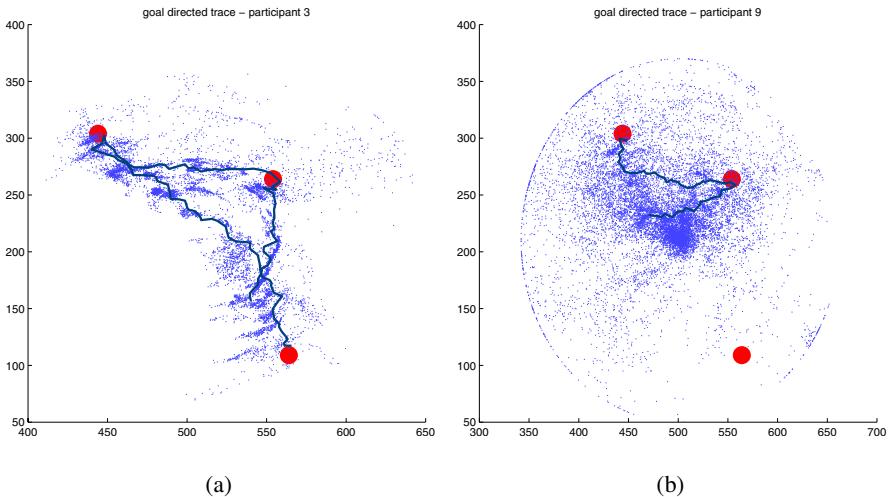


Fig. 10. Tracking performance for the goal directed phase only. One participant completes all 3 targets whereas the other fails to complete all 3.

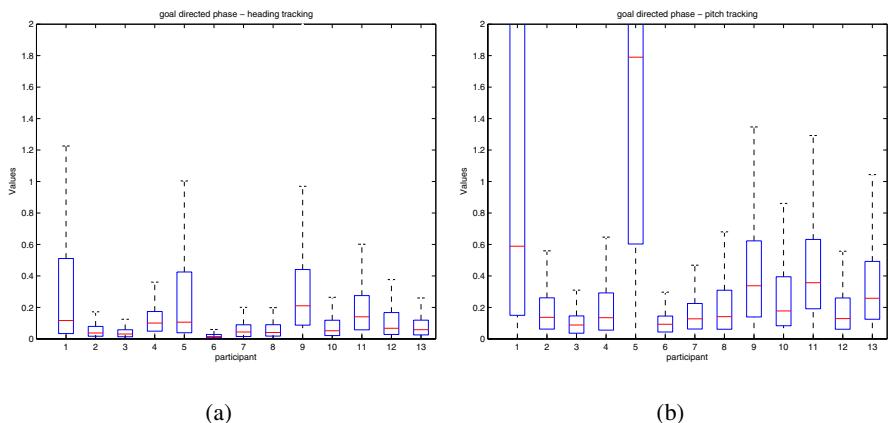


Fig. 11. (a) The mean squared error for all participants in the heading tracking task. (b) The mean squared error for all participants in the tilt tracking task.

1Hz, which produced a hopping effect on the participant's position. Although this is not strictly the effect we would observe from a real GPS position estimate, which is likely to shift much more gradually, and systematically over time, the method used was considered more consistent over all of the participants and was unlikely to lead to large periods of very difficult interaction for some users and not for others. 13 participants in total took part in the experiment ranging from 20-60 yrs of age with 10 males and 3 females.

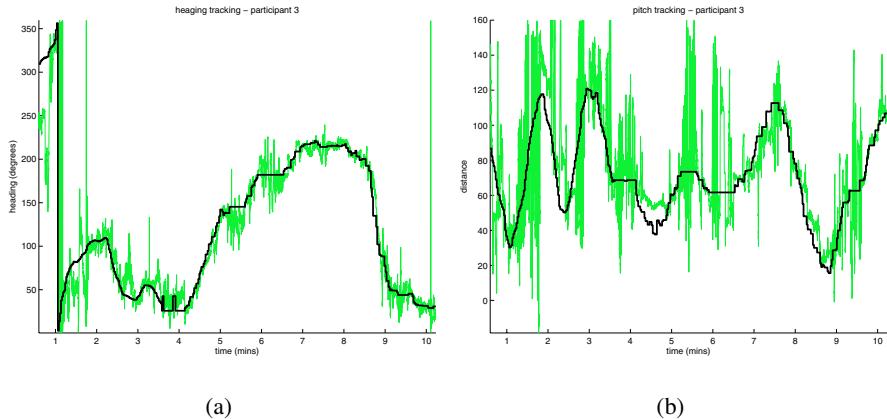


Fig. 12. (a) The participant tracks the heading of the agent very closely but struggles to track as effectively with the pitch (b). The green line represents the participants current heading and the black line represents the bearing of the agent.

4.1 Results

A wide range of performance was observed throughout the trial. 9 participants finished the whole task with all 3 goals, 4 achieved at least one target in the goal directed stage.

Figure 8 shows the difference observed over the whole run for one participant who performed very well and one who performed less well. It is clear to see that participant 3 stuck closely to the target path, shown in more detail in figures 9(a) and 10(a) whereas

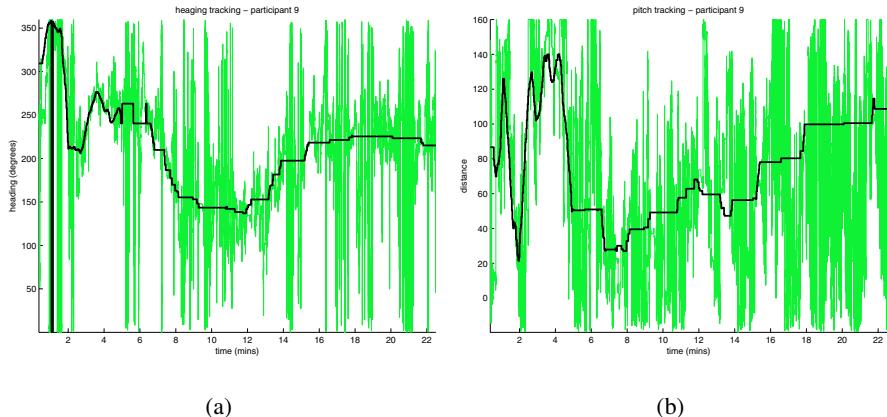


Fig. 13. (a) The participant performs badly in both the heading and pitch tracking tasks (b). The green line represents the participants current look-ahead distance (mapped to the device pitch) and the black line represents the bearing of the agent.

participant 9 performed a much longer search and covered a much larger area of the environment.

Figures 11(a) and 11(b) show the normalised mean squared error (i.e. the difference between the agent's position and the user's search position) in the tracking performance of each participant. It is clear that the participants who performed better overall, i.e. 2, 3, 6 and 7 have a much lower mean error meaning that they tracked the agent much more closely overall. The standard deviation, indicated by the size of the boxes in Figures 11(a) and 11(b) is also lower for these participants, again indicating that there was much less variation over the whole run. Participants with higher mean values and larger standard deviations are those who performed the least well. When examining the performance of participants for the tracking task in both the heading and pitch cases separately it is clear to see from figures 12 and 13 that users generally performed better when tracking the bearing of the agent than they did while tracking the distance of the agent. Figures 12(a) and 13(a) show the heading tracking performances for participants who performed well and poorly respectively and figures 12(b) and 13(b) show the distance tracking performances for the same participants, which shows a clear difference in performance. This backs up comments made by a number of the participants who claimed that they had more problem tracking the distance (back-forward movements) than the bearing of the agent and this is probably due to the fact that appropriate feedback was provided for the bearing tracking task (left-right audio pan) but not the distance tracking task.

5 Discussion and Conclusions

This paper introduced a new form of embodied geosocial interaction using current technology and demonstrates the feasibility of this new form of interaction via a user study that simulates both realistic environmental conditions and human behaviour. The system presented explicitly displays uncertainty in location, bearing and inferred context.

With the experiment it was found that users, with little experience or practice with this kind of system, were able to track a modeled human agent through three different types of behaviour. A number of issues were highlighted. By far the most common problem commented upon by the participants was the difficulty in tracking an agent which was moving towards or away from them. No participants had any significant problem with the heading tracking metaphor since this was a completely natural pointing technique with which we are all familiar. But since audio feedback was only provided for actual contact with the agent and slight movements to the left and right (represented by audio panning) there was great difficulty in the perception of the forwards and backwards movement of the agent. It was clear that different participants were perceiving the behaviour of the agent in different ways even though the agent was given identical behaviour for each run. Some users commented that sometimes the agent was moving around very fast and other times it was a normal speed when in reality the agent was simply further away when it was perceived as being slower and closer when it was perceived as being faster indicating that these participants had an issue with the the metaphor used for distance scanning. An obvious remedy to this problem would be the provision of explicit feedback for the towards and away motion, perhaps through another channel of communication. Another remedy could also be the provision of a more

natural metaphor than the pitch-distance mapping metaphor we used here. This issue does highlight though, the sensitivity that any results in location-aware interaction will have to very detailed aspects of interaction design, whether input or feedback.

This initial feasibility study has demonstrated that this new form of interaction is possible but there exists great potential for the further development of richer interaction design in this context. When designing novel forms of interaction in a virtual environment application it is possible to use the theory of social cognition to examine a user's interaction with the system and any emerging low level human interactions. For example, might we see any signs of social cognition in this kind of system? Social cognitive signals are almost subconscious for humans in the real world but how does this translate into the virtual world? The processes of joint attention and imitation, for example, are fundamental to the investigation of early human behavioral patterns. The ability to infer intentions from overt behaviour in geosocial systems will allow users the potential to evolve new styles of communication, and to use the systems in ways the designers might not have anticipated. This will range from straightforward cases such as strangers detecting a common focus of attention, which allows them to begin interaction with each other, to subtle ongoing turn-taking and imitation between friends who know each other well, and who, given their mutual context awareness, can communicate significant amounts of information about their mood and intentions with subtle movements of their mobile phone.

The experiment in this paper used a simulated agent and a human user, in order to improve repeatability and for more control of the activity levels. It is important to note that in this experiment the user was using their hand to track only the position of the agent. Hand movements are more expressive and more rapidly changing than location, and so the generalisation of this work to mutual interactions between two user's rapidly changing hand movements will require careful consideration. This first step should act as the starting point for further work on joint attention in geosocial interaction. The next challenge is to expand this work to include multi-user interaction to validate the results of this paper with two human users, and to observe the detailed interactions that evolve as people engage and disengage from remote contact with each other.

Acknowledgements. Both authors had positions with the Hamilton Institute, NUI Maynooth when this research was carried out. We are grateful for support from: SFI grant 00/RFP06/CMS052, EPSRC project EP/E042740/1, *Multimodal, Negotiated Interaction in Mobile Scenarios*, IST Programme of the European Commission, under PASCAL Network of Excellence, IST 2002-506778, and the EC-funded *OpenInterface* Project. Nokia provided a donation of funds and equipment.

References

- Benford, S., Greenhalgh, C., Rodden, T., Pycock, J.: Collaborative virtual environments. *Commun. ACM* 44(7), 79–85 (2001)
- Clark, H.: *Using Language*. Cambridge University Press, Cambridge (1996)
- Crossan, A., Brewster, S.: Two-handed navigation in a haptic virtual environment. In: CHI 2006: CHI 2006 extended abstracts on Human factors in computing systems, pp. 676–681. ACM, New York (2006)

- Espinoza, F., Persson, P., Sandin, A., Nyström, H., Cacciatore, E., Bylund, M.: Geonotes: Social and navigational aspects of location-based information systems. In: Abowd, G.D., Brumitt, B., Shafer, S. (eds.) *UbiComp 2001*. LNCS, vol. 2201, pp. 2–17. Springer, Heidelberg (2001)
- Frith, C.D., Frith, U.: How we predict what other people are going to do. *Brain Research, Multiple Perspectives on the Psychological and Neural Bases of Understanding Other People's Behavior* 1079, 36–46 (2006)
- Fröhlich, P., Baillie, L., Simon, R.: Realizing the vision of mobile spatial interaction. *Interactions* 15(1), 15–18 (2008)
- Greenhalgh, C., Benford, S.: Massive: a distributed virtual reality system incorporating spatial trading. In: *Proceedings of the 15th International Conference on Distributed Computing Systems*, pp. 27–34 (1995)
- Jones, M., Jones, S., Bradley, G., Warren, N., Bainbridge, D., Holmes, G.: Ontrack: Dynamically adapting music playback to support navigation. *Personal and Ubiquitous Computing* 12, 513–525 (2008)
- Magnusson, C., Rasmus-Grohn, K.: A virtual traffic environment for people with visual impairment. *Visual Impairment Research* 7(1), 1–12 (2005)
- Murray-Smith, R., Williamson, J., Quaade, T., Hughes, S.: Stane: synthesized surfaces for tactile input. In: *CHI 2008: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pp. 1299–1302. ACM, New York (2008)
- Robinson, S., Eslambolchilar, P., Jones, M.: Point-to-GeoBlog: gestures and sensors to support user generated content creation. In: *Proceedings of the 10th international Conference on Human Computer interaction with Mobile Devices and Services, MobileHCI 2008*, Amsterdam, The Netherlands, pp. 197–206. ACM, New York (2008)
- Strachan, S., Murray-Smith, R.: Geopoke: Rotational mechanical systems metaphor for embodied geosocial interaction. In: *NordiCHI 2008: Proceedings of the fifth Nordic conference on Human-computer interaction*, pp. 543–546. ACM, New York (2008)
- Strachan, S., Murray-Smith, R.: Bearing-based selection in mobile spatial interaction. *Personal and Ubiquitous Computing* 13(4) (2009)
- Strachan, S., Williamson, J., Murray-Smith, R.: Show me the way to monte carlo: density-based trajectory navigation. In: *CHI 2007: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 1245–1248. ACM, New York (2007)
- Williamson, J., Strachan, S., Murray-Smith, R.: It's a long way to Monte Carlo: probabilistic display in GPS navigation. In: *MobileHCI 2006: Proceedings of the 8th conference on Human-computer interaction with mobile devices and services*, pp. 89–96. ACM Press, New York (2006)

Directional Beacons: A Robust WiFi Positioning Method Using Angle-of-Emission Information

Kensaku Kawauchi¹, Takashi Miyaki², and Jun Rekimoto²

¹ Graduate School of Interdisciplinary in Information Studies,

² Interfaculty Initiative in Information Studies,

University of Tokyo 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan

Abstract. WiFi-based positioning has been widely used because it does not require any additional sensors for existing WiFi mobile devices. However, positioning accuracy based on radio signal strength is often influenced by noises, reflections, and obstacles. The Time-of-Arrival (TOA) or Angle-of-Arrival (AOA) methods may be used, but both require additional sensing mechanisms and cannot be applied to existing WiFi mobile devices. In this paper, we propose a new WiFi-based positioning method called directional beaconing. This method uses the Angle-of-Emission (AOE) method instead of the AOA. Using this method, access points (APs) emit beacon signals through rotating directional antennas with angle information encoded in beacons. WiFi devices estimate the direction and distance to the AP by receiving and decoding these beacons. This method integrates the advantages of the AOA and signal strength methods without requiring additional sensors. We constructed revolving directional APs and verified positioning accuracy is improved using the proposed method.

Keywords: WiFi, Location Estimation, Directional Beacon, Directional Antenna, Angle of Arrival, Angular Information.

1 Introduction

Recently, WiFi-enabled mobile devices (e.g., mobile phones, music players, or digital cameras) have become very popular, and WiFi access points (APs) abound in most urban areas. Using this infrastructure, various kinds of WiFi-based location estimation techniques have been proposed [1,2,3]. Location estimation is a fundamental technique for many ubiquitous computing services and its accuracy is crucial to the quality of those services. Therefore, the development of accurate WiFi-based localization techniques is necessary.

There are three well-known methods for location estimation using wireless signals: Received Signal Strength Indicator (RSSI), Time-of-Arrival (TOA), and Angle-of-Arrival (AOA) methods.

The RSSI method estimates the distance between a mobile device and APs using the signal strengths of the beacon signals. By combining distance information to multiple APs and the APs absolute location information, the position

of the mobile device can be calculated [1,2,3]. For this purpose, mobile devices only measure the signal strengths of beacon packets, so this method does not require any special hardware/sensing device for mobile devices. Thus, many existing WiFi mobile devices can use this method without any modifications to the hardware. This is one of the advantages of this method. In addition, for receiving beacon packets, mobile devices do not need to establish a network connection. Thus, the system can cover a wide area, even in outdoor or indoor environments. On the other hand, estimated distance is not always reliable because of unstable signal strengths, which can be affected by fading, attenuation, obstacles, or reflections from environments such as buildings or trees. Thus, the overall accuracy of location estimation tends to be low.

The TOA method employs the arrival time of signals from a mobile device to estimate distance. To elaborate, this technique is based on the comparison of arrival times between more than three APs, and the mobile node is called the Time Difference of Arrival (TDOA) [4,5]. By measuring the arrival times, this method succeeds in being more precise than the RSSI method. This is because the noise caused by multipath reflections can be separable only by observing the beacon packet that has the shortest arrival time. However, the TDOA system needs to have a highly time-synchronized set of APs in the environment, so it is not feasible for covering wider areas such as outdoor urban city environments.

The AOA method uses angular information to estimate the location of mobile devices. A mobile device using this method has a sensor that measures the arrival orientation of radio signals (i.e., angles of arrival). Using this AOA information, it is possible to estimate location by comparing two arrival orientations from two different APs. AOA information is also more robust than radio signal strength information, but this design requires a special signal orientation sensor such as a directional antenna. Thus, this method cannot easily be applied to existing mobile WiFi devices.

In this paper, we propose a new WiFi position estimation method based on Angle-of-Emission (AOE) information. This method also uses signal orientation information, but does not require any modification to mobile devices. Instead, it uses special APs with rotating directional antennas to transmit beacon signal packets along a designated direction.

This approach has the potential to cover wide areas with a high degree of accuracy. In conventional AOA-based WiFi localization systems, equipment capable of measuring the arrival angle information of beacon packets is necessary. However, it is difficult to add such hardware to every mobile device. Our proposed AOE method only requires a transmitting mechanism on the AP side that emits beacons with directional information. Ordinary mobile devices simply have to decode directional information from the beacons to determine the location relative to the AP. Decoding directional information is possible by simply installing software on mobile devices. Thus, the entire localization system can be created in a cost-effective way. In addition, this method can also be combined with the RSSI method in order to increase accuracy.

To evaluate the feasibility and the estimation accuracy of our proposed method, we constructed a customized AP with a rotating directional antenna for emitting beacons with directional information. We call this method directional beaconing.

2 Directional Beaconing

Our proposed method is based on AOE information. This method is designed to more precisely and cost-effectively cover wider areas than other WiFi localization methods.

The directional beaconing method uses a directional antenna to control the orientation of emitted beacon packets (shown in Fig. 1). This controlled orientation information is also embedded in the beacon packets, so mobile devices need only decode a received packet, without measuring signals angles of arrival. Using the rotating directional antenna, this directional beaconing method enables orientation estimation without the need for special antenna transition or time synchronization of multiple APs. It is enough for mobile devices to decode the received beacon packets in the same way as in the RSSI method.

Additionally, there are other kinds of antennas that are capable of emitting controlled directional wave signals. An array antenna [8] is an example of such an antenna. It can control directional wave signals using a fringe pattern. The array antenna can change the orientation of the directional wave without the need for mechanical equipment. Combinations of existing technologies make it possible to construct a rotating mechanism without an array antenna.

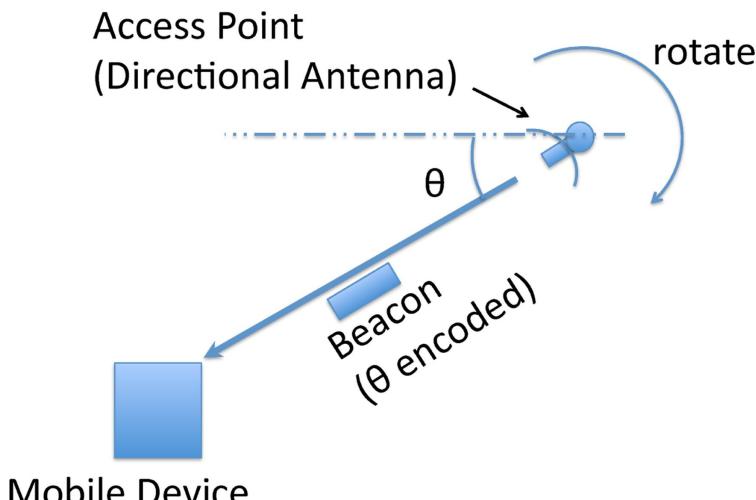


Fig. 1. The concept of directional beaconing: beacons emit directional information from APs through rotating directional antennas. Client location is calculated by decoding the received information.

An AP for directional beaconing is not supposed to access the internet in the same manner as normal APs. However, WiFi mobile devices can receive beacon packets emitted from APs for directional beaconing because our proposed method does not edit the basic frame of a beacon packet. Therefore, it is possible to simultaneously use normal APs and special APs for directional beaconing. In addition, WiFi mobile devices that are capable of extracting AOE information from received packets use both the RSSI method and the directional beaconing method.

3 Comparison with the Conventional AOA Method

Although our approach is also based on angular information, our method has several advantages over the conventional AOA method. There are two ways to estimate location using arrival angles: AP measurement [7] and mobile device measurement of angles [6] (shown in Fig. 2). In the case of a mobile devices measurement of arrival angles, the device needs a special antenna that is capable of detecting arrival angles. However, it is difficult to add such hardware to all mobile devices. Furthermore, the use of mobile devices for measuring arrival angles is affected by the attitude and direction of the mobile device. In order to precisely estimate arrival angles from an AP on a mobile device, an estimating system that is not affected by attitude or direction, such as a tilt compensated compass, is necessary. Therefore, estimating arrival angles with mobile devices tends to be a complex task. In contrast, the measurement of arrival angles by an AP requires a special antenna that is capable of estimating the directions of received wave signals. This requirement also prevents the system from covering a wide area because of its installation costs. In addition, communication between the AP and a mobile device is necessary, but a network connection that can transmit this information is not always available. Therefore, the measurement of arrival angles by an AP is also a complicated task in most environments. On the other hand, our proposed directional beaconing method, which is based on AOE information, is capable of estimating locations using most standard WiFi devices. This is because AOE information is simply emitted from a rotating directional antenna. In order to receive location information, the clients mobile device must simply calculate embedded information in received beacons. Therefore, our method can be applied to existing mobile devices without the need for hardware modification.

Although a rotating directional antenna must be installed in APs, the number of APs necessary for this method is lesser than that for conventional methods. Because APs with directional antennas can transmit beacons 150 m farther than those with non-directional antennas, our method can cover a wider area with a single AP. Furthermore, APs with directional antennas do not require network access because the calculation of location estimation is performed by the mobile devices. A simple system configuration is the only requirement for transmitting beacons with angular information. Moreover, our proposed method can be combined with the conventional RSSI method in order to enhance the accuracy of location estimation.

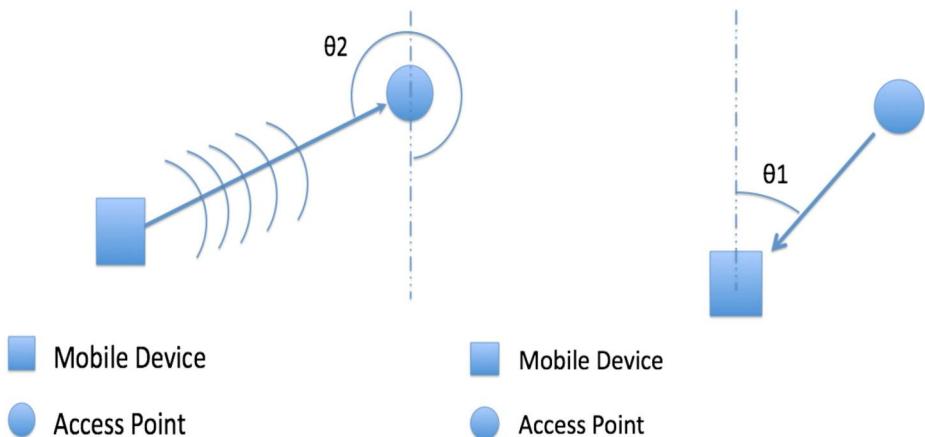


Fig. 2. Comparison with the AOA method: AP measurement of arrival angles [7] (left) and mobile device measurement of arrival angles [6] (right)

4 System Configuration

4.1 Directional Antenna

In order to demonstrate the feasibility of localization using directional beaconing, we constructed a special AP system with a rotating directional antenna. The AP employs a parabolic grid antenna from HyperLink Technologies as a directional antenna [10]. A directional antenna with a narrower beam width angle is preferred because the signal may be distinguished from different directions. Thus, this antenna is mounted in a horizontal position because the beam width is narrower in this position than in the vertical one. Fig. 3 shows RF antenna patterns in vertical and horizontal planes derived from a specification sheet.

4.2 The Directional Beaconsing AP

We constructed a special AP called the directional beaconing access point (DBAP), which is shown in Fig. 4.

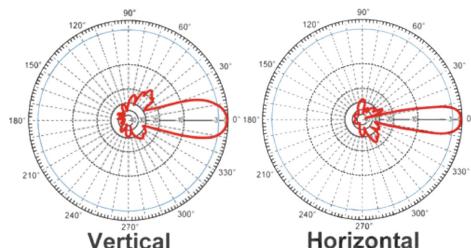


Fig. 3. Hyperlink Technologies HG2415 directional antenna gain patterns

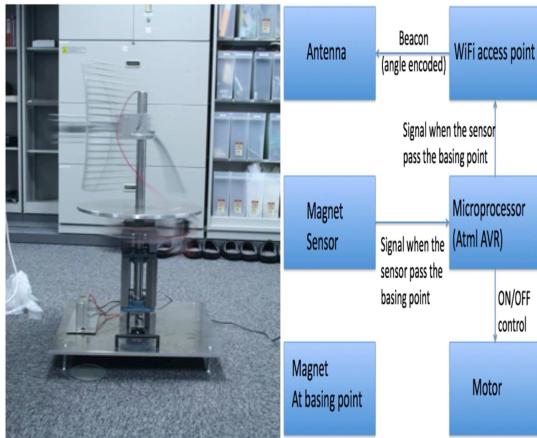


Fig. 4. The DBAPs constructed rotating directional antenna (left) and a block diagram of the system (right)

Our proposed method requires an AP that emits beacon signals with AOE information. In our DBAP, AOE information is embedded in the beacon signals. We connected this directional antenna to the FON WiFi router [9]. FON is based on Linux and its control software is based on open source WiFi AP software. Furthermore, it is possible to modify its beacon emitting algorithm. We then modified the router so that the AOE information is encoded in beacon packet was emitted from the AP.

A system equipped with the FON router determines AOE information with received signals from a microcontroller on the DBAP. The system computes one round time of DBAP with received signals. A formula for determining the AOE information is as follows:

$$\text{AOE} = \frac{(\text{current time} - \text{recent passing time})}{\text{one round time}} \cdot 360 \quad (1)$$

The microcontroller installed on the AP system controls the motors rotation. This microcontroller can also transmit a signal to the WiFi AP when a magnet sensor passes through a base point on the turntable. The base point is located at an angle of 0 degrees. Using this passing time and the time of entire rotation, the system on the WiFi AP determines current angle values of the antenna.

In the current implementation, AOE information is embedded in the Service Set Identifier (SSID) of the beacon packet. Clients can estimate their locations by extracting the AOE information from the SSID.

In our experiment, the DBAP made a complete revolution every 10 s and the FON Wi AP emitted beacon signals every 50 ms. However, after optimizing the settings, the FON Wi AP is capable of emitting beacon signals every 25 ms. The revolution speed is determined by the emitted degree of interval and the number

of beacons with the same AOE information. If a DBAP emits AOE information at intervals of 5 degrees using the optimized setting, clients will need about 1 s to receive all the AOE information five times.

5 Experiment

5.1 Experiment 1

Relations between the signal strength and the relative location from a directional antenna were examined. This experiment requires a space in which there are no obstacles to affect the signals. In order to capture signals from the directional antenna, the experiment was performed on a baseball field (110 by 120 m).

There were no obstacles to affect the received signal strength in this experiment. Thus, the line of sight (LOS) condition is supposed. Beacon signals from the antenna were captured on a laptop while walking around the observation area. Fig. 5 shows the signal strength map of the received beacons (1141 captured signals) from the directional antenna. In Fig. 5, directional characteristics are confirmed by considering the signal strength.

Signals along the direction of radiation were extracted from the captured samples. Fig. 6 shows that the signals were extracted from line (a) of Fig. 5. Sample signals near the antenna are strong, while points far from the antenna exhibit low signal strengths. Therefore, it can be asserted that signals from a

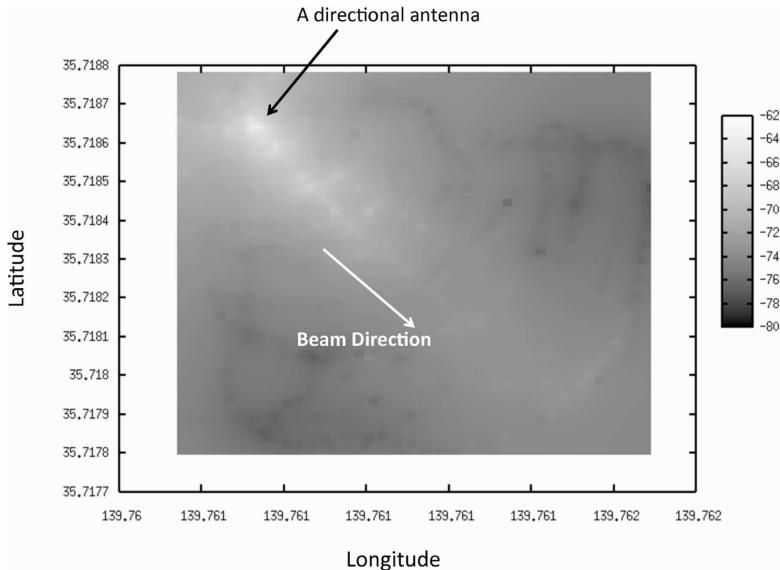


Fig. 5. Signal strength map of the DBAP in an ideal LOS environment without rection (the data was captured at a baseball field in order to minimize obstacles) [10]

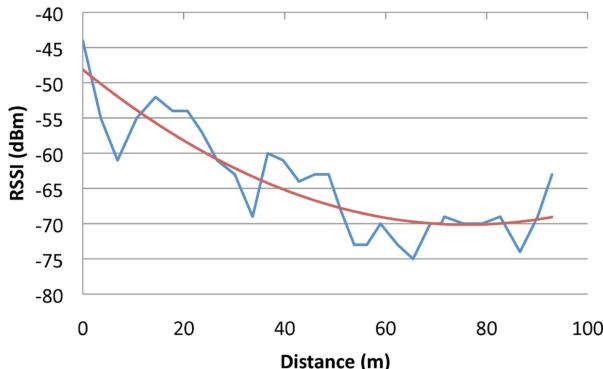


Fig. 6. Relationship between the distance from the AP and the signal strength surveyed in the LOS environment (Fig. 5)

directional antenna are weakened by distance. However, the signal strength at the back yard of the observation area was increased because of reflection from a fence. This experiment showed the performance of a directional antenna in LOS conditions.

5.2 Experiment 2

We verified whether or not WiFi mobile devices are capable of receiving peak signal strength from the DBAP. The DBAP was installed at point A in Fig. 7. An observation point was located at a 45-degree angle to the base point of the DBAP. The distance from the observation point to the DBAP was 100 m. WiFi mobile devices located at the observation point collected beacon signals from the DBAP. Then, the average signal strengths were calculated for each sample. Although there were at least five packets on each degree, the number of received packets was inconsistent.

Therefore, the average signal strengths were calculated using the first five packets received from the DBAP. Fig. 8 shows the results of this calculation.

The mobile devices collected AOE information at 5-degree intervals. It was determined that directional beaconing could successfully estimate the observation point when the highest signal strength was 45 degrees. The positioning accuracy of our proposed method is not influenced by noise, reflections, or obstacles. Therefore, the directional information derived from the directional beaconing method could be effectively used for WiFi-based positioning.

Next, we evaluated the differences between the actual angle and the AOE information from the DBAP in order to estimate the accuracy of the directional beaconing method. The observation area was on the broken line in Fig. 7. DBAPs were installed on point A and point B in Fig. 7.

Beacon signals from the DBAPs were captured at 25 grid points by a WiFi device. At each point, the WiFi client collected signals and decoded AOE information in the same manner as in the previous verification. An estimated angle

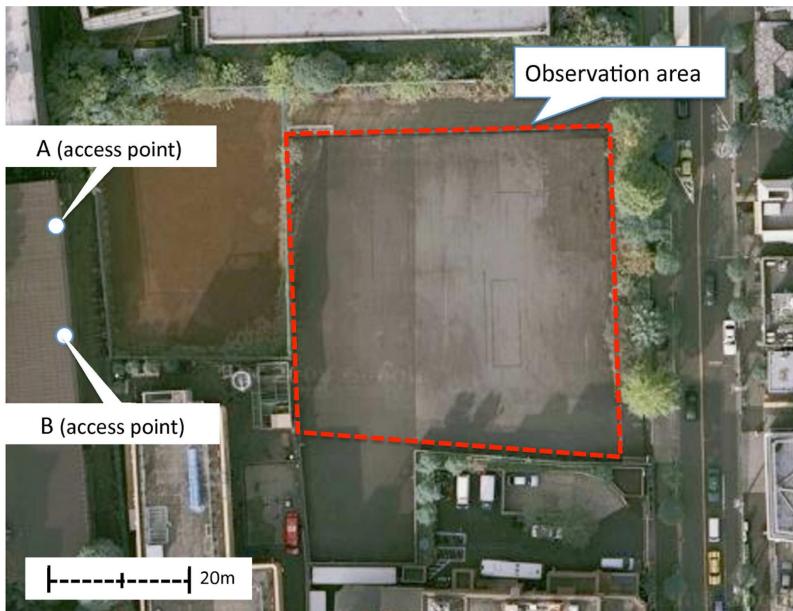


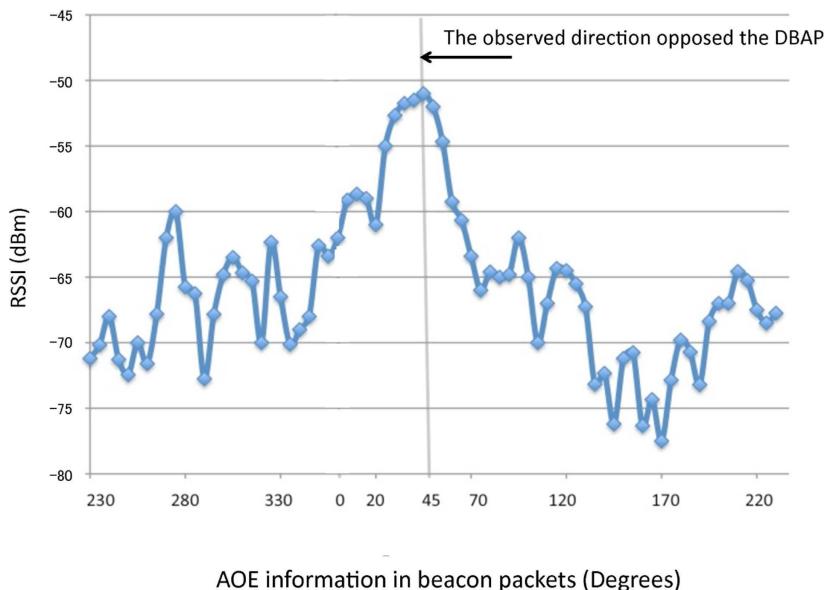
Fig. 7. The second experiments observation area map

on each point was determined by AOE information of the highest signal strength from received signals on each point of the correspondent. Fig. 9 shows the differences between the actual angle and the result from decoded AOE information. As a result, 40 out of 50 points (80% of the samples) are covered within a margin of error of ± 10 degrees.

On the other hand, the signal strength map (Fig. 10) was constructed to verify whether RSSI can be incorporated in the directional beaconing method. If RSSI could be incorporated in the directional beaconing method, WiFi devices could estimate a location using only one DBAP. Because the RSSI method provides standards of judgment regarding the location of a WiFi device, the estimation area of the device may be further reduced.

However, distance is not related to signal strength in Fig. 10. Moreover, in order to survey the observation point, the distribution of the signal strength on the observation point (Fig. 11) was determined and compared with the ideal LOS environment (shown in Fig. 6). As shown in this graph, in the area that is closer to the AP ($40 \sim 60$ m in (A) line), the captured signal strength values are similar to those in the LOS environment. However, in the area that is $60 \sim 80$ m from the AP, the captured signal strength tends to be higher than in the LOS environment.

From Fig. 12, it can be seen that the signal strength from the non-directional antenna was not inversely proportional to the interval. In brief, a distance estimation result using the RSSI method tends to be lower because of reflections from buildings.



AOE information in beacon packets (Degrees)

Fig. 8. Characteristics of signals from the DBAP: AOE information and signal strength when the WiFi device received beacon packets from the DBAP are shown. The AOE information of the highest RSSI value is equal to the observed direction opposed the DBAP.

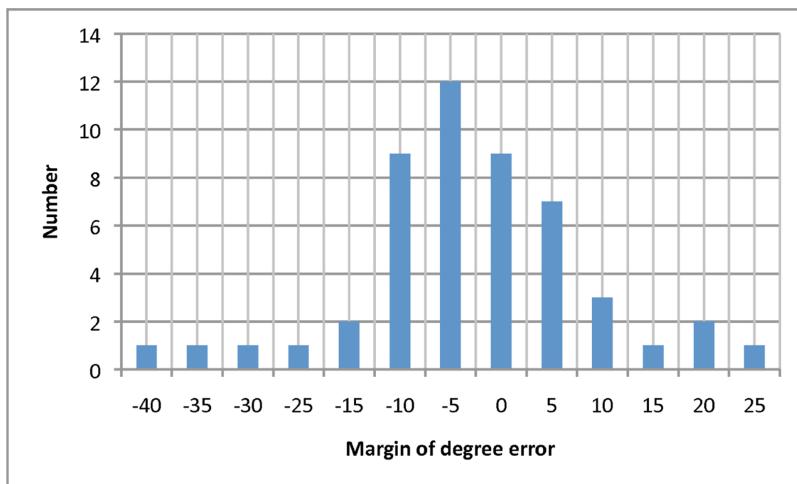


Fig. 9. Distribution of angular error of captured directional beacons

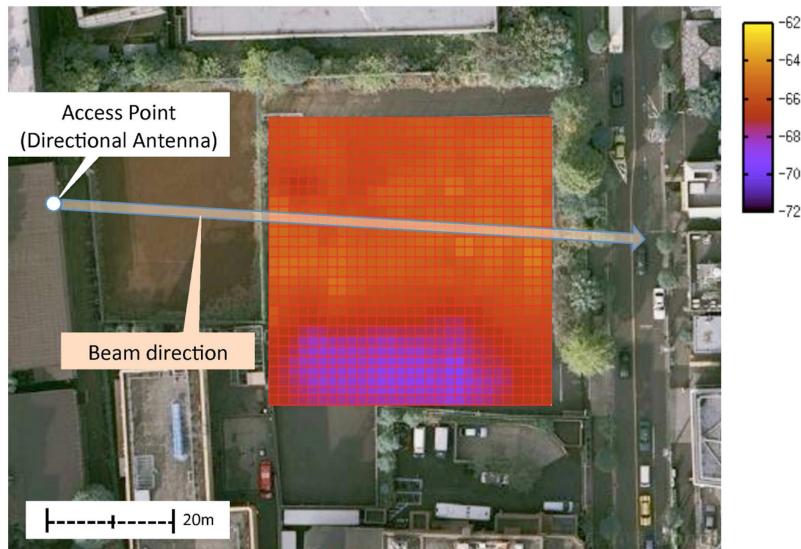


Fig. 10. Signal strength map of the directional antenna in Experiment 2

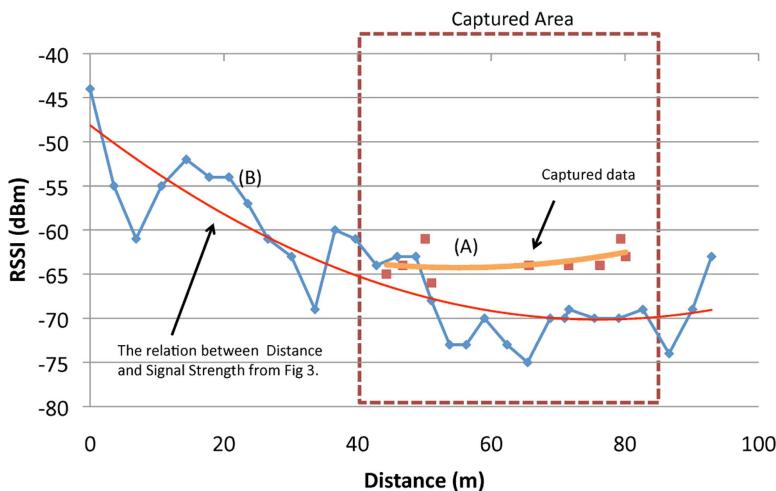


Fig. 11. Correlational differences between the observation area and an ideal LOS environment: the (B) line was captured in the LOS environment. The orange line represents the relationship between distance and signal strength in the observation area.

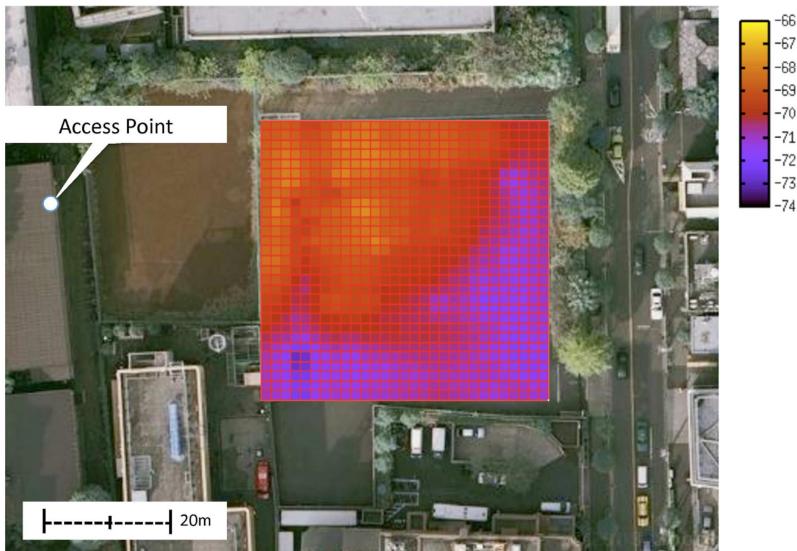


Fig. 12. Signal strength map of the non-directional antenna in Experiment 2

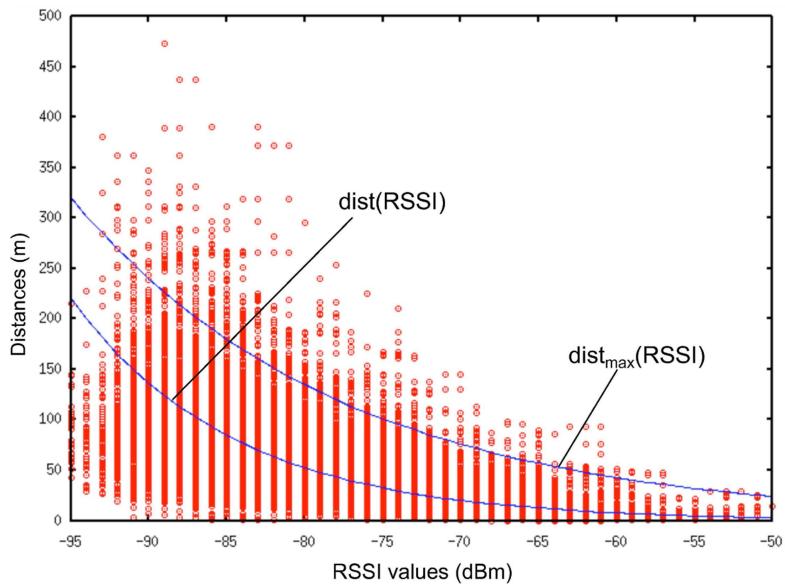


Fig. 13. PlaceEngine: the relationship between signal strength and distance

5.3 Experiment 3

In an ideal LOS environment, a mobile device can estimate distance using the correlation between RSSI and distance, as shown in Fig. 6. However, in urban

areas, where there is considerable reflection from obstacles (such as the observation area in Experiment 2), it has been found that estimating the precise distance between APs and a device is a difficult task (Fig. 11). Therefore, in order to combine a distance estimation technique used in the RSSI method with the techniques used in the directional beaconing method, a correlation model between the signal strength and the distance is required for a directional antenna used in urban areas.

In the PlaceEngine [3] project, an RSSI-based city-wide WiFi positioning method previously proposed by authors group, we had already captured signal strength values from dense APs in urban areas. The captured area included Non-Line-of-Sight (NLOS) environments with reflections from obstacles.

Fig. 13 shows the results of observing 800 WiFi APs and capturing 15,000 observations. From this data, a statistical reference model was developed for urban environments. This model is actually used in order to estimate distance in the PlaceEngine project.

Therefore, if a similar statistical model of the correlations between received signals and distance using a directional antenna is constructed, it would be possible to simultaneously combine distance estimations from the RSSI method and orientation estimations from the directional beaconing method.

6 Discussion

6.1 Location Estimation

From the experiments in the previous section, it can be seen that the directional beaconing method can be applied in real-world environments. In the directional beaconing method, time-varying beacon signals with embedded directional information enable mobile devices to decode orientation information. Because of its capabilities, this method enables precise location estimations using fewer APs than other methods. For the evaluation of the methods characteristics, a comparison between directional beaconing and the RSSI method (the most popular localization technique for urban areas) was conducted. We considered the important of the number of APs with regard to the accuracy of each method in ideal environments.

Fig. 14 shows an estimation candidate area for both the directional beaconing and RSSI methods in ideal LOS environments. Fig. 14(A) shows location estimation using the RSSI method and a single non-directional antenna. The area in which WiFi devices can receive wave signals from an AP is circled. The mobile device did not have the means to detect orientation to the AP, thus the estimation candidate is designated by the green-colored area in the RSSI method.

Directional beaconing using single AP estimation is shown in Fig. 14(B). The directional beaconing method requires a DBAP for emitting beacon signals containing AOE information. From the results of Experiment 2, it can be seen that the accuracy of the directional beaconing method using one DBAP showed a margin of error of ± 10 degrees in most cases (80% of all observations). Because

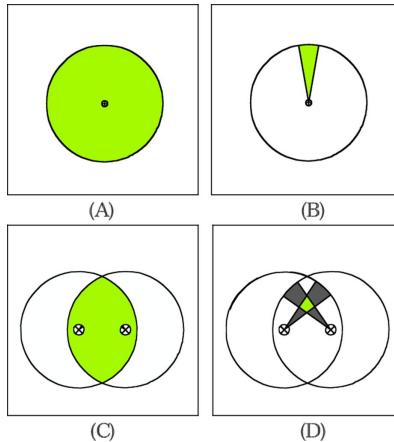


Fig. 14. Estimating accuracy with each method: (A) the RSSI method using one normal AP; (B) the directional beaconing method using one DBAP; (C) the RSSI method using two normal APs; (D) the directional beaconing method using two DBAPs

WiFi devices are capable of estimating the direction of a DBAP, the estimation area of the WiFi device is reduced to 5% of the beacons area of reach when compared with (A).

In addition, the RSSI and directional beaconing methods using two APs are shown in Figs. 14(C) and (D), respectively. In the case of the RSSI method (C), the possible estimation area is the area where the two circles overlap. However, using AOE information (D), the estimation area is the overlapped area of two beam width, which is even smaller than (B) or (C). Moreover, using both AOE information and signal strength information, as shown in Figs. 14(B) and (D), there is the possibility of reducing the estimation candidate area.

It can then be seen that our proposed directional beaconing method can be applied to accurately estimate a location. In addition, fewer APs are required. In order to construct an estimation system using the directional beaconing method, only a small number of AP-side transitions are required, making this approach cost-effective and suitable for various urban environments.

6.2 Accuracy of Estimation

Next, we discuss the accuracy of the directional beaconing method using the result of Experiment 2 in this paper. In Experiment 2, beacon signals from multiple DBAPs (placed at points A and B in Fig. 7) are captured at the same time. The measurement of the estimation accuracy used two results of extracted and analyzed AOE information within the beacon signals from each DBAP. An estimation of location using the directional beaconing method is determined by the estimation angles point of intersection. Fig. 15 shows the estimation area image. Line x and line y (Fig. 15) represent the estimated angles of each DBAP.

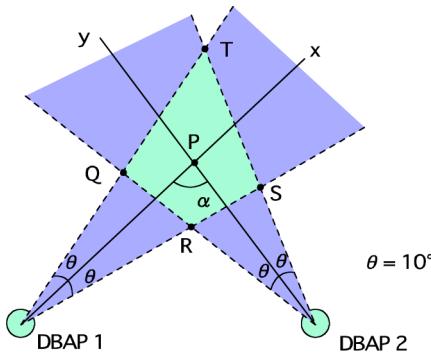


Fig. 15. Estimation area image: In this figure, the estimation area is surrounded by point Q, R, S, and T

The point of intersection (point P) is the ideal estimation point. It is clear from the results of Experiment 2 that estimation accuracy of the directional beaconing method shows a margin of error of ± 10 degrees. Therefore, the directional beaconing method estimates that a client is in an area surrounded by points Q, R, S, and T.

The estimation point was compared with the actual point. Observation points that are less than 60 m from points A and B (Fig. 7) have a low margin of error with regard to distance. In the best case scenario, the margins between an estimated point and an actual point are 7.5 m. Conversely, at an observation point farther than 60 m from points A and B (Fig. 7), margins of error with respect to distance are high.

If the angle α decreases (Fig. 15), the distance between point R and point T increases. The position estimation results are very sensitive to errors relating to estimation angles. As a result, estimation accuracy tends to be lower in detached areas. However, the extended area is restricted by the RSSI and directional beaconing methods, as described in a previous subsection.

6.3 Configuration

In this paper, AOE information is embedded in the SSID of the beacon frame so that devices that cannot analyze received beacon packets can estimate location. Changing the SSID means that the WiFi devices cannot connect to the AP. However, it is possible to embed AOE information in another field in an IEEE802.11 standard beacon frame. If the embedded field is changed, clients will be able to access the Internet because the SSID remains unchanged. There is a trade-off between the revolution speed of the DBAP and the quality of network bandwidth. Further study is needed in order to determine the ideal revolution speed.

Using a directional antenna and a turntable, a DBAP can be easily constructed. The structure is large because the antenna and the turntable are separate parts. In addition, to maintain mechanical systems like a turntable on

DBAP is needed. A DBAP requires that the antenna be directional and that the direction of this antenna be changeable. Therefore, the directional beaconing method uses an array antenna for the DBAP because it satisfies these requirements. It is possible to use other antennas and systems for DBAPs, provided they satisfy the above requirements.

7 Conclusion and Future Research

In this paper, we proposed a new WiFi-based positioning method called directional beaconing. This method uses access points (APs) with rotating directional antennas and combines the benefits of the radio-signal strength method and the Angle-of-Arrival (AOA) method. Instead of measuring received radio signal orientations, APs with rotating directional antennas are used to emit beacons containing orientation information, termed as Angle-of-Emission (AOE) information. Since the orientation information is embedded in the beacon packets, this method can be used to determine the positions of normal, unmodified WiFi mobile devices. Our pilot evaluation shows that this method correctly estimates the position of a device, even in an environment with obstacles or reflections.

The essential idea of directional beaconing is to embed beacon emission conditions (i.e., orientation) in a beacon packet so that a mobile device that receives this beacon can estimate orientation. We can also generalize this idea in order to embed other dynamically changing conditions in beacon packets. We call this generalized idea dynamic beaconing. For example, suppose that the emission power of beacons changes dynamically. This might be beneficial when an AP is used for indoor positioning with many radio reflections. A beacon with a very low emission power will only be received by nearby mobile devices, which eliminates the problem of radio reflection.

During the experiments, we also noted that the received signal strength consistently changed according to the orientation of the AP. If the AOE orientation and the signal strength correlation is consistent, we can use this information as a kind of WiFi fingerprint, even if the signal strength does not conform to a radio-signal decay model (because of reflections or obstacles). In contrast to existing WiFi signal strength-based methods, this approach may support positioning without greatly increasing the number of APs because one directional AP can emit beacons in different directions and with varying signal strengths. We are especially interested in applying this idea to indoor positioning.

References

1. Bahl, P., Padmanabhan, V.N.: RADAR: An In-building RF-based User Location and Tracking System. In: INFOCOM, vol. 2, pp. 775–784 (2000)
2. LaMarca, A., Chawathe, Y., Consolvo, S., Hightower, J., Smith, I., Scott, J., Sohn, T., Howard, J., Hughes, J., Potter, F., Tabert, J., Powledge, P., Borriello, G., Schilit, B.: Place Lab: Device Positioning Using Radio Beacons in the Wild. In: Gellersen, H.-W., Want, R., Schmidt, A. (eds.) PERVASIVE 2005. LNCS, vol. 3468, pp. 116–133. Springer, Heidelberg (2005)

3. Rekimoto, J., Miyaki, T., Ishizawa, T.: LifeTag: WiFi-based Continuous Location Logging for Life Pattern Analysis. In: Hightower, J., Schiele, B., Strang, T. (eds.) LoCA 2007. LNCS, vol. 4718, pp. 35–49. Springer, Heidelberg (2007)
4. Golden, S.A., Bateman, S.S.: Sensor Measurements for Wi-Fi Location with Emphasis on Time-of-Arrival Ranging. IEEE Transactions on Mobile Computing 6(10), 1185–1198 (2007)
5. AirLocation,
<http://www.hitachi.co.jp/wirelessinfo/airlocation/index.html>
6. Arora, A., Ferworn, A.: Pocket PC Beacons: Wi-Fi Based Human Tracking and Following. In: 2005 ACM Symposium on Applied Computing, pp. 970–974 (2005)
7. Niculescu, D., Nath, B.: VOR Base Stations for Indoor 802.11 Positioning. In: Proceedings of the 10th Annual International Conference on Mobile Computing and Networking, pp. 58–69 (2004)
8. Taromaru, M., Ohira, T.: Electronically Steerable Parasitic Array Radiator Antenna. Electronics and Communications in Japan (Part II: Electronics) 87(10), 25–45 (2004)
9. Fon, <http://www.fon.com>
10. HyperLink directional antenna specification,
http://www.l-com.com/multimedia/datasheets/DS_HG2415G-NF-5PK.PDF

Bluetooth Tracking without Discoverability

Simon Hay and Robert Harle

University of Cambridge Computer Laboratory
`{sgeh3,rkh23}@cam.ac.uk`

Abstract. Outdoor location-based services are now prevalent due to advances in mobile technology and GPS. Indoors, however, even coarse location remains unavailable. Bluetooth has been identified as a potential location technology that mobile consumer devices already support, easing deployment and maintenance. However, Bluetooth tracking systems to date have relied on the Bluetooth inquiry mode to constantly scan for devices. This process is very slow and can be a security and privacy risk. In this paper we investigate an alternative: connection-based tracking. This permits tracking of a previously identified handset within a field of fixed base stations. Proximity is determined by creating and monitoring low-level Bluetooth connections that do not require authorisation. We investigate the properties of the low-level connections both theoretically and in practice, and show how to construct a building-wide tracking system based on this technique. We conclude that the technique is a viable alternative to inquiry-based Bluetooth tracking.

1 Introduction

Location tracking is fast becoming an essential service for mobile devices. Underpinning this trend is the ubiquity and accuracy of GPS. However, GPS continues to struggle indoors due to the failure of satellite signals to penetrate buildings. To address this shortcoming, there have been many attempts at indoor location tracking, achieving a wide range of accuracies [1]. None of the proposed solutions have been widely adopted, primarily due to the cost of deploying and maintaining building-wide location technology. We have therefore become interested in finding a reliable location technology that is viable for quick deployment across entire buildings using standard infrastructure. This rules out fine-grained tracking systems such as the Bat system [2,3] since these require the retro-fit of dedicated infrastructure and the need to issue building occupants with custom tracking devices.¹

From our experiences with indoor location in many forms over many years, we know that even coarse location can provide useful location-aware applications. The Active Badge project [4] was adopted by many at the University of Cambridge because the room-level location it provided was sufficient to enable a key

¹ Our experience with the Bat system is that users tend to forget to wear their Bats. Additionally, there is a constant maintenance task in keeping Bats operational (especially with respect to battery power) and keeping the model of the world up-to-date.

application: finding co-workers quickly. This simple application demands only that users can be reliably located with room-level accuracy. The usage of that system waned because users forgot to carry their badges, didn't take responsibility for replacing batteries, and found that tracking failures were common (badges were easily obscured by clothing).

We are seeking a technique that will permit room-level tracking, both for colleague searching and for building optimisation. With the latter, we are particularly interested in dynamically optimising heating, cooling and computing devices based on room usage — see [5] for more details. To achieve this we must satisfy the following goals:

- The system must be able to locate users to a small area, ideally placing them in a single room.
- The system must be able to track users as they move to permit dynamic optimisation on timescales of the order of seconds.
- The system must be able to track multiple users simultaneously.
- The system must be easily adopted by a very high percentage of building users and it must avoid the issue of users forgetting to enable tracking in some way.

In order to meet the final criterion, we wish to derive indoor location from the mobile telephones that people carry. These devices are ubiquitous in many societies, feature an increasing array of communications technologies, are carried everywhere by their owners and responsibility for charging them lies with the user.

To obtain in-building tracking using unmodified mobile telephones, we wish to overlay location tracking on established indoor communications technologies. Such tracking has been demonstrated using signals from the mobile telephony networks, using WiFi and using Bluetooth. At present, positioning from the telephony networks is too coarse for in-building location [6]. WiFi-based tracking has received much attention in recent years [7,8], but it is not ideal for general tracking. It is associated with high power demands, difficulty in set up and maintenance, small market penetration of suitable (WiFi-enabled) handsets and difficulty in convincing users to leave handset WiFi turned on.

Bluetooth has more potential. It was designed for low power devices and almost all deployed handsets support it. For this reason, it has been a popular choice for other location tracking projects [9,10,11,12,13,14,15,16,17]. These projects all locate mobile devices based on the results of constantly scanning for mobile devices from fixed devices ('beacons'), or continually scanning for beacons from mobile devices. The disadvantages of tracking by continual scanning are studied in detail in the following section, but the key issues are the length of time each scan takes (up to 10.24s) and the requirement for the device being scanned to be discoverable (this is a serious privacy risk in the beacons-scan-handsets scheme [18] and is not even possible on many of the latest phones).

In this paper, we explore a novel way to track almost any Bluetooth device (following a one-off registration) without making it discoverable. The remainder of the paper looks at quantifying the properties of Bluetooth that we exploit

when tracking, before going on to consider the architecture of a large-scale tracking system based on the technique.

2 Bluetooth Tracking of Mobile Devices

Most Bluetooth tracking systems are proximity-based [19,20]. That is, if a user can be contacted by a base station, then the user is coarsely located to the base station position. Bluetooth was designed as a short-range communications system with range of comparable size to a room (class 2 Bluetooth devices have a nominal range of 10m in free space, less indoors), so proximity-based location is simple to implement and relatively reliable. Bluetooth tracking has also been demonstrated using the RSS measurement techniques first developed for WiFi location systems [16,21], either based on radio propagation models or ‘fingerprinting’ (matching current radio conditions to a previously-measured radio map).

As mentioned in the previous section, most of these methods locate a device using the inquiry, or scan, mode of Bluetooth. In this mode, a base station transmits a discovery packet on each of 32 radio channels. Devices set to ‘discoverable’ respond to this packet, identifying themselves. However, the response follows a random delay in order to minimise the chance of response collisions when multiple devices respond. The result of this protocol is that an inquiry must run for 10.24s to reliably detect all devices in range (and longer still if the radio conditions are unfavourable).

Initially, many systems tracked mobile devices by continually issuing inquiry packets from a network of fixed beacons. This has the advantage that no custom code need be deployed on the handset, but is often perceived as a privacy risk since anyone can track a handset by creating their own network of beacons. Additionally, handset manufacturers are increasing security by changing the ‘discoverable’ mode to be a time-limited handset state; this is the case for the iPhone and the G1. Thus more recent tracking attempts have concentrated on the mobile handset scanning for the fixed beacons. This is more secure since Bluetooth does not require that a scan packet identify its source address. However, it requires custom application code on the handset, will typically draw more power, and requires a data communications channel to publish positions. Regardless of these issues, both schemes suffer from a series of further problems:

High Tracking Latency. Since each scan can take 10.24s, these systems have a very slow update rate which does not support dynamic tracking.

Devices in the System must be Discoverable. In order for a scan to locate a nearby device, that device must be discoverable. It therefore announces itself to the world, and becomes a target for hackers regardless of whether it is a handset or a beacon.

Connection Disruption. Each scan must flood the Bluetooth channels in turn, disrupting normal Bluetooth communications whilst scanning a channel that is in use.

Ideally, then, Bluetooth tracking would not involve the inquiry process nor require any device to be permanently discoverable.

3 Inquiry-Free Tracking

Inquiry-based tracking detects proximity between handsets and fixed beacons by continually scanning for all devices. We introduce the alternative notion of *connection-based* tracking, whereby two specific devices are characterised as proximate if one can connect to the other i.e. if beacon B can connect to handset H then B and H are proximate.

In many senses this is notionally similar to an inquiry-based system where each inquiry targets a specific device rather than every device. This has the obvious disadvantage that multiple such inquiries would be needed to find a multitude of local devices. However, the specificity also allows devices to be targeted quickly rather than with extended broadcasts i.e. a single inquiry should complete faster, allowing for multiple inquiries.

The key motivation for investigating connection-based tracking is that its Bluetooth implementation can potentially address the inquiry-based shortcomings we have already identified. The remainder of this Section is devoted to characterising the relevant properties of Bluetooth that permit this connection-based tracking. This characterisation is done both theoretically and experimentally. For all testing we used a desktop Linux machine (running Ubuntu 8.04 with the BlueZ 3.26 Bluetooth stack) with an attached class 2 generic Bluetooth 2.0 USB dongle and a range of mobile telephones taken as representative of those available on the market (see Table 1).

3.1 Bluetooth Connections

The Bluetooth standard defines three different types of connection which are layered to form the Bluetooth stack. Only two of the three are appropriate for connection-based tracking. The most fundamental is the Asynchronous Connectionless Link (ACL). No more than one ACL can exist between any two Bluetooth devices at any given time, and it must be established before any other connection can be made. An ACL will disconnect only when no higher-layer connections have existed for a certain time (2s seems to be the preferred value of current stacks).

Directly above the ACL is the Logical Link Control and Adaptation Protocol (L2CAP) layer. This is a packet-based layer that provides guaranteed packet

Table 1. The test handsets used

Handset	Description
T-Mobile G1	Android
Apple iPhone 3G	iPhone OS 2.2
Nokia 6300	Series 40 3rd Edition, Feature Pack 2
Nokia N80	Series 60 3rd Edition (Symbian OS 9.1)

sequencing and a selectable degree of delivery reliability. Once established, an L2CAP connection remains open until either end explicitly closes it, or the Link Supervision Time Out (LSTO) expires. The LSTO is the time for which communicating devices are out of range before the L2CAP connection is destroyed. The default LSTO is 20s in many stacks, but can be configured dynamically per-ACL.

Above L2CAP sits the Radio Frequency Communications (RFCOMM) layer, which is the reliable stream-based protocol used by most Bluetooth applications. It represents the type of connection most people mean by ‘Bluetooth connection’. It is almost never feasible to use this layer for connection-based tracking due to authorisation requirements (see the subsequent Section) and we do not consider it in any detail here.

3.2 Connection Authorisation

The biggest obstacle to connection-based tracking using RFCOMM connections is the requirement for explicit *pairing* of every handset with every fixed beacon. The pairing process is governed by the Bluetooth security manager and involves creating a shared secret (a passphrase or PIN) between the two devices. Because each pairing operation requires human input, it is not practical to pair every handset with a large deployment of beacons.

There are some communications services, however, that do not usually require this level of authentication. In particular, the creation of an ACL and a basic L2CAP connection is *almost* universally authorisation-free. Although the resultant connections are limited in use for communications (they support little more than low-level testing) they are sufficient for tracking usage because if they are successfully established, the two devices must be within range of each other. Additionally, the low-level tasks they do support, such as RSSI measurement and L2CAP echo requests (directly analogous to the familiar ICMP ping packet in IP), allow continual monitoring of the connection.

3.3 Connection Time

A connection-based tracking system will constantly make connections between devices. The expected latencies in tracking will be dictated by the connection times, which we seek to quantify here.

Bluetooth uses frequency hopping for channel robustness. Every device continually retunes its transceiver to one of 79 Bluetooth channels. The precise sequence of changes is derived from its address and its local Bluetooth clock. For two devices to communicate they must have the same hopping sequence and phase at any given moment. To reach this state, Bluetooth defines a protocol that the two devices must follow. The protocol is known as *paging* and involves the master device sending search packets (‘pages’) addressed to the *slave* device until it replies. It is useful to consider the behaviour of the master and the slave separately, and Figure 1 may help in understanding the process.

Slave. The slave device periodically listens for page requests on a particular radio channel for 11.25ms. A total of 32 channels are used for paging, and the

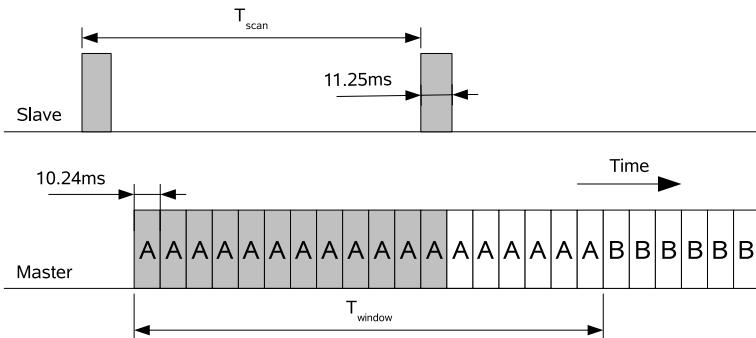


Fig. 1. The paging process. A slave periodically listens on a single frequency for 11.25ms. When paging, the master pages the 16 A frequencies in turn, each cycle taking 10.24ms. After T_{window} seconds, it repeats using the B frequencies. In this example, shaded cycles for the master indicate the cycles needed to connect to the slave shown.

frequency the slave listens on is changed every 1.28s according to a sequence also derived from its clock and its device address. If the slave does not detect a page, it sleeps for set period of time, T_{scan} . The Bluetooth specification defines three *SR* modes which a device can adopt and which provide a limit on T_{scan} . These modes are R0 ($T_{\text{scan}} = 0$), R1 ($T_{\text{scan}} \leq 1.28\text{s}$) and R2 ($T_{\text{scan}} \leq 2.56\text{s}$). The default mode is R1, but some mobile devices adopt R2 to save power.

Master. With each page sent out, the master needs to estimate the frequency that the slave will be listening on. The device's address tells it the hopping sequence, but it can only know the current sequence position by having an estimate of the Bluetooth clock on the slave. Assume temporarily that it has such an estimate. It then computes the most likely frequency to transmit on. However, in the 11.25ms that the handset listens for, it has time to send pages to 16 channels. Thus it chooses a set of 16 frequencies that are adjacent in the hopping sequence and centred on its best estimate of the listening frequency. This set of 16 frequencies is known as 'train A' and allows for a degree of error in the estimate of the slave's clock. It then cycles over this train of frequencies continuously for some $T_{\text{window}} \geq T_{\text{scan}}$ seconds. Assuming the slave is listening on a train A frequency, it will hear the page and respond, implicitly syncing the two devices. On average this should take $\frac{1}{2}T_{\text{scan}}$ to complete.

How does the master get that important estimate of the slave's clock? Normally it gets this by completing a lengthy inquiry process. If it does not have an estimate (e.g. inquiry is avoided as in the connection-based technique) it is forced to centre train A on a randomly chosen paging frequency. There is now a 50% chance that the slave will not be listening on a train A frequency. After T_{window} seconds without a reply, the master repeats the entire process

using the other 16 frequencies; ‘train B’. On average, the pager will have to use train B half of the time. Thus, the average connection time is

$$\langle T_{conn} \rangle = \frac{1}{2} \cdot \frac{1}{2} T_{scan} + \frac{1}{2} (T_{window} + \frac{1}{2} T_{scan}) \quad (1)$$

$$= \frac{1}{2} (T_{scan} + T_{window}) \quad (2)$$

and $T_{scan} \leq 2T_{window}$.

So, for default values of $T_{scan} = 1.28s$ and $T_{window} = 2.56s$ a successful page will take 0.64s on average if we know have a good clock offset estimate between master and slave, and 1.92s on average if not. In the worst possible case, a page should complete within 5.12s, which is still twice as fast as an inquiry.

It should be noted that once a connection is established, the master can cache the clock offset for the slave and use it for subsequent connections to the slave. Over time, however, the clocks drift apart and the estimate will degrade.

None of this analysis, however, incorporates any setup overheads at the master or slave once paging has succeeded. We expect there to be a systematic increase to the connection times. Figure 2 shows the connection times to an Apple iPhone. For each datum, a new connection was created and its creation time measured using the UNIX C function `gettimeofday()` on the connecting system. The start of each connection was spaced by 5000ms, chosen to be indivisible by 1280ms.

Since the iPhone has $T_{scan} = 1.28s$, each subsequent connection attempt would start at a different point in the page scan cycle of the iPhone². Ignoring connection overheads, the connection time should move linearly between zero (connection starts just as the slave’s page cycle does) and 1280ms (the connection starts just after the cycle ends). We observed that the experimental connection times incorporate an additional offset of approximately 200ms, which is the connection setup overhead. Additionally, one connection attempt failed altogether, resulting in a connection time of two page slots.

Ideally, connection would be near instantaneous for connection-based tracking. With unmodified handsets, it is realistic to expect connection times to be bounded by 1.28s. Until a connection is established or 1.28s has elapsed, a given beacon will not be able to verify whether or not a handset is in range. The equivalent figure for inquiry-based searching is 10.24s, although as noted previously, multiple handsets can be queried simultaneously.

3.4 Disconnection Time

Bluetooth connections can be either be shutdown in an orderly fashion by either end or severed when out of range. For the latter, a complete disconnection occurs after LSTO seconds have elapsed without communication. However, we have experimented with our test phones and found that when a connection was lost

² Note that BlueZ cached the clock offset so we expect all connections to occur in train A.

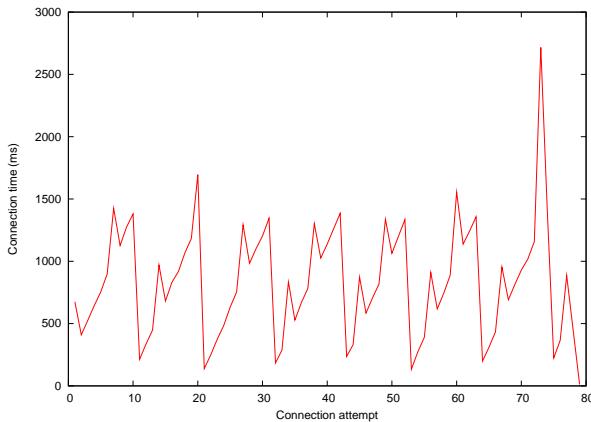


Fig. 2. Experimentally measured L2CAP connection times for the iPhone

due to range, the connection did *not* re-establish itself if the handset re-entered the radio range before the LSTO completed. This means that if a handset leaves communications range of a host and returns within the LSTO, we may not be able to ‘see’ it from the original host until the LSTO has expired. The LSTO value can be set dynamically and it is logical to set this to a small value since we expect frequent disconnection resulting from mobility.

3.5 Connection Monitoring

Once a handset and beacon are connected, a forced disconnection signals that they are no longer co-located. Whilst this is useful information, it may be possible to infer more by monitoring the connection quality.

In general there are three metrics used to monitor a Bluetooth connection: Received Signal Strength Indicator (RSSI); Link Quality (LQ); and echo response time. The standard does not require a Bluetooth chip to report the RSSI or LQ, although most do. We assume that the Bluetooth chip of the beacons, where we wish to measure the metrics, can be chosen to support these features.

The advantage of using the RSSI or LQ measurements are twofold. Firstly, they do not involve sending extra data wirelessly and so do not consume power at the mobile handset. Secondly, our experiences indicate that many manufacturers update the reported RSSI value at 1Hz or even faster, allowing for fast monitoring of a link if so desired.

An alternative method to monitor the connection is to use the round-trip time for an echo packet (c.f. ICMP ping). Table 2 shows the experimentally-measured maximum rate of echo requests for each of our test handsets when close to the sending host.

We have found the round-trip time to be of the order of 40ms for a strong connection, so update rates faster than 1Hz are easily achievable. However, we

Table 2. Experimentally measured ping rates for different handsets

Handset	Echo process rate (Hz)
T-Mobile G1	13.00
Apple iPhone 3G	20.14
Nokia 6300	21.97
Nokia N80	20.26

note that this approach does require the handset to be more active and therefore reduces its battery lifetime; we investigate this shortly.

RSSI/LQ And Proximity. Having established that the metrics can be read, there is value in asking how useful they are. There is a growing body of work on the use of RSSI values for location determination through fingerprinting techniques [19,20,22]. We note that ideas described thus far could be used to gather RSSI readings at a higher rate than the usual approach (using inquiries) and so may improve those methods. However, fingerprinting requires detailed surveying (and continual re-surveying) which does not meet our goals of minimal maintenance and we do not pursue fingerprinting here.

There have also been many attempts to relate the RSSI or LQ values to absolute distances with which to derive position estimates [21]. We feel that such approaches have demonstrated only limited success to date and opt for much coarser use of the RSSI.

We surveyed a series of offices using an iPhone to measure RSSI and a co-located Bat [3] to simultaneously measure locations to within a few centimetres. The traces from three hosts are shown in Figure 3. Figure 4 shows the observed relationship between the reported RSSI and the distance from the fixed Bluetooth master, based on data collected for 10 different masters. Note that an RSSI of -13 is actually a disconnection. Madhavapeddy and Tse performed further studies of Bluetooth propagation in our offices using a similar technique [23]. It is clear from Figure 4 that there is no deterministic relationship between distance and RSSI, but that there is a qualitative trend (larger distances are associated with more negative RSSI values). We simply use this trend to associate a falling RSSI reading with a handset leaving the area, which is generally sufficient for our purposes.

3.6 Battery Costs

Battery power is always a key issue for mobile devices. Providing any new functionality such as tracking ability is useless if users disable it to save power. It is, however, very difficult to accurately model the energy cost of the components of a generalised handset. Instead we created a custom rig for the T-Mobile G1 and Nokia N80 handsets that allowed us to accurately sample instantaneous power draw at 1kHz. The apparatus involved a replacement battery connected to the real one via circuitry to monitor the voltage and current, based on the design used by Hylick et al. [24] to analyse hard drive energy consumption. We used

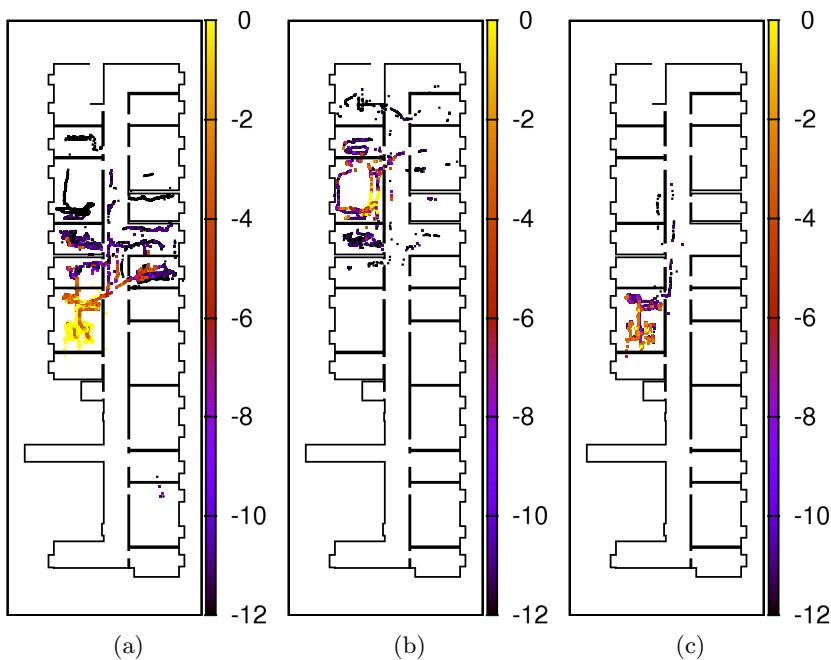


Fig. 3. Experimentally measured RSSI values from three hosts, plotted against locations recorded using the Bat system

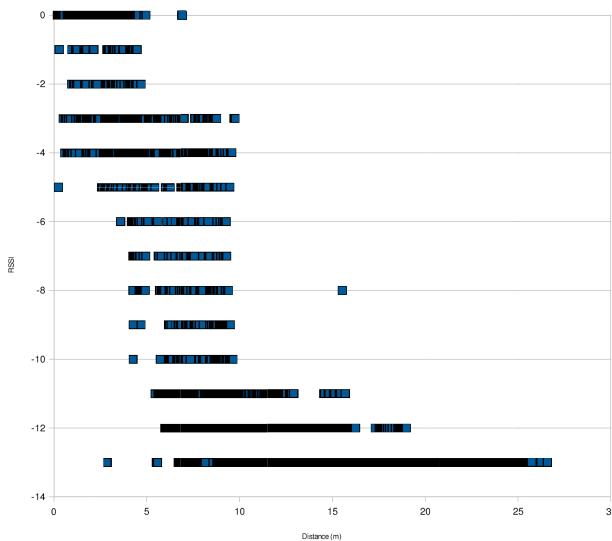


Fig. 4. Measured RSSI against euclidean distance from the host

this rig to investigate the power draw associated with monitoring a connection using either RSSI or echo response times.

The results are reported in Table 3, each given a unique test ID for reference here. Tests 1–3 provide baseline power draws; tests 4–6 provide power draws for continuous connection states; tests 7–15 provide power draws for discontinuous state (e.g. the connection was created, an RSSI taken and the connection shut down until the next update time). For all of these tests, we ensured that the handset screen was powered off, no SIM card was in place, no other connections were active and no applications (apart from any operating system requirements) were running on the devices. The power draws were computed by averaging over appropriate time periods (mostly 10 minutes).

The results are broadly as expected. A handset that is page scanning need only take action every T_{scan} seconds, and then only for 11.25ms. Thus we observe only a small associated cost. Saturating a permanent connection with echo requests (test 5) is very energy-intensive, whilst continually measuring its RSSI lies between the two extremes (and equates to the energy cost of maintaining an L2CAP connection).

When considering less frequent monitoring, instead of holding a connection open it may be more energy efficient to connect, take the reading, and then disconnect. Tests 7–15 concerned connection monitoring with such a model and we observe that, within error bounds, there is little to choose between using an echo packet or reading an RSSI. The main cost lies in creating and destroying a connection, which is relevant to both metrics. These data show that if the update period exceeds 5s, it is more efficient to break the connection and reform it when it is next needed than maintain the connection. For period less than

Table 3. Experimentally measured power draws

Test ID	Handset State	G1 (mW)	N80 (mW)
1	Idle	12.81	19.44
2	WiFi connected but idle	170.66	438.08
3	Bluetooth on, discoverable, unconnected	15.45	22.17
4	Bluetooth on, continually being scanned by host	16.07	31.80
5	Continuous Bluetooth echo at maximum rate	321.07	234.43
6	Continuous Bluetooth RSSI measurement	74.97	89.20
7	Bluetooth echo every 30s (with reconnect)	23.76	26.17
8	Bluetooth echo every 20s (with reconnect)	25.19	28.17
9	Bluetooth echo every 15s (with reconnect)	28.02	—
10	Bluetooth echo every 10s (with reconnect)	30.53	42.27
11	Bluetooth echo every 5s (with reconnect)	40.13	50.61
12	Bluetooth RSSI every 30s (with reconnect)	29.93	28.05
13	Bluetooth RSSI every 20s (with reconnect)	35.86	29.93
14	Bluetooth RSSI every 10s (with reconnect)	47.59	36.04
15	Bluetooth RSSI every 5s (with reconnect)	75.72	51.88

5s it is unlikely to be possible to disconnect since the underlying ACL does not close with the L2CAP connection.

3.7 Connection Saturation

The Bluetooth specification states that a Bluetooth device can maintain up to seven simultaneous connections to other devices. To reduce power consumption and complexity, however, there are reports that some handset manufacturers have reduced this limit in their handsets. We attempted to verify this, but were unable to find a handset that did not support the full seven connections. We speculate that handset capabilities are now sufficiently advanced to permit full specification compliance without adverse impact.

4 A Tracking System

Having characterised the fundamental properties of Bluetooth in the context of connection-based tracking, we now turn to the issue of exploiting these properties to build a wide-area tracking system.

4.1 System Architecture

We assume a large building, within which a large number of inexpensive Class 2 Bluetooth radios are distributed. Each radio must be connected to a host machine of fixed, known location and the host machines must be networked together in some way. In most scenarios, we envisage simply adding a Bluetooth dongle to desktop PCs, making deployment both quick and easy. We refer to these fixed machines with Bluetooth simply as *hosts*.

A central system that can connect to any host maintains a database of handset device IDs, their owners and any known properties of the handset (T_{scan} etc). Each entry in the database is the result of a one-off registration of the handset with the system.

The central system then maintains a model of where people are and uses this to instruct hosts to monitor associated handsets at appropriate times. For example, as a user left their office the system may instruct hosts at either end of the corridor to monitor the handset in order to infer the direction in which he or she went.

4.2 Distributed Clock Offsets

This system involves continually connecting and disconnecting hosts and handsets and so it is extremely important to minimise the connection times. From the analysis of Section 3.3, we seek to ensure that the host always incorporates the handset's current listening frequency within the paging train A. If only one host was involved, this could be achieved by accepting a lengthy first connection and thereafter caching the clock offset associated with that handset.

However, the connection/disconnection here involves many hosts, typically in sequence. We propose that, once a host discovers the offset between its local

Bluetooth clock and that of a handset, it distributes the offset to other hosts. For this to work, each host must be able to estimate the offset between its local Bluetooth clock and that of the handset. The estimate must be sufficiently accurate that the host predicts a listening frequency for the handset that is within 8 frequency hops of the true value. This ensures that the true frequency will be in train A.

The handset advances its listening frequency every 1.28s. Thus, hosts other than the original may incorporate an error of up to $8 \times 1.28 = 10.24s$ in their clock offset and still connect using train A. Therefore distributing clock offsets can be trivially achieved by using NTP to synchronise the system clocks of all hosts, and having them report two offsets: the clock offset between their system clock and their Bluetooth clock, and the offset between their Bluetooth clock and that of the handset.

We verified this approach in principle. We first time-synchronised two desktop machines using NTP. We connected both machines in turn to a particular handset and measured the offsets between the local Bluetooth clock and both the handset clock and the local system clock. We were able to predict the offsets of one machine given only the data from the other to an accuracy far greater than was required to connect in train A. In practice, however, we have found that the BlueZ code that permits specifying the clock offset to use for a particular connection is not yet functional. We believe, therefore, that it is justifiable to assume that all connections take place within T_{scan} seconds, which gives an expected connection time of 0.64 seconds in the default case.

4.3 Search

At its simplest level, this setup can be used to emulate an inquiry-based system for a single user. A subset of hosts that represents the minimum set needed to cover the entire building would simultaneously be instructed to connect to the associated handset. Any successful host(s) then report back, allowing the central system to localise the user. Those devices that are out-of-range and still attempting to connect after localisation has completed can simply cancel their paging, allowing the user to be localised quickly and the next update to begin.

Whilst this algorithm will localise the user significantly faster than an equivalent inquiry-based system, it does not scale well with the number of users. An inquiry-based system may only provide position updates every 10.24s, but it handles multiple handsets simultaneously. We would expect a connection-based system used in the manner described to be able to localise around ten handsets in the time that an inquiry-based handset would localise almost all handsets.

We can start to address this shortcoming in two ways. The first is to use the connection monitoring techniques of Section 3.5 to identify those users are moving away from their nearest host (the current ‘home’ host) and who therefore need to be tracked. From our previous analysis of people’s working habits we know that the vast majority of workers will be sedentary at any given time [5] and therefore the number of mobile users should be small enough to localise all users within reasonable time-frames.

The second is to limit the extent of search where possible. If, for example, a user's current home reported a very weak RSSI (indicating lost proximity) t seconds ago, the only hosts that can reasonably observe him now are those within vt of the reporting host, where v is the speed of the user. This type of information permits simultaneous searching for different handsets known to be in different building areas. This, combined with the fact there may be disjoint subsets of hosts that completely cover any given region, should permit tracking of many users simultaneously with update rates similar to that of inquiry-based tracking. However, the system will struggle to cope with many users moving at once or collected within a small area with few covering hosts. In such a scenario, the update rate for tracking will degrade, but the system will self-heal when the crowd disperses.

4.4 Bootstrapping

The discussion so far has ignored how a registered handset returning to the building will get discovered before being tracked. We currently have two options:

Constant Round-Robin Polling. The obvious solution is to continually cycle through all the registered handsets that are not being tracked and poll them from a variety of hosts. Because a connection can take up to 5.12s to connect (worst case), cycling through 100 registered handsets can take over eight minutes to complete. Additionally, it places an extra load on hosts that may be in use for monitoring tracked handsets, reducing update rate.

Out-of-Band Events. Our preferred solution is to use out-of-band events to signal that a particular handset is likely to be in the building. These events can be easily generated from computer system events (e.g. the associated user logs in) and from building security (e.g. the user gains access to the building by swiping their ID card). It may also be possible to infer presence based on cell-ID from the mobile telephony network.

Additionally, we note that many people follow a daily routine which there is potential to learn autonomously. Doing so would allow prediction of when a user is likely to arrive, permitting for targeted polling of the subset of expected handsets rather than all handsets.

4.5 Inverted System

There is, of course, the possibility to invert the system and have the handset as the master, connecting to the hosts as slaves. This would require custom software on the handset and a wireless data channel to receive a model of the host positions and addresses. This has the advantage that no bootstrapping would be necessary since the handset localises itself. However, care would be needed to ensure that a given host was not saturated with connection requests from a series of nearby handsets. Hence a degree of centralised control is still likely to be needed.

We note that such inversion is common for the inquiry-based approach because it permits the handset to localise itself and still remain anonymous. Unfortunately, in the inverted connection-based model, whenever a handset connects to a host it must explicitly identify itself and anonymity is broken. Given also that paging is more power-hungry than page scanning, the inverted technique is not compelling for connection-based systems.

5 Discussion

We believe we have demonstrated that connection-based tracking is a viable alternative to inquiry-based tracking, although neither is ideal. We characterise the advantages of connection-based tracking as:

Faster Update Rate. Connections can generally be *established* faster than a scan can be completed. Connections can be *monitored* at a very fast rate.

Variable Update Rate. The rate can be adapted to suit the context of the mobile device and preserve battery life. If, for example, we believe the user to be sedentary, we can choose to reduce the query rate.

No Scanning. The lack of need for constant inquiries is a significant enhancement to Bluetooth usage, security and privacy.

Privacy. Users retain their right to opt out of being tracked by switching off Bluetooth on their handset.

There are, however, a number of drawbacks:

Devices must be Known. Connection establishment requires that at least one party knows of the existence of the other in order that it can issue a connection request; however, this convenience drawback is also a privacy advantage.

Security Policies are not Universal. Not all Bluetooth security policies permit L2CAP connections without authorisation. Similarly, not all Bluetooth chips permit querying the connection metrics such as RSSI. However, such queries are performed on the beacon's Bluetooth chip, which we are free to choose.

Connection Limits. Care must be taken to ensure Bluetooth limits on the number of connections are not reached.

Table 4 compares connection and inquiry based tracking side by side. Ultimately, Bluetooth was designed as a short-range wireless communications protocol and as such it is unreasonable to expect it to provide ideal location tracking. Nonetheless, both inquiry-based and connection-based tracking are at least feasible, and both can be implemented without modifying handset software or hardware, which is an important consideration.

In terms of security and privacy, we note that any handset with Bluetooth turned on is likely to be trackable by the techniques in this paper if an observer knows the Bluetooth address of it (this requires the handset to have trusted the

Table 4. Comparison of inquiry-based and connection-based tracking

	Inquiry-based Handset-scans-Host	Inquiry-based Handset-scans-Handset	Connection-based
Works on Unmodified Handsets	Most	Some	Almost all
Requires Custom Handset Software	Yes	No	Yes
Deployability	Med	High	High
Requires Discoverable Handset	No	Yes	No
Requires Separate Data Channel	Yes	No	No
Location Update Frequency	~0.1Hz	~0.1Hz	Varies (0-20Hz)
Supports Dynamic Update Rates	<0.1Hz	<0.1Hz	Yes
Scalability	Simple	Simple	Complex
User Privacy	High	Low	Med-High
Handset Power Drain	High	Low	Varies (Med)

third party at some point in the past). This means a third party can potentially track specific handsets without their knowledge.

Looking forward, we believe that connection-based tracking will improve in the short-term as handsets advance. Already we see handsets that incorporate accelerometers (which could be used to infer movement or the lack of it, or to dynamically update the monitoring rate). Similarly, advanced development platforms for mobile devices are emerging and these will hopefully provide low-level access to handset subsystems such as Bluetooth. This in turn will permit handsets to be more active in the location process since APIs and behaviours will standardise and applications will be easier to deploy.

6 Conclusions and Further Work

In this paper we have identified and evaluated an alternative to inquiry-based Bluetooth tracking in the form of connection-based tracking. This form of tracking does not demand that devices be permanently discoverable (something which is no longer possible on many mobile devices), nor that the mobile handsets are modified in any way.

We have studied the properties of Bluetooth that enable connection-based tracking both theoretically and experimentally and concluded that there is potential for the approach. There is little doubt that inquiry-based and connection-based tracking have their advantages and disadvantages (many of which do not overlap). We recognise that neither makes the ideal tracking system, but both have very little in the way of deployment costs, allowing the construction of large testbeds. We intend to develop the tracking algorithms and evaluate them by

deploying a connection-based tracking system across our building and encouraging many users of it.

References

1. Hightower, J., Borriello, G.: Location systems for ubiquitous computing. *Computer* (January 2001)
2. Addlesee, M., Curwen, R., Hodges, S., Newman, J.F., Steggles, P., Ward, A., Hopper, A.: Implementing a sentient computing system. *IEEE Computer* 34(8), 50–56 (2001)
3. Harter, A., Hopper, A., Steggles, P., Ward, A., Webster, P.: The anatomy of a context-aware application. In: *MobiCom 1999: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pp. 59–68 (1999)
4. Want, R., Hopper, A., Falcao, V., Gibbons, J.: The Active Badge location system. *ACM Transactions on Information Systems* 10(1), 91–102 (1992)
5. Harle, R., Hopper, A.: The potential for location-aware power management. In: *UbiComp 2008: Proceedings of the 10th International Conference on Ubiquitous Computing* (September 2008)
6. Rehman, W., Lara, E., Saroui, S.: Cilos: a CDMA indoor localization system. In: *UbiComp 2008: Proceedings of the 10th International Conference on Ubiquitous Computing* (September 2008)
7. Bahl, P., Padmanabhan, V.: RADAR: an in-building RF-based user location and tracking system. In: *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings*, vol. 2, pp. 775–784. IEEE, Los Alamitos (2000)
8. Youssef, M., Agrawala, A.: The Horus WLAN location determination system. In: *MobiSys 2005: Proceedings of the 3rd international conference on Mobile systems, applications, and services* (June 2005)
9. Anastasi, G., Bandelloni, R., Conti, M., Delmastro, F., Gregori, E., Mainetto, G.: Experimenting an indoor Bluetooth-based positioning service. In: *Proceedings of the 23rd International Conference on Distributed Computing Systems Workshops, April 2003*, pp. 480–483 (2003)
10. Cheung, K., Intille, S., Larson, K.: An inexpensive Bluetooth-based indoor positioning hack. In: *UbiComp 2006: Proceedings of the 8th International Conference on Ubiquitous Computing Extended Abstracts* (2006)
11. Bargh, M., Groote, R.: Indoor localization based on response rate of bluetooth inquiries. In: *MELT 2008: Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments* (September 2008)
12. Jevring, M., de Groote, R., Hesselman, C.: Dynamic optimization of Bluetooth networks for indoor localization. In: *AASN 2008: First International Workshop on Automated and Autonomous Sensor Networks* (2008)
13. Huang, A.: The use of Bluetooth in Linux and location aware computing. Master of Science dissertation
14. Bruno, R., Delmastro, F.: Design and analysis of a Bluetooth-based indoor localization system. *PersonalWireless Communications*, 711–725 (2003)
15. Hallberg, J., Nilsson, M., Synnes, K.: Positioning with Bluetooth. In: *ICT 2003: Proceedings of the 10th International Conference on Telecommunications*, vol. 2(23), pp. 954–958 (2003)

16. Pandya, D., Jain, R., Lupu, E.: Indoor location estimation using multiple wireless technologies. In: PIMRC 2003: 14th IEEE Proceedings on Personal, Indoor and Mobile Radio Communications, August 2003, vol. 3, pp. 2208–2212 (2003)
17. Naya, F., Noma, H., Ohmura, R., Kogure, K.: Bluetooth-based indoor proximity sensing for nursing context awareness. In: Proceedings of the 9th IEEE International Symposium on Wearable Computers, September 2005, pp. 212–213 (2005)
18. Jakobsson, M., Wetzel, S.: Security weaknesses in Bluetooth. In: Naccache, D. (ed.) CT-RSA 2001. LNCS, vol. 2020, pp. 176–191. Springer, Heidelberg (2001)
19. LaMarca, A., Chawathe, Y., Consolvo, S., Hightower, J., Smith, I., Scott, J., Sohn, T., Howard, J., Hughes, J., Potter, F., Tabert, J., Powledge, P., Borriello, G., Schilit, B.N.: Place lab: Device positioning using radio beacons in the wild. In: Gellersen, H.-W., Want, R., Schmidt, A. (eds.) PERVASIVE 2005. LNCS, vol. 3468, pp. 116–133. Springer, Heidelberg (2005)
20. Graumann, D., Lara, W., Hightower, J., Borriello, G.: Real-world implementation of the location stack: the universal location framework. In: WMCSA 2004: Proceedings of the 5th IEEE Workshop on Mobile Computing Systems and Applications, pp. 122–128 (2003)
21. Gwon, Y., Jain, R., Kawahara, T.: Robust indoor location estimation of stationary and mobile users. In: INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 2, pp. 1032–1043 (2004)
22. Schilit, B.N., LaMarca, A., Borriello, G., Griswold, W., McDonald, D., Lazowska, E., Balachandran, A., Hong, J., Iverson, V.: Challenge: Ubiquitous location-aware computing and the place lab initiative. In: WMASH 2003: Proceedings of the First ACM International Workshop on Wireless Mobile Applications and Services on WLAN (2003)
23. Madhvapedy, A., Tse, A.: A study of Bluetooth propagation using accurate indoor location mapping. In: Beigl, M., Intille, S.S., Rekimoto, J., Tokuda, H. (eds.) UbiComp 2005. LNCS, vol. 3660, pp. 105–122. Springer, Heidelberg (2005)
24. Hylick, A., Sohan, R., Rice, A., Jones, B.: An analysis of hard drive energy consumption. In: MASCOTS 2008: Proceedings of the 16th Annual IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (2008)

Error Estimation for Indoor 802.11 Location Fingerprinting

Hendrik Lemelson¹, Mikkel Baun Kjærgaard², Rene Hansen³, and Thomas King¹

¹ Department of Computer Science, University of Mannheim, Germany

{lemelson, king}@informatik.uni-mannheim.de

² Department of Computer Science, University of Aarhus, Denmark

mikkelbk@cs.au.dk

³ Department of Computer Science, University of Aalborg, Denmark

rhansen@cs.aau.dk

Abstract. 802.11-based indoor positioning systems have been under research for quite some time now. However, despite the large attention this topic has gained, most of the research focused on the calculation of position estimates. In this paper, we go a step further and investigate how the position error that is inherent to 802.11-based positioning systems can be estimated. Knowing the position error is crucial for many applications that rely on position information: End users could be informed about the estimated position error to avoid frustration in case the system gives faulty position information. Service providers could adapt their delivered services based on the estimated position error to achieve a higher service quality. Finally, system operators could use the information to inspect whether a location system provides satisfactory positioning accuracy throughout the covered area. For position error estimation, we present four novel algorithms that take different features into account. Advantages of the combination of these four algorithms are explored by using a machine-learning approach. We evaluate our proposed algorithms in two different real-world deployments by using real-world data and emulation. The results show that our algorithms work independently of the environment and the positioning algorithm, and with an average precision for estimating the position error of up to 1.45 meters. The algorithms can – by adjusting parameters – realize different tradeoffs between underestimating and overestimating errors. Furthermore we comment on the algorithms’ space and time complexity.

1 Introduction

Nowadays, powerful and light mobile devices, such as smart-phones and laptops, provide nomadic users with computing and networking services wherever they go. However, the device’s usefulness can be highly leveraged by so-called *location-based services* (LBS): These services adapt their behavior proactively depending on the user’s current location. A service that informs a user about the location of nearby friends and points of interest is a good example of an LBS.

Today, the most prominent source of location information is the *Global Positioning System* (GPS) [1]. GPS works well in outdoor open-sky environments but fails to deliver reliable location information where users spend most of their time: indoors and

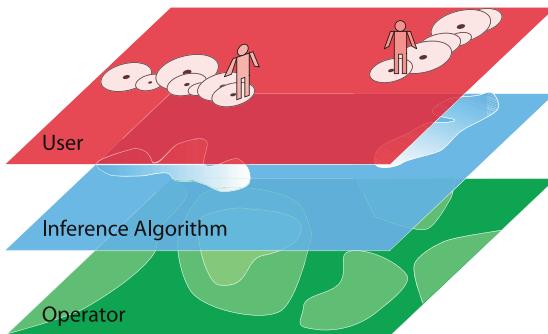


Fig. 1. The use of information about expected errors by users, inference algorithms and operators

in so-called street canyons where tall buildings prevent users from getting a line-of-sight to GPS satellites [2]. Due to these drawbacks of GPS, in recent years a lot of effort has been spent on the development of alternative systems to estimate a user's position in GPS-free areas. However, current systems either use specialized equipment to work accurately in confined indoor environments (e.g., Ubisense¹) or provide metropolitan-scale coverage at the expense of coarse positioning accuracy (e.g., Place Lab [3], RightSPOT [4]).

For three reasons, positioning based on 802.11 [5] offers a viable solution where other systems fail: Firstly, no specialized hardware is required, because, nowadays, almost all modern mobile devices are equipped with 802.11 network interfaces. Secondly, due to the proliferation of 802.11 wireless networks, radio signals from at least a few 802.11 access points can almost always be collected where people work and live in the developed world (e.g., [2] and [6]). Thirdly, 802.11 signals can be used to estimate a user's position indoors with an accuracy that is generally sufficient for most LBSs [7].

However, as with GPS, 802.11-based positioning systems comprise an inherent position error that is caused by radio propagation effects (e.g., reflection, diffraction, and scattering) and limits of the hardware (e.g., the measurement accuracy of sensors). Even worse, under sub-optimal conditions such as sparse coverage of access points or many moving obstacles that cause radio signals to fluctuate the accuracy of 802.11-based positioning systems tends to decrease.

GPS handles variations in the position accuracy by offering a measure called *Dilution Of Precision* (DOP). Computed from several different system parameters, the DOP value indicates the level of position error that has to be expected for a given estimated position. Position error estimation is mainly valuable in three ways (see Figure 1): Firstly, the user can be notified that the estimated position might contain some position error. This helps him to make better use of the LBS [8] and also avoids him becoming frustrated because of many unexpected wrong positioning results. For instance, if a friend-finder service does not only show the estimated position of a friend but also the estimated position error (e.g., as a circle around the friend's position), a user might look around in the area indicated by the position error estimate in case he is not able to

¹ <http://www.ubisense.net>

find his friend at the place where the service estimated him to be. Secondly, so-called inference algorithms utilize position estimates to derive additional information (e.g., activities or routes). Position error estimates allow the inference algorithms to assess the level of trust that these position estimates contain. As a result, the algorithms can give a higher priority to estimates with a low estimated error. This helps to deliver a higher service quality. Thirdly, operators of 802.11-based positioning systems can utilize position error estimates to optimize their system to the precision required by the application. For instance, an operator might add additional access points to the parts of the operation area that show position error estimates above a certain threshold.

In this paper, we propose several algorithms to estimate the expected position error of 802.11-based positioning systems. We focus on the static case of position estimation without any knowledge of spatial or temporal history and selected two well-studied systems proposed by Bahl et al. [9] and Haeberlen et al. [10] as positioning systems. These systems use a so-called *location fingerprinting* approach. Location fingerprinting consists of a two-stage mechanism: A *training phase* and a *position determination phase*. During the training phase, signal strength samples of nearby access points are gathered at reference spots in the operation area and stored together with their physical coordinates in a database: They are called *fingerprints*. In the position determination phase, the user's mobile device performs signal strength measurements at its (yet unknown) position and searches for similar patterns in the database. The closest match is selected, and its coordinates are returned as the position estimate.

We selected these two systems, as they represent the two main types of location fingerprinting algorithms, i.e., a probabilistic and a deterministic approach. They are easy to understand and therefore do not blur the view for the mechanisms used by our own error prediction algorithms. The two positioning systems differ in the way they process fingerprints and how they select the closest match. The system proposed by Haeberlen et al. builds signal strength distributions for each fingerprint based on the signal strength samples it contains. In the position determination phase, for each fingerprint a probability is calculated based on the currently collected signal strength samples. The probability calculation is done by utilizing Bayes' rule. The fingerprint with the highest probability is selected, and its position is returned as a position estimate. In contrast, the system proposed by Bahl et al. does not use distributions and probabilities. Instead, for each fingerprint the different signal strength samples are averaged. Based on these averages, the fingerprint that is closest to the average of signal strength samples collected during the position determination phase is selected. The calculation of the distances is done by applying the Euclidean distance.

We propose position error estimation algorithms that are able to predict the expected position error in advance, using only the collected training data, as well as algorithms that infer the expected position error from live measurements in the position determination phase.

We make the following contributions in this work: First of all, we propose novel algorithms for estimating the expected error of an indoor 802.11-based positioning system. Secondly, we provide a deep investigation by means of emulation to show that the proposed algorithms can estimate expected position errors with high accuracy. The emulation is based on collected real-world data and shows that our error estimation

algorithms work independent of the environment and the positioning algorithm. Thirdly, we evaluate the idea of using a machine-learning algorithm to combine the proposed algorithms and to find an optimal combination of the proposed algorithms. Finally, we analyze the time and space consumption of the proposed algorithms.

The remainder of this paper is structured as follows: The work of other researchers related to our own findings is presented in Section 2. In Section 3, we introduce the proposed error estimation algorithms. This is followed by a description of the experimental setup and research methodology used to evaluate our error estimation algorithms in Section 4. Subsequently, the experimental results for the different algorithms alone as well as for a combination of algorithms are presented in Section 5. In Section 6, we discuss the results of the evaluation. Finally, Section 7 concludes the paper.

2 Related Work

Some work that tries to understand the position errors for 802.11-based positioning systems already exists. However, the main difference compared to our work is that they try to form a priori analytical models in order to understand the position error in general – e.g., to understand the relationship between the number of access points or the fingerprinting grid size and the positioning error. Our algorithms instead use information from actual fingerprints and position determination measurements to estimate the expected errors in an actual deployment area and at the time of use.

Analytical models have been proposed by Wallbaum [11], Kaemarungsi et al. [12], Krishnakumar et al. [13] and Youssef et al. [14]. They propose analytical models built from information such as access point placement, system parameters and floor layout. Using these models they analyze the general impact of access point placement, number of access points, grid size, signal variance, propagation constants, and floor layout on the position error. However, as commented earlier, these models are analytical tools for doing general analysis. Our algorithms go into the opposite direction, estimating expected errors at a specific site and at a specific point in time. However, it is an interesting future path of research to further analyze the relationship between algorithms for error estimation and analytical error models.

More related to our work are the methods for estimating outdoor GSM positioning errors proposed by Dearman et al. [8]. Motivated by a study on how to visualize expected GSM position errors they propose initial methods for how to estimate such information. They propose two methods for outdoor error estimation: A machine-learning method that uses a linear regression technique on basic signal strength features and a ground truth method that requires the collection of additional test data. The outdoor accuracy of these methods is up to 50 meters when considering the median accuracy. Their machine-learning method is similar to our effort of combining features, however, the used set of features is different. Their method is based on basic features such as signal strength variance. Our results show that for 802.11 such basic features can be outperformed by more advanced features such as the proposed fingerprint clustering algorithm. Furthermore, a machine-learning method might not be optimal because it will try to optimize the average precision. In several of our machine-learning experiments this led to a high percentage of either overestimations or underestimations for the expected position error.

3 Error Estimation

In this section, we propose our four novel algorithms for position error estimation. The following two algorithms can predict the expected positioning error in advance, using only the data collected in the training phase:

A. Fingerprint Clustering

B. Leave Out Fingerprint

The two other algorithms infer the expected position error from live measurements in the position determination phase:

C. Best Candidate Set

D. Signal Strength Variance

3.1 Fingerprint Clustering

The fingerprint clustering algorithm uses the idea, that inside structurally limited areas the signals received from an access point – despite inevitable small scale fading effects – tend to be within a certain range. For instance, the signal strength measurements collected in one single office room often cover only a quite limited range of the generally possible values. As in such a case all the fingerprints collected in this office room are very similar, a positioning algorithm will hardly be able to make an exact position estimation. Instead, it will probably select one of the other fingerprints collected somewhere else in the room.

If such an area of similar signal properties has large physical dimensions, we expect the position error to be larger because generally the number of similar fingerprints is higher and also their physical distribution space is larger. We exploit this behavior in the following way. Our novel algorithm lays out a grid of cells over the complete covered area of operation of the 802.11-based positioning system. Then the following steps are performed in order to find regions of similar signal properties:

1. Initially each grid cell represents a single cluster containing all the training samples collected inside the physical area of the cell.
2. After that, the algorithm randomly selects a cluster. By using a similarity measure, the algorithm then checks for the selected cluster whether the similarity to adjacent clusters lies above a given threshold. If this is the case for any adjacent cluster, the two clusters are merged into one new cluster.
3. The previous step is repeated until, for all remaining clusters, no further clusters can be merged.
4. As a next step, the set of clusters is checked for clusters that only comprise one single cell. If such a cluster is found, it is merged with its most similar adjacent cluster without considering the threshold. The reason for this procedure is that these single cell clusters most often are a result of very time and space constrained phenomena during the collection phase of the training data, and eliminating them helps the algorithm to deliver better and more stable results.
5. In the end, each remaining cluster represents a region of similar signal properties in the area of operation. The size and shape of the region is defined by the grid cells the cluster is comprised of. The estimated error for an estimated position is deduced from the size of the region the estimated position is located within.



Fig. 2. Example for a region map created for the Mannheim testbed

An example for a region map resulting from the fingerprint clustering algorithm is given in Figure 2.

The similarity measure for a pair of clusters is calculated by comparing the distributions of signal strengths in the clusters as follows: For each access point of which signal strength measurements are contained in the samples of both clusters, and for each of the two clusters the mean and the standard deviation of the measured signal strengths are computed. These values are then used to create two normal distributions, each one representing the signal strength distribution of the specific access point and cluster. Afterwards, the intersection area of these two normal distributions is calculated (see Figure 3). This step is repeated for all access points until finally, the average size of all intersection areas is computed and interpreted as the similarity of the two clusters. Access points of which signal strength measurements are only contained in the samples of one cluster are not considered. Even though this could be easily done, e.g., by subtracting a penalty from the computed similarity value, we decided to omit such a step. Due to the spatial neighbor property of the clusters that are pairwise considered, the case where there are major differences in the set of contained access points generally is very rare.

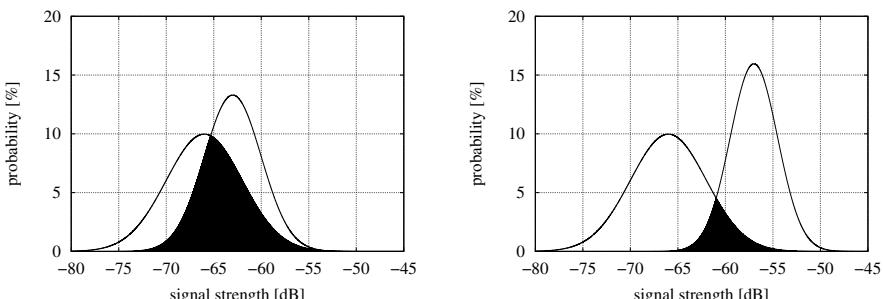


Fig. 3. Examples for the matching of signal strength distributions of one access point and adjacent clusters (good match on the left, bad match on the right hand side)

To figure out the influence of the similarity threshold on the performance of the algorithm, we conducted a pre-evaluation with varying settings for the threshold ranging from 0.4 to 0.6. In general, a higher similarity threshold results in the regions becoming smaller. This helps to decrease the discrepancy between estimated and true error in cases where the correct region is chosen. But it also increases the rate of wrong region selections (also refer to Section 5.3). From the evaluated values, for our scenarios 0.51 performs best in terms of quantitative error estimation. Using this similarity threshold, the real and the estimated position are in the same region in 90% of all cases. Additionally, in half of the remaining cases, the real position is located in a cell directly adjacent to the selected region. Only in 5% of all cases, the real position is somewhere else. Therefore we have selected the value 0.51 as the similarity threshold for all further experiments.

3.2 Leave Out Fingerprint

The leave out fingerprint algorithm estimates errors by evaluating positioning performance using only the data collected during the training phase.

The performance evaluation uses an $(n-1)$ cross evaluation with the constraint of considering only folds which contain data from a single fingerprint. This means that a positioning algorithm itself, for one current fingerprint, determines the closest match amongst all other fingerprints (the other $(n-1)$ folds). The result is a good indicator of the error that can occur if the data collected during the position determination phase is noisy.

The position error result from this evaluation is used to compute an error map where each fingerprinted position is assigned an error estimate. In the position determination phase the position estimated by the positioning algorithm is looked up in the error map, and the error estimate stored for the corresponding position is returned.

For each fingerprinted position p , the error map contains an error estimate that is created by the following four steps.

1. Create a radio map using all fingerprints except the one for position p .
2. Run emulation using m samples as test data taken randomly from the fingerprint for position p .
3. Calculate the observed errors for the position estimates from the emulation.
4. Calculate the error estimate for position p as the average plus two times the standard deviation of all the observed m errors. This method for computing the error estimate was selected because initial experiments showed that a conservative estimate gives the best results.

As a final step when all error estimates have been calculated, the error map is averaged to filter out local variations. This is performed by assigning a new estimate to each position. The new estimate is calculated as 66% of the original estimate and 33% as the average of the estimates for all neighboring fingerprints. We chose these parameters for the weights because they provide a good balance between the error estimates for the different fingerprint positions. In our evaluation this interpolation step increases the accuracy of the algorithm by several percent.

3.3 Best Candidate Set

When an 802.11-based positioning algorithm estimates a position, it generally returns only the position of the fingerprint that offers the best match to the data collected in the position determination phase. However, the best candidate set algorithm exploits the fact that positioning algorithms can return information about the second, third etc. best matches. The best candidate set algorithm uses the n best estimates returned from a positioning algorithm to estimate the expected position error. The rationale for using the n best estimates is based on the observation that positioning algorithms will often estimate a user to be at any of the nearby positions to his actual position. This is a result of the fact that adjacent fingerprinted positions will often exhibit highly overlapping signal strength properties while online samples similarly have sufficient signal strength variance to choose at random between close locations.

The best candidate set algorithm deduces an error estimate by computing the average distance between the best estimate and the next $(n - 1)$ best estimates. The algorithm contains the following steps:

1. Form the set of the n best estimates as outputted from an 802.11-based positioning algorithm.
2. Compute the distance between the position of the best estimate and all the other $(n - 1)$ best estimates.
3. Return the average distance as the estimated error.

In addition to using the average distance between the best and the $(n - 1)$ next best estimates, we also tried two other options to compute an error estimate from the best candidate set. The first option was to use the maximum distance between the best estimate and any of the $(n - 1)$ next best estimates. The second option was to compute the maximum distance between any two of the n best estimates. Both options led to more conservative error estimates but they were also more sensitive to large position estimation errors and increasing n . In effect, both options produced a higher proportion of large over-estimations when the performance of the positioning algorithm degraded or when the size of the best candidate set was increased. The average-distance approach in contrast handled these situations better. Using the average distance, it was found that n set to three gave the best overall performance. Higher values of n made the error estimates more conservative while gradually decreasing performance due to the inclusion of more faraway positions.

3.4 Signal Strength Variance

Like most 802.11-based positioning systems, the systems selected for this paper are error-prone to signal strength variations, because the position estimation process depends on probabilities or average values calculated on the signal strength samples.

The reasons for the variation of signal strength samples are two-fold: Firstly, small-scale effects and multi-path radio propagation cause signal strength values to vary significantly if the mobile device moves within the wavelength (for 802.11 the wavelength is 12.5 centimeters). So, both during the training as well as during the position determination phase, even if the mobile device is supposed to be static, small movements

can have a big impact on the measured signal strength. Secondly, movements in the surrounding of the mobile device in question (e.g., persons roaming around or doors being automatically opened or closed) cause radio signals to travel different ways which in consequence causes the signal strength to fluctuate.

If the variance between different signal strength samples is high the probability that the actual closest fingerprint is selected is rather small, and a fingerprint far away might reach a higher probability. The same is true for the average calculation, because the average value might differ if the variance between different signal strength samples is high. Therefore an increase in variance also increases the chance of position errors.

The idea of the signal strength variance algorithm is to estimate the position error based on the signal strength variance of the samples that are used in the position determination phase to compute a position estimate. The algorithm can be described as follows:

1. For each access point that is part of the signal strength samples, find the largest signal strength value (in dB).
2. Based on the largest signal strength value that is specific for a certain access point, subtract this value from all signal strength samples that are available for this given access point.
3. For each access point calculate the signal strength variance for the calculated values.
4. Average the signal strength variance values calculated for each access point to get an overall variance value. This variance value can be perceived as an indicator of the expected position error.

We also tried several other ways to calculate an overall signal strength variance value. However, this did not lead to an algorithm with better results than achieved with the algorithm presented above.

3.5 Random Value

As a baseline and to make the results of our own algorithms better comparable, we also implemented a simple random error estimation algorithm. This algorithm returns, independent of the supplied data, uniformly distributed random error estimates between 0 and ten meters.

4 Experimental Setup and Methodology

In this section, we describe the experimental setup and the measurement methodology used to evaluate the proposed algorithms.

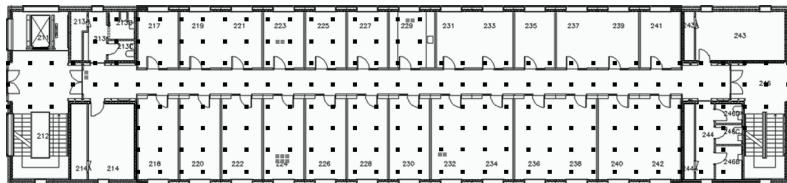
4.1 Local Test Environments

We deployed our 802.11-based positioning systems in two different environments: On the second floor of the Hopper building on the campus of the University of Aarhus

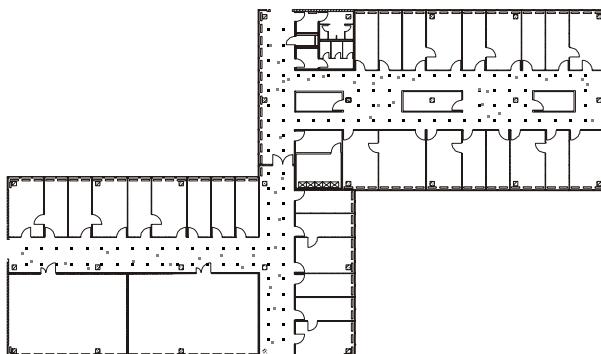
and on the second floor of the office building A5,B on the campus of the University of Mannheim.

The former one is a newly built office building consisting of many offices and a long hall (see Figure 4a). The area is covered by 23 access points from different vendors; only five of these access points can be detected in at least half of the measurements. Nine far-off access points are even only detectable in less than ten percent of all measurements. The average number of access points contained in the fingerprints for this environment is 6.9 while the average number contained in the samples used for positioning is 4.8. The 802.11-based positioning system covers an area of about 56 meters times 13 meters.

The latter environment is also situated in an office building and consists of many offices and three longer hallways (see Figure 4b). The area is covered by 25 access points in total although our data shows that most of the access points only cover parts of the operation area. In fact, only two access points cover the operation area completely. On average, 14.7 access points are contained in each fingerprint for this environment and 10.5 access points are contained in each set of measurements used for positioning during the position determination phase. The operation area is nearly 57 meters wide and 32 meters long.



(a) The second floor of the Hopper building.



(b) The second floor of the A5-B building.

Fig. 4. The two test environments. The reference spots are marked in black and the test spots are depicted in light grey.

4.2 Hardware and Software Setup

As a client, we used a Lucent Orinoco Silver PCMCIA network card supporting 802.11b. This card was plugged into an IBM Thinkpad R51 laptop computer. To collect signal strength samples, we used our own set of tools [15].

4.3 Data Collection

For both environments we applied a grid of reference spots with a grid spacing of 1.5 meters to the operation areas. For the Hopper building this resulted in 225 reference spots while the Mannheim environment comprises 130 reference spots (see the black markers in Figures 4a and 4b). For the former environment 120 signal strength measurements and for the latter one 110 signal strength measurements were subsequently collected at each reference spot.

For the position determination phase, we then selected 14 spots in the Hopper building and 46 spots in the A5-B building. At each of these spots, we collected 110 signal strength samples as well. In Figure 4a and Figure 4b, these spots are marked by light grey dots.

4.4 Metrics

Regarding the metric used to compare our error estimation algorithms, we use the *error distance difference*. For each position estimate, we compute the physical distance between the real and the estimated position (*the real error*) and then subtract the result from the value that our error estimation algorithms return as the expected error. By using the error distance difference, it is e.g., easy for us to analyze whether our algorithms tend to estimate the position error too high or rather tend to estimate it too low.

4.5 Experimental Methodology

We performed an intense evaluation to analyze the properties of our four error estimation algorithms. To keep the results comparable, we examined the four algorithms in parallel. To achieve this, we used our suite of positioning-related tools [15] and modified it to suit our special needs.

The basic experiment consists of the following steps: At first, one of the two used positioning algorithms is initialized with 25 randomly selected samples per reference spot to build up its fingerprint database. Subsequently, our four different error estimation algorithms are initialized as well. Those that make use of the training data are additionally provided with exactly the same data as the positioning algorithm itself. The same is then done with the data for the position determination phase. At first, the positioning algorithm – for each test position – is given five online samples to estimate a position. Afterwards, the same samples used for the position determination as well as also the estimated position are provided to our error estimation algorithms. Each of the algorithms then solely computes an error estimate which is recorded and stored for later reference.

The values used for the number of training and position determination samples were chosen according to [7]. To achieve statistically stable results, we repeated our basic

Table 1. Position accuracy for positioning algorithms and test environments

Positioning Algorithm	Aarhus [m]	Mannheim [m]
Bahl et al. [9]	8.10	2.90
Haeberlen et al. [10]	3.79	2.56

experiment 100 times. We ran our basic experiment by using the positioning algorithm proposed by Bahl et al. [9] and the one proposed by Haeberlen at el. [10] on data collected from both of our test environments.

As listed in Table 1, the position accuracy of the positioning algorithms by Bahl et al. [9] and Haeberlen at el. [10] is quite different when considering the data for each of our two test environments. The main reason being the number of access points, the properties of the environments and the performance of the positioning algorithms themselves.

5 Experimental Results

In this section, we present our evaluation results focusing on the results for the more accurate Haeberlen et al. positioning algorithm.

5.1 Estimation Accuracy

The proposed error estimation algorithms show noticeable differences in their single performances as well as in their reactions to different environmental conditions and to the used positioning algorithms.

Overall, for the Aarhus dataset and with the Haeberlen et al. positioning algorithm the fingerprint clustering approach delivers the best results (e.g. 1.13 meters at the median) even though the second best algorithm, the best candidates set, is quite close regarding the 95th percentile (see Figure 5a).

Looking at the Mannheim dataset, the situation has changed. Considering the 25th, 50th, and 75th percentiles, the best candidates set algorithm now takes the lead with 1.10 meters at the median (see Figure 5b). This impression is also confirmed when taking a look at the average values for the error distance difference of the four algorithms and the two environments (see Table 2). Here as well, the best candidates set algorithm performs best for the Mannheim dataset at 1.45 meters while the fingerprint clustering algorithm is ahead for the Aarhus dataset at 2.24 meters.

The reason for these differences are the different properties of the datasets. For example, as listed in Section 4.5 the average positioning error for the Mannheim dataset is noticeably lower mainly due to the on average higher number of access points available. As the fingerprint clustering algorithm tends to make very conservative error estimations, this leads – compared to the rather optimistic estimations of the best candidates set algorithm – to the just introduced results.

The opposite is true for the Aarhus dataset. Here, due to the generally larger errors of the positioning algorithms, the fingerprint clustering algorithm can deliver the best

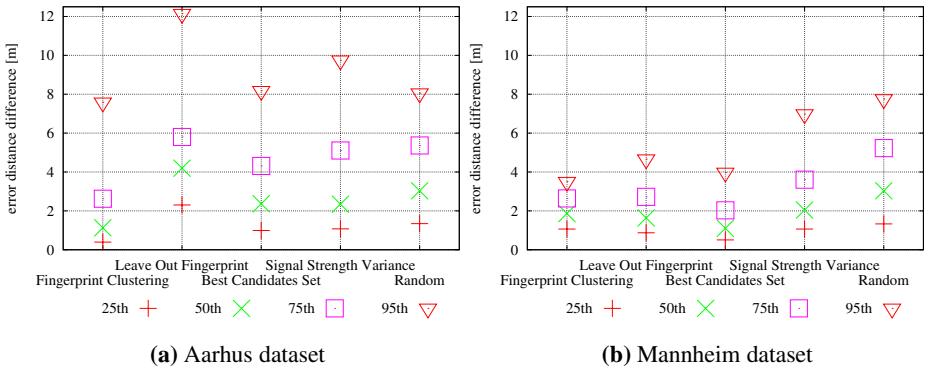


Fig. 5. Overview of the performance of the different algorithms for the Aarhus as well as for the Mannheim dataset using the positioning algorithm by Haeberlen et al.

Table 2. Average error distance difference for the Aarhus and the Mannheim dataset using the positioning algorithm by Haeberlen et al.

Algorithm	Aarhus Dataset		Mannheim Dataset	
	Avg. Error [m]	Std. Dev. [m]	Avg. Error [m]	Std. Dev. [m]
Fingerprint Clustering	2.24	2.91	1.90	1.09
Leave Out Fingerprint	4.68	3.53	1.95	1.47
Best Candidates Set	3.06	2.61	1.45	1.26
Signal Strength Variance	3.92	5.08	2.69	2.45
Random	3.58	2.84	3.43	2.39

results while the other error estimation algorithms have difficulties especially with estimating outlying errors.

For the results with the positioning algorithm by Bahl et al. the error estimation algorithms again perform best on the Mannheim data set. However, for this positioning algorithm the fingerprint clustering algorithm has the best performance on both data sets with an average accuracy of 2.03 meters for the Mannheim data and 6.56 meters for the Aarhus data. The reason that the result with the Aarhus data set is several times worse is that the accuracy of the error estimation is impacted by the low average position accuracy of 8.10 meters in this case. However, when comparing the decrease in position accuracy and error estimation accuracy it can be noted that they are on the same scale.

5.2 Combining Algorithms

In addition to evaluating the accuracy of each algorithm we have investigated if accuracy can be improved by combining the algorithms using a machine-learning algorithm. We used the Weka machine-learning toolkit [16] for our experiments. In order to feed the output of our evaluations to Weka, we implemented a small Java program that reads the evaluation results for the proposed error estimation algorithms and outputs an ARFF

Table 3. Results for different combinations of the proposed algorithms and test and training data for the positioning algorithm by Haeberlen et al.

Algorithms		Training Dataset	Test Dataset	Avg. Error [m]
All	Aarhus	Mannheim	1.37	
Two Best	Aarhus	Mannheim	1.35	
All	Mannheim	Aarhus	2.40	
Two Best	Mannheim	Aarhus	2.36	

file for the Weka tool. Before finally selecting a machine-learning method to use for combining the results, we experimented with different methods, and concluded that the *Least Median Squared Linear Regression* method was most appropriate. This method uses regression to learn weights for a linear model. The weights are learned from a training set (e.g. either the Aarhus or Mannheim data set) and then the learned model is tested using a test set (e.g. the set that was not used for training).

The results of different combinations of the proposed algorithms as well as test and training data are listed in Table 3. When combining error estimates of all the algorithms and testing with the Mannheim data an average result of 1.37 meters is achieved which is a marginal improvement over the 1.45 meters for the Best Candidate Set algorithm alone. When testing with Aarhus data the result is 2.40 meters which is a bit worse than 2.24 meters for the fingerprint clustering algorithm alone. The table also lists results when only the output of the best candidate set and fingerprint clustering algorithms are combined. These results are only a few centimeters better on average than when combining all algorithms. Given the small improvement, using a single algorithm seems like a better option than combining several algorithms, especially when considering the additional system complexity and increased computational requirements for running several algorithms in parallel and then later combining their output.

5.3 Over Estimation versus under Estimation

Some applications might have preferences with respect to overestimation or underestimation of position errors. For instance, an application that has to alert a user in case the estimated error is above a threshold might prefer overestimation in order to avoid to not alert the user (false negatives). On the other hand, an application that sends the user information about shops in his proximity might prefer an underestimation of the error to avoid annoying the user with too many messages. In Figure 6 the error distance difference distributions are depicted for the two most promising algorithms. The positioning system proposed by Haerberlen et al. is used and data for both test environments is displayed. In these distributions underestimations result in negative values while overestimations lead to positive values. Our results show that the best candidates set algorithm has a tendency to produce a higher amount of large overestimations. Also, when comparing the results for the Aarhus and Mannheim datasets, the distributions reveal that both algorithms produce a higher amount of large under- and over-estimates for the less accurate Aarhus dataset.

Our proposed algorithms offer the possibility to adjust the error estimation performance to favour either under- or overestimation with algorithm-specific parameters.

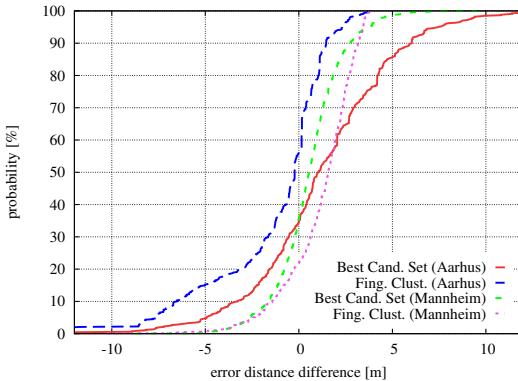


Fig. 6. Distributions of the difference between estimated and real error

This comes at the expense of reliability, though. For example, for the best candidates set algorithm, the size of the set can be varied. When considering more candidates in our evaluations, the number of errors that are underestimated decreases. But at the same time, the amount of overestimation increases because more, possibly physically far distinct positions are taken into account. The same is true for the fingerprint clustering algorithm. When adjusting the similarity threshold for the merging of clusters, the size of the resulting regions increases. This also increases the probability of the estimated and real position being located in the same region. But at the same time it also increases the average error distance difference in our evaluations. Therefore, here again, we have a tradeoff between accuracy and reliability.

6 Discussion

This section discusses the space and time complexity of the proposed algorithms, the influence of the number of position determination samples and the number of access points on the error estimation, and the GPS approach for the dilution of precision.

6.1 Space and Time Complexity

Considering the space and time complexity of our algorithms, we have to distinguish between those operating only on the training data and those also taking the measurements during the position determination phase into account.

For the former ones, computational complexity is rather not critical as the algorithms need to be run on the training data only once. This can be done on powerful systems in advance of the deployment. The opposite is true for the space complexity. As the result of the algorithms, depending on the setup of the positioning system, might be stored and processed on the mobile device, here the amount of data is important.

For the latter ones, both space and time complexity are important. For each single position determination, a computation based on the current set of measurements as well as the training data takes place. Considering that this might as well take place on a

Table 4. Comparison of the space and time complexity for the proposed algorithms, where c is the number of initial cells, n is the number of fingerprints, p is the time complexity of the positioning algorithm, a is the number of access points in the online samples, b is the size of the best candidate set and h is the number of stored samples

	Fingerprint Clustering	Leave-Out Fingerprint	Best Candidate Set	Signal Strength Variance
Time complexity	$O(c)$	$O(n * p)$	$O(b)$	$O(a * h)$
Space complexity	$O(c)$	$O(n)$	$O(b)$	$O(a * h)$

mobile device, the algorithms have to be economical regarding CPU time and memory consumption. However, it turns out that both the fingerprint clustering and the best candidate set algorithms are light-weight in terms of computations and space use, please refer to Table 4 for the details.

6.2 Number of Position Determination Samples and Number of Access Points

From experimental results King et al. [7] conclude that the 802.11 position error is mainly determined by the number of access points and the number of signal strength samples used in the position determination phase. In this work, we do not present any algorithm that takes this fact explicitly into account. However, the previously presented algorithms indirectly adhere to this fact. For instance, the fingerprint clustering algorithm (see Section 3.1) when comparing similarity between clusters iterates over the set of access points and by such explicitly handles the number of access points. The number of position determination samples is explicitly covered by the signal strength variance algorithm (see Section 3.4) as the variance value stabilizes if the number of signal strength samples increases. However, it would be an interesting path of future work to analyze if the proposed algorithms could be extended to more explicitly use such results.

6.3 Dilution of Precision

In addition to the proposed algorithms, we have also implemented an algorithm that computes a GPS-like DOP value based on the geometry of access points. The algorithm uses the equations for DOP values described in Borre et al. [17]. However, a DOP value is not an error estimate in meters but a factor that has to be combined with an error estimate. Therefore we have evaluated if the accuracy of the proposed algorithms can be improved by combining their output with a DOP value by multiplication. However, for all of the algorithms the combination with a DOP value strongly reduces their accuracy. This suggests that GPS-like DOP values do not seem to be a promising error estimation approach for indoor 802.11-based positioning. The main reason for that is that the complex indoor propagation environment makes the geometry of the access points less important for the positioning accuracy than it is in the case of outdoor satellite-based positioning. This result is consistent with the analytical results for the impact of the geometry of access points on the position accuracy presented by Wallbaum et al. [11].

7 Conclusion and Further Work

In this paper, we have introduced four novel algorithms to estimate the position error of 802.11-based positioning systems. Our algorithms exploit several different features to fulfill their task. We have proposed error estimation algorithms that only use training data as well as algorithms that take only the data available during the position determination phase into account.

All our algorithms deliver good results for estimating the position error both for different environments and for the different positioning algorithms. The fingerprint clustering algorithm and the best candidates set algorithm though perform especially well. With either of these two algorithms it is possible to estimate the position error of an 802.11-based positioning system to a high degree of accuracy and with sufficient reliability. This is in contrast to the geometry-based algorithms used for GPS which fail to work for 802.11-based positioning systems.

We have evaluated if using a machine learning technique to combine several of the proposed algorithms can improve the precision of the error estimates. In our case though, the gain is very small or the results even worsen. We therefore suggest not to use any combination of the different algorithms but rather use one algorithm alone. Looking at the targeted use of the algorithms on mobile devices, this suggestion makes even more sense as the use of multiple algorithms – even though each single one economical in terms of processing and storage demands – would waste precious resources.

This work opens up several paths of future work. Firstly, to analyze if the proposed algorithms could be extended to more explicitly take information such as the number of position determination samples into account. Secondly, to further explore the relationship between algorithms for error estimation and analytical error models. Thirdly, the evaluated positioning algorithms were for reasons of analyzability not extended with advanced methods such as tracking. Therefore, a further path of future research will be to analyze how to optimize our algorithms when applying them to a positioning algorithm that uses e.g., tracking.

References

1. Kaplan, E., Hegarty, C. (eds.): *Understanding GPS: Principles and Applications*, 2nd edn. Artech House Incorporated (December 2005)
2. LaMarca, A., Chawathe, Y., Consolvo, S., Hightower, J., Smith, I., Scott, J., Sohn, T., Howard, J., Hughes, J., Potter, F., Tabert, J., Powledge, P., Borriello, G., Schilit, B.: Place Lab: Device Positioning Using Radio Beacons in the Wild. In: Gellersen, H.-W., Want, R., Schmidt, A. (eds.) *PERVASIVE 2005*. LNCS, vol. 3468, pp. 116–133. Springer, Heidelberg (2005)
3. Cheng, Y.C., Chawathe, Y., LaMarca, A., Krumm, J.: Accuracy Characterization for Metropolitan-scale Wi-Fi Localization. In: Proceedings of the Third ACM International Conference on Mobile Systems, Applications, and Services (2005)
4. Krumm, J., Cermak, G., Horvitz, E.: RightSPOT: A Novel Sense of Location for a Smart Personal Object. In: Dey, A.K., Schmidt, A., McCarthy, J.F. (eds.) *UbiComp 2003*. LNCS, vol. 2864, pp. 36–43. Springer, Heidelberg (2003)
5. Institute for Electrical and Electronics Engineers, Inc.: *ANSI/IEEE Standard 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications* (1999), <http://standards.ieee.org/getieee802/>

6. Bychkovsky, V., Hull, B., Miu, A., Balakrishnan, H., Madden, S.: A Measurement Study of Vehicular Internet Access Using In Situ Wi-Fi Networks. In: Proceedings of the 12th International Conference on Mobile Computing and Networking (2006)
7. King, T., Haenselmann, T., Effelsberg, W.: Deployment, Calibration, and Measurement Factors for Position Errors in 802.11-based Indoor Positioning Systems. In: Hightower, J., Schiele, B., Strang, T. (eds.) LoCA 2007. LNCS, vol. 4718, pp. 17–34. Springer, Heidelberg (2007)
8. Dearman, D., Varshavsky, A., de Lara, E., Truong, K.N.: An exploration of location error estimation. In: Krumm, J., Abowd, G.D., Seneviratne, A., Strang, T. (eds.) UbiComp 2007. LNCS, vol. 4717, pp. 181–198. Springer, Heidelberg (2007)
9. Bahl, P., Padmanabhan, V.N.: RADAR: An In-Building RF-based User Location and Tracking System. In: Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (2000)
10. Haeberlen, A., Flannery, E., Ladd, A.M., Rudys, A., Wallach, D.S., Kavraki, L.E.: Practical Robust Localization over Large-Scale 802.11 Wireless Networks. In: Proceedings of the Tenth ACM International Conference on Mobile Computing and Networking (2004)
11. Wallbaum, M.: A priori error estimates for wireless local area network positioning systems. *Pervasive Mobile Computing* 3, 560–580 (2007)
12. Kaemarungsi, K., Krishnamurthy, P.: Modeling of indoor positioning systems based on location fingerprinting. In: Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (2004)
13. Krishnakumar, A.S., Krishnan, P.: On the accuracy of signal strength-based estimation techniques. In: Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (2005)
14. Youssef, M., Agrawala, A.: On the optimality of wlan location determination systems. In: Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation Conference (2004)
15. King, T., Butter, T., Lemelson, H., Haenselmann, T., Effelsberg, W.: Loc{lib,trace,eva,ana}: Research Tools for 802.11-based Positioning Systems. In: Proceedings of the ACM WiN-TECH (2007)
16. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, San Francisco (2005)
17. Borre, K., Akos, D., Bertelsen, N., Rinder, P., Jensen, S.: A Software-Defined GPS and Galileo Receiver. Birkhäuser (2007)

Activity Recognition from Sparsely Labeled Data Using Multi-Instance Learning

Maja Stikic¹ and Bernt Schiele²

¹ Fraunhofer IGD, Germany

² TU Darmstadt, Germany

{stikic,schiele}@cs.tu-darmstadt.de

Abstract. Activity recognition has attracted increasing attention in recent years due to its potential to enable a number of compelling context-aware applications. As most approaches rely on supervised learning methods, obtaining substantial amounts of labeled data is often an important bottle-neck for these approaches. In this paper, we present and explore a novel method for activity recognition from sparsely labeled data. The method is based on multi-instance learning allowing to significantly reduce the required level of supervision. In particular we propose several novel extensions of multi-instance learning to support different annotation strategies. The validity of the approach is demonstrated on two public datasets for three different labeling scenarios.

1 Introduction

A growing interest in context-sensitive applications has resulted in a wealth of research on activity recognition. Scalable and unobtrusive recognition of human activities offers a number of opportunities for new application domains in industrial, educational, and assisted-living environments. Most existing approaches to activity recognition are based on supervised machine learning techniques requiring significant amounts of labeled training data. However, annotating activity data still remains one of the main challenges in real-world settings.

There exists a wide range of annotation techniques used for capturing ground-truth in activity recognition studies. An important difference between methods is whether they depend on online annotations during execution of the activities (e.g. experience sampling [1], time diary [2], and direct observations [3]) or rely on offline annotations through video and audio recordings [4], subject's self-recall [5], or indirect observations of sensor data [6]. Figure 1 illustrates a typical trade-off between accuracy of an annotation method and the time required for annotation. Methods that provide accurate annotations such as direct observations or video and audio recordings are labor-intensive, scale poorly to large numbers of users and activities, and are often not acceptable due to privacy concerns. In contrast, experience sampling and time diary require user involvement which can lead to recall errors and inaccurate annotations of short term activities, lack of temporal precision, and frequent interruptions that might change the activity itself and disrupt the user.

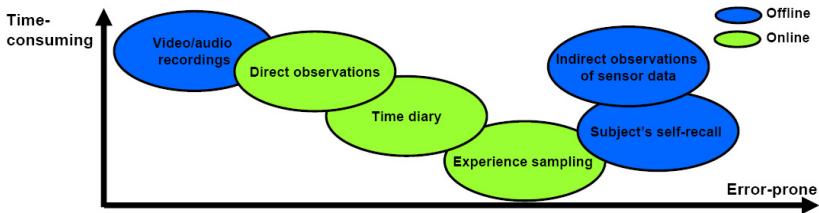


Fig. 1. The most commonly used annotation techniques for activity recognition

In this work, we address the labeling challenge by introducing a novel activity recognition approach that requires only coarse-grained annotations. The method utilizes multi-instance learning for activity recognition which enables learning of activity models from sparsely labeled data by propagating knowledge from a few provided labels to the neighboring data points. The two major benefits of the proposed approach are: 1) It enables to decrease the level of annoying interruptions by probing a user only occasionally about performed activities and 2) A user does not have to provide accurate start and ending times of activities.

The main contributions of the paper are threefold. First, we reduce the level of supervision in activity recognition by acquiring knowledge from incomplete and ambiguous labels using multi-instance learning. Second, we extend multi-instance learning to support different annotation strategies by initializing the algorithm with a few correctly labeled data samples and adjusting it to the multi-class setting. Third, we explore and analyze the applicability of the approach in three different experience sampling scenarios: 1) A user is asked only about his current activity, 2) A user provides information about all activities he performed during a given time interval but without exact time when the activities occurred, and 3) A user provides information only about the activity he was performing most of the time during a given time interval. In all three cases, we achieve high recognition performance with a small amount of experience sampling probes. That way, the level of experience sampling annoyance is reduced. We also report results on two different public datasets in order to avoid bias to a particular dataset only.

The rest of the paper is structured as follows. In Section 2 we put our work into context by discussing related work. Section 3 describes the multi-instance learning framework and the applied algorithm. Section 4 motivates and presents three different annotation strategies evaluated in this paper. In Section 5 and Section 6 we introduce the datasets used in the experiments and present the goals of our evaluation procedure, respectively. The following Sections 7, 8 and 9 report on our modified extensions of multi-instance learning for activity recognition and experimental results of all three labeling scenarios evaluated in this paper. Section 10 discusses the results of comparative evaluation procedure. In Section 11 we conclude with a summary and outlook on future work.

2 Related Work

Research on human activity recognition has achieved impressive results both on low-level activities (e.g. sitting, standing, lying, and walking [3,7] and high-level activities (e.g. Activities of Daily Living [6,8,9,10], daily routines [2], maintenance tasks [11], and office activities [12]). The types of sensors proposed and used for activity recognition range from motion sensors such as accelerometers [8,13] and infra-red sensors [14] over RFID tag readers [9], state change sensors [6] and cameras [10,15] to combinations of different sensor modalities [11,16]. Most of the work in the area of activity recognition is based on supervised machine learning algorithms such as *Naive Bayes* [6], *Decision Trees* [4,8], *Hidden Markov Models* [7,17], *Nearest Neighbor* [3,5], *Support Vector Machines* [13,17], and different *Boosting* techniques [13]. Moving beyond fully supervised settings, researchers have started studying the feasibility of other machine learning techniques to lower the annotation burden. Three general directions explored so far are:

1) Unsupervised learning. Unsupervised techniques enable discovery of structure in activity data without any labeled training data needed. Previous work in motif discovery [18] and topic models [2] focused on discovery and modeling of short-term motion primitives, and daily routines as a probabilistic combination of low-level activity patterns, respectively. However, one still must provide at least a few labels to perform classification. There are also attempts to decrease labeling efforts by manually defining common sense models of daily activities [16] or mining these models from the web [9].

2) Active learning. The goal of active learning is to focus labeling efforts on the most profitable instances. Several active sampling functions have been proposed for activity recognition based on a multi-sensor approach [19] and for Hidden Markov Models [20]. Moreover, in an office-centered setting, different experience sampling strategies based on active learning [21] have been evaluated [22] for predicting user's interruptibility.

3) Semi-supervised learning. In a semi-supervised setting [23], typically there is only a small set of labeled training data available in addition to a substantial amount of unlabeled training data. In the context of activity recognition, there have been attempts to learn from both labeled and unlabeled data by using iterative algorithms such as self-training, co-training [19], and En-Co-Training [24]. For graphical models it has been proposed to use virtual evidence [25] mechanism for depicting evidence in a dynamic Bayesian network (DBN) during joint reasoning about low-level activities and spatial context, as well as semi-supervised virtual evidence boosting [26] for training conditional random fields (CRF) on data collected from wearable sensors. Furthermore, in [27] expectation-maximization (EM) algorithm was used for semi-supervised learning of a partially observable Markov decision process (POMDP) for the task of handwashing.

The goal of this paper is to explore another direction for decreasing the level of supervision in activity recognition. In the machine learning community, multi-instance learning [28] has emerged as a promising approach for problems with

incomplete knowledge about labels of training samples. As multi-instance learning is robust to labeling noise and can achieve competitive classification accuracy using only a small amount of training data, we argue that it is excellently suited for the activity recognition problem where labels are hard to obtain and training data is often ambiguously labeled. In the next section, we describe the multi-instance learning framework and the applied algorithm.

3 Multi-Instance Learning

Standard supervised learning requires labeled training data, i.e. each training instance is associated with its corresponding class label. Multi-instance learning relies on a significantly weaker assumption about the labeling information. Here, the labels are not assigned to the individual training instances, but rather to sets of instances, namely *bags-of-instances*. The bags are labeled based on the following two rules: 1) The bag is labeled *positive* if *at least one* instance in the bag is positive, and 2) The bag is labeled *negative* if *all* instances in the bag are negative. Even though the labels of individual patterns are not provided, multi-instance learning successfully copes with the ambiguity of not knowing which of the instances in positive bags are actual positive instances and which ones are not. It aims to find the optimal labeling of the instances in positive bags that consequently lead to the optimal classifier.

Multi-Instance SVM (miSVM). It has been shown in the past [13,17] that discriminative classifiers such as Support Vector Machines (SVM) achieve high recognition performance on activity data. Thus, in our experiments we adopt a modified version of the support vector framework that enables activity recognition from sparse labels - called multi-instance SVM (miSVM) [29].

Standard supervised SVM [30] is a linear binary classifier. It constructs an optimal separating hyperplane in the feature space of labeled training data that minimizes the expected generalization error by maximizing the margin, i.e. the distance from the hyperplane to the nearest data points - so-called support vectors. In the case of non-separable classes, soft-margin is maximized by introducing slack variables that allow misclassification of training data. Additionally, a misclassification penalty parameter controls the trade-off between training error and margin. A nonlinear classifier is obtained by using a kernel transformation, i.e. the data are implicitly mapped to a high dimensional feature space where it is more likely that the two classes are linearly separable.

In the multi-instance setting the bag labels provide only partial information about the labels of their comprising instances. A negative bag label imposes negative label to each instance in the bag. In contrast, unique labels of the instances in positive bags are not known. Intuitively, in order to extend the support vector setting to the multi-instance case, based on the above bag labeling rules, one would construct the maximum-margin hyperplane in the following way: There should be at least one instance from every positive bag in the positive halfspace, while all instances belonging to negative bags are in the negative

halfspace. Practically, miSVM treats the labels of instances in positive bags as unobserved indicator variables. The goal is to maximize the regular hyperplane margin (or soft-margin), jointly over hidden label variables (i.e. possible label assignments) as well as over linear (or kernelized) discriminant functions (i.e. possible hyperplanes). This leads to a hard mixed integer programming problem and therefore, an iterative local optimization heuristic is applied in the following manner. Initially, all instances in positive bags are assumed to be positive. Given this initial labeling assumption, the regular SVM margin optimization problem is solved. Based on the found maximum-margin hyperplane, instances in positive bags are classified. Furthermore, since positive bags contain at least one positive instance, additional enforcement is applied if a current classifier discriminant function classifies all instances in a positive bag as negative. The “least negative instance” is assigned the positive label, i.e. the instance with the highest value of discriminant function. The classifier is then re-trained based on the refined labels and the procedure is iteratively repeated until it converges to a local minimum of the SVM objective function, i.e. until the labeling of instances in positive bags does not change any more. The final classifier can then be used in a standard way.

4 Bag-of-Activities Generators

In this section, we show how multi-instance learning can be applied to activity recognition by introducing *bags-of-activities*. This way, the activity labels do not have to be provided for each data point but rather on a very coarse level: sensor data is grouped into bags and the labels are provided for the bags. We explore the applicability of multi-instance learning for three different experience sampling strategies. It is clearly desirable to decrease the number of experience sampling probes to a minimum. Thus, we aim to prompt a user to provide information about the performed activities as rarely as possible by employing longer experience sampling time intervals. Additionally, the provided level of annotation details is significantly weaker because a user does not have to recall exact start and ending times of activities. The bags comprise activity data between the two successive experience sampling prompts. We evaluate the following three bag-of-activities label generators that partially fulfill multi-instance bag labeling rules defined in Section 3. In order to make multi-instance learning applicable to these three settings, we will discuss our proposed extensions and modifications in Sections 7, 8, and 9.

1) Single-labeled bags. This bag-of-activities label generator is based on user’s periodical information about the activity he is performing at a moment. Thus, in this case each bag is assigned a single label, i.e. the user’s current activity. For longer experience sampling time intervals, bags typically consist of several activities. This introduces noise in bags which can be handled by multi-instance learning. It is highly likely that a user is going to continue the current activity for a certain time after a prompt, i.e. the surrounding instances around the probe have a high probability to belong to the activity provided by a user in the latest

prompt. Therefore, we can reduce the level of noise in bags by centering the bags around successive experience sampling prompts and by shortening the bag. This case can be handled directly with a slightly modified multi-instance algorithm. Details will be given in Section 7.

2) Multi-labeled bags. In this case, a user provides information about all activities he performed in a given time interval, but not the exact times when the activities occurred. Thus, each bag is assigned multiple labels, comprising all activities the bag is composed of. Unlike the single-labeled bags, where very short events are likely to be missed, the multi-labeled bags allow to recover all activities that took place during the observed time period. The necessary extensions of multi-instance learning for this case will be presented in Section 8.

3) Majority voting bags. The third bag-of-activities label generator is based on the assumption that it is easier for a user to recall activities lasting for extended time periods than relatively short activities. Hence, in this case, a user is periodically prompted to provide information on which activity he was performing most of the time during a time interval. In other words, the label of a bag is induced by majority voting of individual bag instances. Generating bag labels in this manner creates relatively clean bags, i.e. a large proportion of instances in a bag matches the bag label. The shortcoming of this bag-of-activity label generator is that short activities are often missed, especially when aiming for longer experience sampling time intervals. Further details about the algorithms used for this setting will be described in Section 9.

5 Datasets

In the following, we give a short overview of the datasets used for evaluation of our approach. As we aim for the standardized evaluation procedure in the field of activity recognition, we use two public datasets, namely PLCouple1 [4] and TU Darmstadt [2]. Both datasets are recorded over longer periods of time in real-world settings comprising typical non-scripted daily activities of a single subject. What makes these datasets especially challenging is the fact that the amount of data for activities varies a lot for different activities reflecting the natural distribution and duration of activities in real life. The datasets include fine-grained annotations of activities which make them suitable for systematic analysis of our approach, i.e. we can compare performance of the algorithm to the standard supervised approaches that use the provided detailed annotations.

PLCouple1 dataset [4] contains 10 weeks of recordings from 3 3D accelerometers worn on the dominant wrist and hip and the non-dominant thigh. In our experiments we use 68 hours of annotated data collected on 9 separate days. We focus on the same set of activities as the authors of the original paper [4]: *using computer, actively watching tv or movies, eating, using phone, meal preparation, grooming, hygiene, and dishwashing*. All other activities in the dataset are considered as an unknown class. Since data is recorded at relatively high frequency

of 20Hz, we extract the following features: *mean*, *variance*, *energy*, *spectral entropy*, *area under curve*, *pairwise correlation between the three axes*, and the first ten *FFT coefficients*, which overall sums up to 96-dimensional feature vectors. It has been shown in [17] that for high-level activities, it is sufficient to extract features over longer time periods. Thus, we compute the features over a sliding window of 30 seconds shifted in increments of 15 seconds.

The TU Darmstadt dataset [2] consists of data from 2 3D accelerometers worn on the dominant (right) wrist and in the right hip pocket recorded during a period of 16 days. Data from 7 days are annotated, resulting in 84 hours of usable annotated data. Although the dataset contains 34 distinct activities, in our experiments we could use only a subset of 20 different activities for the following reason: In our experiments we use leave-one-day-out cross-validation. Since many activities appeared only on one day during the recorded time period, our classifier would have no chance of recognizing those activities. Therefore, we target the following activities along with the unlabeled class: *sitting/desk activities*, *lying while reading/using computer*, *having dinner*, *walking freely*, *driving car*, *having lunch*, *discussing at whiteboard*, *driving bike*, *standing/talking on phone*, *walking while carrying something*, *walking*, *picking up cafeteria food*, *sitting/having a coffee*, *queuing in line*, *personal hygiene*, *using the toilet*, *washing dishes*, *brushing teeth*, *standing/using the toilet*, and *washing hands*. Since in the original paper [2] the use of frequency features did not improve recognition performance due to the relatively coarse resolution of data (2.5Hz), for this dataset we use, as in [2] only *mean*, *variance*, and *time-of-day* as features over the same sliding window length used for PLCouple1 dataset producing 13-dimensional feature vectors.

6 Evaluation

In this section, we present the goals of our experiments and describe our evaluation procedure. The experiments are built in such a way to systematically compare the three experience sampling strategies described in Section 4. We evaluate performance of multi-instance learning for different experience sampling time intervals, starting from 10 minutes to 180 minutes. Bags consisting of up to 180 minutes of activity data might include significant amounts of ambiguously labeled data. Besides, it might be hard for a user to recall performed activities during such long stretches of time. Thus, we additionally evaluate performance of the algorithms trained on a shorter 10 minutes time interval of the entire bag centered in the middle of the bag. That way, the amount of labeling noise in bags is reduced, but overall amount of training data is also decreased, which makes the recognition more challenging.

We use two common baselines for all three labeling scenarios: 1) the results obtained with standard supervised SVMs using all available training data as individually labeled instances, and 2) the results obtained with supervised SVMs using all available training data labeled based on bags' labels, i.e. all instances in negative bags are labeled as negative and all instances in positive bags are labeled as positive. That way, one can obtain a better understanding of the algorithm

behavior and insights into the benefits and limitations of the approach in the three evaluated settings.

In order to extend binary SVMs to our multi-class activity recognition setting, we apply the typical “one vs. rest” approach by training N SVMs, each separating a single class from all remaining classes. In the experiments, we use the Gaussian Radial Basis Function (RBF) kernel $K(\mathbf{x}, \mathbf{y}) = \exp\{-\gamma \|\mathbf{x} - \mathbf{y}\|^2\}$. The parameter γ and misclassification penalty parameter C are determined by coarse grid search over the parameter space. For the fully labeled dataset, we obtained the following parameters: $\gamma = 0.1$, $C = 10$ for PLCouple1 dataset, and $\gamma = 1$, $C = 10$ for TU Darmstadt dataset. These parameters are consistently used in all following experiments.

All results reported in the following sections are cross-validated in a leave-one-day-out fashion by repeatedly training the algorithms on data from all but one day, and testing the trained classifier on the left-out-day’s data.

Interestingly, regular SVMs applied to completely annotated activity data obtain slightly better recognition performance comparing to the results reported in the original publications where PLCouple1 and TU Darmstadt datasets were introduced, based on Decision Trees and Naive Bayes classifiers, respectively. With SVMs, we achieve an accuracy of 71.3% on PLCouple1 dataset and 76.3% on the TU Darmstadt dataset. The previously reported results [4] for the PLCouple1 dataset are 72.2% average ROC area (corresponding to an accuracy of 53.6% reported in [19]). In [2] an accuracy of 72.7% was achieved on the TU Darmstadt dataset.

In the following sections we introduce our extensions of the multi-instance learning framework and report on the experiments of their applicability to activity recognition.

7 Single-Labeled Bags

In this section, we report on the experiments conducted on the single-labeled bags from Section 4. In this case, the label of the bag represents only the activity a user was performing at the moment of the experience sampling prompt. Thus, unlike the standard multi-instance setting, negative single-labeled bags might comprise not only negative instances but also a smaller amount of positive instances. In order to overcome this issue, we make use of the fact that for single-labeled bags, in principle we know to which instance in a bag the provided bag label belongs to. By carrying out the experiments with only that limited amount of accurately provided labeled activity data, standard supervised SVMs already perform surprisingly well. Therefore, we initialize the iterative multi-instance learning procedure of miSVM presented in Section 3 by the SVM model learned in a standard supervised way on that restricted set of provided accurate labeled data. Furthermore, we keep the labels of this data fixed during multi-instance learning.

As already mentioned in Section 5, the activities in the datasets used in our experiments vary greatly in duration and their frequency of occurrence in real

life. In order to compensate for highly unbalanced classes (i.e., activities) in these two datasets, we apply activity priors that could be easily acquired, for example from the existing time-use study data [31]. We utilize the activity priors during both stages of the multi-instance learning miSVM algorithm. For training purposes, we employ different cost factors for misclassification of positive and negative instances based on their prior probabilities. Also, during the classification phase of multi-instance learning, miSVM assigns a constrained proportion of positive and negative labels to the instances in positive bags reflecting the activity priors. In other words, we allow the assignment of positive labels only to the limited amount of instances for which SVM provides the highest prediction scores. The rest of the instances in positive bags are classified as negative. That way, we maintain the underlying distribution of activities in our dataset and avoid the risk of large classes dominating the multi-instance learning procedure.

Results. To illustrate the improvement we can achieve by assigning to the instances in the bags labels obtained through multi-instance learning, in Figure 2 we show labeling accuracy of instances in the bags at the beginning and end of

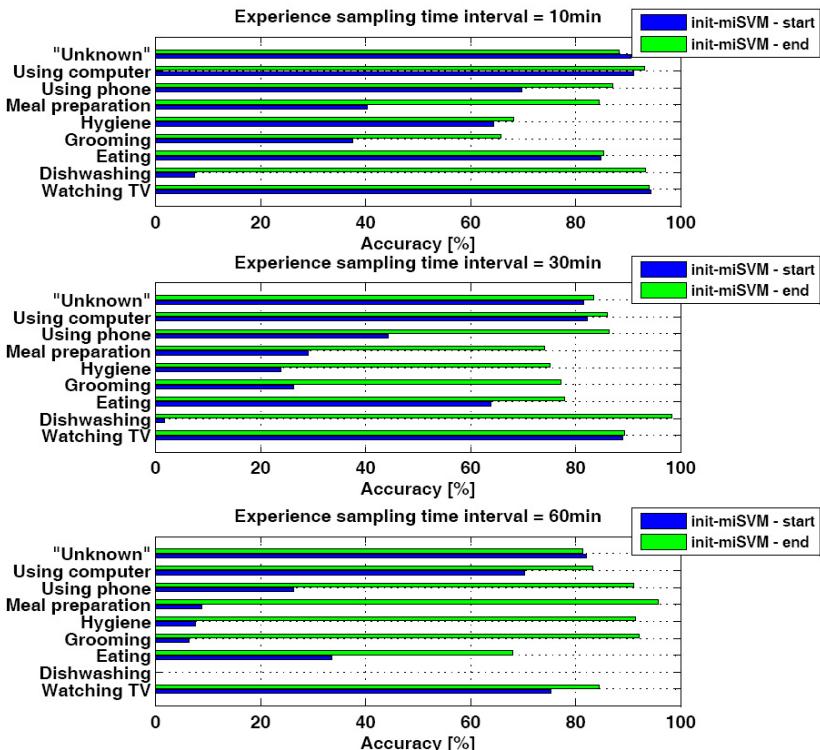


Fig. 2. PL Couple1 dataset: Labeling accuracy at the beginning (i.e. first iteration) and end (i.e. last iteration) of multi-instance learning iterative training procedure

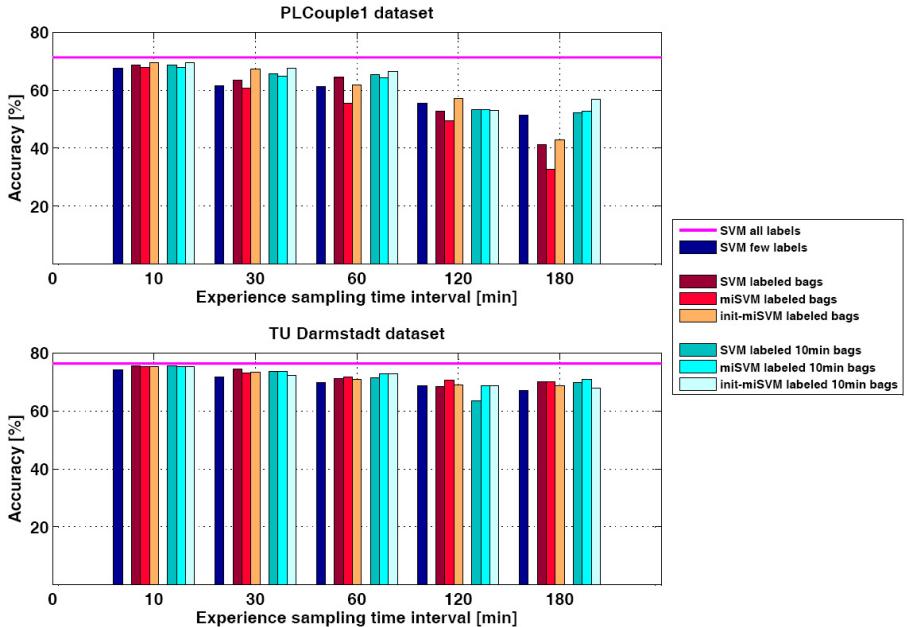


Fig. 3. Single-labeled bags: Comparative performance of supervised baselines (*SVM all labels*, *SVM few labels*, *SVM labeled bags*, *SVM labeled 10min bags*) and multi-instance learning approaches (*miSVM labeled bags*, *init-miSVM labeled bags*, *miSVM labeled 10min bags*, *init-miSVM labeled 10min bags*) for different experience sampling time intervals in case of PLCouple1 (top) and TU Darmstadt datasets (bottom)

our extended multi-instance learning iterative training algorithm for the PLCouple1 dataset for different experience sampling time intervals. From the plots one can observe a consistent improvement for almost all classes of activities. The improvement is impressive for activities *dishwashing*, *grooming*, *hygiene*, and *meal preparation*. When starting with just a few provided labels for these activities, multi-instance learning still enables to find the right labeling of these activities. It can occasionally happen that there are no provided labels for some activities, when using larger experience sampling intervals (e.g. activity *dishwashing* when using a 60min experience sampling time interval).

Figure 3 shows the achieved classification accuracy on PLCouple1 and the TU Darmstadt datasets for the standard multi-instance learning miSVM algorithm for both types of single-labeled bags: larger bags (*miSVM labeled bags*) and 10minutes bags (*miSVM labeled 10min bags*). It also shows the performance of our extended version of the miSVM algorithm, initialized with a few correctly labeled data points, for single-labeled bags (i.e. *init-miSVM labeled bags* and *init-miSVM labeled 10min bags*). The performance of these two algorithms is compared to the following supervised baselines: 1) *SVM all labels* - standard SVM algorithm when using all training data as labeled, 2) *SVM few labels* - SVM algorithm when using only a few correctly labeled data obtained through

single-labeled bags annotation strategy, 3) *SVM labeled bags* and *SVM labeled 10min bags* - SVM algorithm when using single-labeled bag labels as labels of all its instances for both types of bags. All algorithms are evaluated for different experience sampling time intervals.

A few trends in Figure 3 stand out. First, using experience sampling time intervals up to 60 minutes does not significantly decrease the performance of the algorithms. Thus, the level of interruptions can be significantly reduced by decreasing the number of experience sampling prompts to only one prompt per hour. Second, 10minutes bags typically achieve competitive (in case of the TU Darmstadt dataset) or even better recognition rates (in case of the PL Couple1 dataset) than the larger bags. These are two very important findings of our experiments that enable an easier way of annotating data for activity recognition.

Furthermore, our modified version of the multi-instance learning algorithm (i.e. *init-miSVM*) improves the performance of the regular *miSVM* algorithm up to 10% for PL Couple1 dataset when using the larger bags. The improvement on 10minutes bags is not that significant due to the smaller amount of noise in these bags. For the TU Darmstadt dataset, performance is occasionally even decreased due to the poor performance of *SVM few labels* algorithm, which is the initialization part of our *init-miSVM* algorithm. Even though *init-miSVM* typically outperforms *SVM labeled bags* and *SVM labeled 10min bags*, these two supervised approaches still preserve surprisingly high performance, despite of the bag labeling noise.

Lastly, performance of the standard *miSVM* algorithm is often decreased comparing to *SVM labeled bags* baselines. This is most likely due to the fact that the single-labeled bags do not meet the standard multi-instance requirement of clean negative bags. We also conducted an experiment where we discarded the noisy parts of the negative bags, and the performance of the standard *miSVM* algorithm was significantly improved (up to 20% for the PL Couple1 dataset and up to 10% for the TU Darmstadt dataset). In order to further explore this direction in a more realistic setting, we conducted an extensive set of experiments on multi-labeled bags that we report in the next section.

8 Multi-Labeled Bags

The multi-labeled bags introduced in Section 4 provide complete information about all activities a bag consists of, but without precise assignments of labels to the individual instances. Thus, in this case the multi-instance learning requirement of clean negative bags is fulfilled. As the applied miSVM algorithm utilizes a binary SVM classifier, we are able to deal with multi-labels in a principled way by using the same bag multiple times as a positive bag (i.e. when constructing a binary classifier for each activity that appears in that particular bag).

Since in this case we have information about all activities appearing in the bag, we additionally extend multi-instance learning to the multi-class setting by adjusting the classification phase of miSVM iterative learning algorithm in the following manner. During classification of instances in a bag, only the classifiers

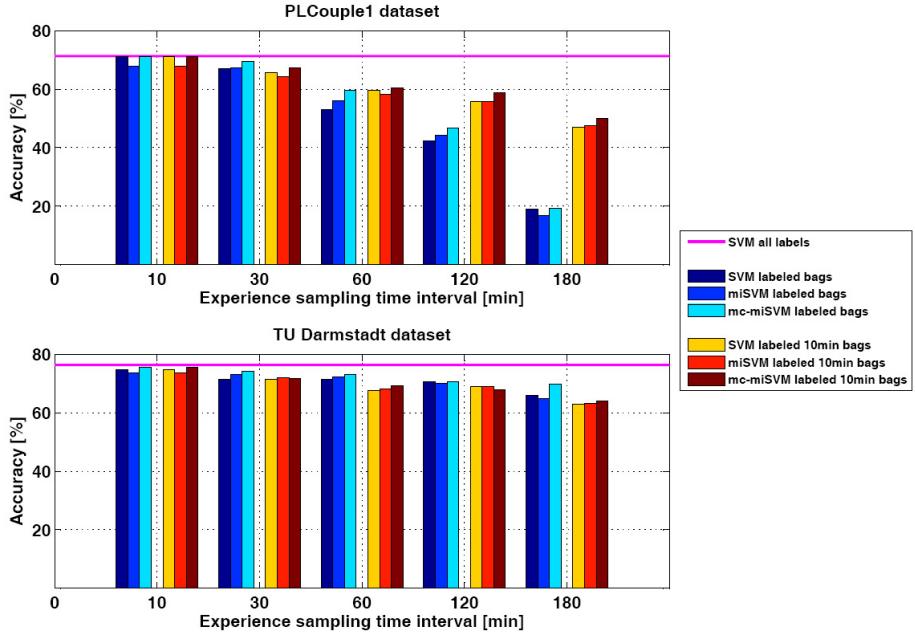


Fig. 4. Multi-labeled bags: Comparative performance of supervised baselines (*SVM all labels*, *SVM labeled bags*, *SVM labeled 10min bags*) and multi-instance learning approaches (*miSVM labeled bags*, *mc-miSVM labeled bags*, *miSVM labeled 10min bags*, *mc-miSVM labeled 10min bags*) for different experience sampling time intervals in case of PL Couple1 (top) and TU Darmstadt datasets (bottom)

of the activities that are present in that particular bag are allowed to compete for instances in the bag. The classifier that provides the highest prediction score for an instance assigns its label to that instance. That way, we disable unnecessary misclassification of instances by other activity classifiers not included in the bag.

Results. Figure 4 shows the results for multi-labeled bags on the PL Couple1 and TU Darmstadt datasets. The plot compares the performance of the following multi-instance learning algorithms: 1) Standard miSVM algorithm on larger multi-labeled bags (*miSVM labeled bags*) and 10minutes multi-labeled bags (*miSVM labeled 10min bags*), 2) Our extended multi-class version of standard miSVM algorithm both for larger (*mc-miSVM labeled bags*) and 10minutes bags (*mc-miSVM labeled 10min bags*) to the supervised baselines: 1) *SVM all labels* - standard SVM algorithm on fully labeled data and 2) *SVM labeled bags* and *SVM labeled 10min bags* - SVM algorithm when using multi-labeled bag labels as labels of all its instances for both types of bags.

From the first plot it can be clearly observed that the performance of the multi-instance algorithms does not significantly decrease by using larger experience sampling time intervals compared to the fully supervised baseline (*SVM all labels*). The performance is especially preserved in the case of 10minutes

bags. The only significant drop in the classification accuracy occurred on the PLCouple1 dataset when using the larger bags of 180 minutes experience sampling time interval. This is due to the unbalanced distribution of activities in the dataset. The most frequently occurring activities were included in literally all bags, which consequently led to the complete lack of negative bags for these activities. This effect is less prominent in the 10minutes bags, which enables competitive recognition rates even when using long experience sampling time intervals. This way, a user is required to recall only recent activities occurred in the last 10 minutes. This finding should additionally increase user acceptance of this annotation strategy.

Moreover, the plot shows a clear tendency that our extended multi-class version of multi-instance learning (*mc-miSVM labeled bags* and *mc-miSVM labeled 10min bags*) outperforms the standard multi-instance learning (*miSVM labeled bags* and *miSVM labeled 10min bags*) and the supervised baseline (*SVM labeled bags* and *SVM labeled 10min bags*).

9 Majority Voting Bags

Majority voting bags are labeled based on the largest activity appearing in a bag. That can, as in the case of single-labeled bags, introduce a certain amount of noise in negative bags, but of lower significance. The amount of noise in the bags is significantly decreased in this way. Thus, in the following experiments we explore capability of the standard version of miSVM algorithm introduced in Section 3 to cope with that reduced amount of noise in the bags.

Results. Figure 5 shows the performance of the standard multi-instance learning algorithm (*miSVM labeled bags* and *miSVM labeled 10min bags*) and the supervised baselines: 1) *SVM all labels* - standard SVM algorithm on fully labeled data and 2) *SVM labeled bags* and *SVM labeled 10min bags* - SVM algorithm when using majority voting bag labels as labels of all its instances.

As in the previous settings, discussed in Section 7 and Section 8, again the performance of the algorithms does not decrease when using larger experience sampling intervals up to 60 minutes compared to the fully supervised baseline *SVM all labels*. This clearly indicates the potential for reducing the level of typically annoying experience sampling disruptions.

In this particular labeling scenario, the labels provided on the bag level introduce almost no noise, because the labeling is based on majority voting. Thus, multi-instance learning might not be appropriate for this setting. As can be seen from the plot in Figure 5, the performance of *miSVM labeled bags* and *miSVM labeled 10min bags* is often lower than the performance of *SVM labeled bags* and *SVM labeled 10min bags* algorithms. This is due to the fact that multi-instance learning aims at finding the correct labeling of all instances in the bags. But, in this case, large classes dominate the bags, and it is not possible to recover the instance labels of small classes because there are very few, if any, bag labels provided for these small classes.

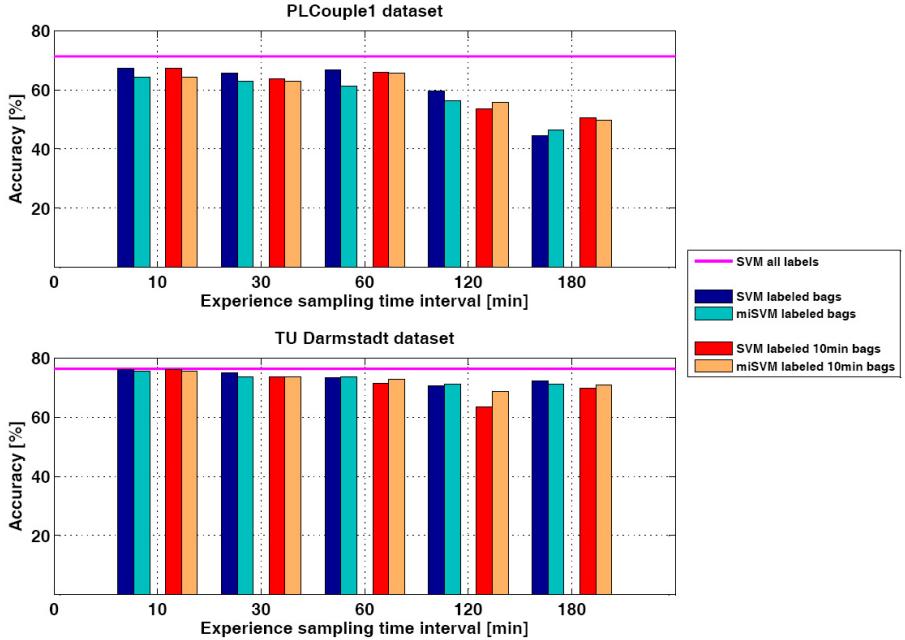


Fig. 5. Majority voting bags: Comparative performance of supervised baselines (*SVM all labels*, *SVM labeled bags*, *SVM labeled 10min bags*) and multi-instance learning approaches (*miSVM labeled bags*, *miSVM labeled 10min bags*) for different experience sampling time intervals in case of PLCouple1 (top) and TU Darmstadt datasets (bottom)

10 Discussion

Table 1 and Table 2 summarize the best results achieved by the evaluated algorithms in all three explored annotation strategies for the PLCouple1 and TU Darmstadt datasets, respectively.

For single-labeled bags, clearly the best performance is accomplished with *init-miSVM* algorithm, our extended version of the standard miSVM for the PLCouple1 dataset. In the case of the TU Darmstadt dataset, the best results are typically achieved with the *miSVM* algorithm. As already stated in Section 7 this is because of the poor performance of the *SVM few labels* algorithm (i.e. initialization phase) on this particular dataset.

For multi-labeled bags, the results indicate that our extended multi-class version of the standard miSVM algorithm, *mc-miSVM*, consistently yields the best recognition rates.

For majority voting bags, surprisingly, the best performance is achieved when applying the regular SVM algorithm on the bag instances without separation of positive and negative instances in noisy positive bags. Notably, the level of noise in this bag-of-activity label generation strategy is so low that multi-instance learning is not even necessary for achieving high recognition performance. The main drawback of this labeling strategy is that short events are typically missed

Table 1. PL Couple1 dataset: Comparison of the best results for single-labeled bags, multi-labeled bags and majority voting bags

Time interval	Single-labeled bags	Multi-labeled bags	Majority voting bags
10min	69.4%	init-miSVM	71.2%
30min	67.5%	init-miSVM	69.4%
60min	66.5%	init-miSVM	60.3%
120min	57.2%	init-miSVM	58.7%
180min	56.8%	init-miSVM	49.9%
		mc-miSVM	mc-miSVM
			67.3%
			65.7%
			66.8%
			59.6%
			50.6%
			SVM

Table 2. TU Darmstadt dataset: Comparison of the best results for single-labeled bags, multi-labeled bags and majority voting bags

Time interval	Single-labeled bags	Multi-labeled bags	Majority voting bags
10min	75.7%	SVM	75.6%
30min	74.5%	SVM	74.1%
60min	72.9%	miSVM	73.1%
120min	70.6%	miSVM	70.6%
180min	70.9%	miSVM	69.7%
		mc-miSVM	mc-miSVM
			76.2%
			75.0%
			73.5%
			71.2%
			miSVM
			72.3%
			SVM

and the classifier is unable to learn models for these events that might be very important for certain application domains.

We also compare all three annotation strategies (i.e. single-labeled bags, multi-labeled bags, and majority voting bags) by highlighting the recognition results that are in 2% interval around the highest scores (in Table 1 and Table 2) achieved with different experience sampling time intervals. For both datasets the recognition rates do not vary significantly for different annotation strategies. In the case of the TU Darmstadt dataset, all three annotation techniques perform surprisingly well and by increasing the experience sampling time interval from 10 minutes up to 180 minutes, the recognition performance is decreased by only 4.8%, 5.9%, and 3.5% for single-labeled bags, multi-labeled bags, and majority voting bags, respectively. Thus, our experimental results suggest that competitive recognition performance might be achievable by all three annotation strategies. That can significantly reduce the level of supervision in activity recognition.

In summary, we conclude that it is possible to attain high recognition performance even with very rare experience sampling prompts by extending multi-instance learning to handle different annotation strategies for activity recognition.

11 Conclusions and Future Work

This paper introduced a novel approach for reducing the required level of supervision in activity recognition. The approach is based on multi-instance learning. It enables automatic recognition of activities from sparsely labeled data.

To this end, we have proposed several novel extensions of multi-instance learning for handling different annotation strategies. We demonstrated the feasibility of the approach on two public datasets for three different labeling scenarios. All three scenarios support less constrained ways of annotating data by requiring neither detailed annotations nor precise start and end times of activities. The experimental results suggest that our extended multi-instance learning algorithms can obtain high recognition performance even though only coarse-grained annotations are provided. We strongly believe that this observation offers a number of still unexplored possibilities for lowering the annotation burden in activity recognition. The results presented in this paper are but a first step towards better scalability and applicability of activity recognition in real-world settings.

We plan to continue our work on decreasing the level of supervision in activity recognition by following several research directions. First, we want to further investigate the potential of multi-instance learning by using more expressive variants of miSVM algorithms, based on deterministic annealing that enables soft-assignments of labels to the instances in the bags. Second, we plan to evaluate the performance of the proposed multi-instance learning algorithm against the typically used semi-supervised techniques in the field of activity recognition. Third, we aim to extend our experiments further towards real-world scenarios by including users to participate in the proposed experience sampling strategies over longer periods of time. In order to additionally ease the necessary labeling efforts, varying bag sizes could be used where the bag boundaries would be placed at the recognized activity transitions.

Acknowledgments

The authors would like to express their gratitude to Beth Logan, Jennifer Healey, Emmanuel Munguia Tapia, Stephen Intille, and Tâm Huỳnh for providing access to the PLCouple1 dataset and the TU Darmstadt dataset. We would also like to thank Diane Larlus and Christian Wojek for stimulating discussions and comments. This work has been supported in part by Fraunhofer's Doktorandinnenprogramm.

References

1. Froehlich, J., Chen, M.Y., Consolvo, S., Harrison, B., Landay, J.A.: MyExperience: A System for In situ Tracing and Capturing of User Feedback on Mobile Phones. In: Proceedings of the 5th International Conference on Mobile Systems, Applications, and Services (MobiSys) (2007)
2. Huynh, T., Fritz, M., Schiele, B.: Discovery of Activity Patterns using Topic Models. In: Proceedings of the 10th International Conference on Ubiquitous Computing (UbiComp) (2008)
3. Maurer, U., Smailagic, A., Siewiorek, D., Deisher, M.: Activity Recognition and Monitoring Using Multiple Sensors on Different Body Positions. In: Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks (BSN) (2006)

4. Logan, B., Healey, J., Philipose, M., Tapia, E., Intille, S.: A Long-Term Evaluation of Sensing Modalities for Activity Recognition. In: Krumm, J., Abowd, G.D., Seneviratne, A., Strang, T. (eds.) *UbiComp 2007*. LNCS, vol. 4717, pp. 483–500. Springer, Heidelberg (2007),
http://architecture.mit.edu/house_n/data/PlaceLab/PlaceLab.htm
5. Laerhoven, K.V., Kilian, D., Schiele, B.: Using Rhythm Awareness in Long-Term Activity Recognition. In: Proceedings of the 12th IEEE International Symposium on Wearable Computers (ISWC) (2008)
6. Tapia, E.M., Intille, S.S., Larson, K.: Activity Recognition in the Home Using Simple and Ubiquitous Sensors. In: Ferscha, A., Mattern, F. (eds.) *PERVASIVE 2004*. LNCS, vol. 3001, pp. 158–175. Springer, Heidelberg (2004)
7. Lester, J., Choudhury, T., Borriello, G.: A Practical Approach to Recognizing Physical Activities. In: Fishkin, K.P., Schiele, B., Nixon, P., Quigley, A. (eds.) *PERVASIVE 2006*. LNCS, vol. 3968, pp. 1–16. Springer, Heidelberg (2006)
8. Bao, L., Intille, S.: Activity Recognition from User-Annotated Acceleration Data. In: Ferscha, A., Mattern, F. (eds.) *PERVASIVE 2004*. LNCS, vol. 3001, pp. 1–17. Springer, Heidelberg (2004)
9. Wyatt, D., Philipose, M., Choudhury, T.: Unsupervised Activity Recognition Using Automatically Mined Common Sense. In: Proceedings of the 20th National Conference on Artificial Intelligence (AAAI) (2005)
10. Duong, T.V., Bui, H.H., Phung, D.Q., Venkatesh, S.: Activity Recognition and Abnormality Detection with the Switching Hidden Semi-Markov Model. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) (2005)
11. Ogris, G., Stiefmeier, T., Lukowicz, P., Tröster, G.: Using a complex Multi-modal On-body Sensor System for Activity Spotting. In: Proceedings of the 12th IEEE International Symposium on Wearable Computers (ISWC) (2008)
12. Oliver, N., Horvitz, E.: A Comparison of HMMs and Dynamic Bayesian Networks for Recognizing Office Activities. In: Ardissono, L., Brna, P., Mitrović, A. (eds.) *UM 2005*. LNCS, vol. 3538, pp. 199–209. Springer, Heidelberg (2005)
13. Ravi, N., Dandekar, N., Mysore, P., Littman, M.: Activity Recognition from Accelerometer Data. In: Proceedings of the 17th Conference on Innovative Applications of Artificial Intelligence (IAAI) (2005)
14. Wren, C.R., Tapia, E.M.: Toward Scalable Activity Recognition for Sensor Networks. In: Hazas, M., Krumm, J., Strang, T. (eds.) *LoCA 2006*. LNCS, vol. 3987, pp. 168–185. Springer, Heidelberg (2006)
15. Aghajan, H., Augusto, J.C., Wu, C., McCullagh, P., Walkden, J.A.: Distributed Vision-Based Accident Management for Assisted Living. In: Okadome, T., Yamazaki, T., Makhtari, M. (eds.) *ICOST 2007*. LNCS, vol. 4541, pp. 196–205. Springer, Heidelberg (2007)
16. Wang, S., Pentney, W., Popescu, A.M., Choudhury, T., Philipose, M.: Common Sense Based Joint Training of Human Activity Recognizers. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI) (2007)
17. Huynh, T., Blanke, U., Schiele, B.: Scalable Recognition of Daily Activities with Wearable Sensors. In: Hightower, J., Schiele, B., Strang, T. (eds.) *LoCA 2007*. LNCS, vol. 4718, pp. 50–67. Springer, Heidelberg (2007)
18. Minnen, D., Starner, T., Essa, I., Isbell, C.: Discovering Characteristic Actions from On-Body Sensor Data. In: Proceedings of the 10th IEEE International Symposium on Wearable Computers (ISWC) (2006)

19. Stikic, M., Laerhoven, K.V., Schiele, B.: Exploring Semi-Supervised and Active Learning for Activity Recognition. In: Proceedings of the 12th IEEE International Symposium on Wearable Computers (ISWC) (2008)
20. Anderson, B., Moore, A.: Active Learning for Hidden Markov Models: Objective Functions and Algorithms. In: Proceedings of the 22nd International Conference on Machine Learning (ICML) (2005)
21. Kapoor, A., Horvitz, E.: On Discarding, Caching, and Recalling Samples in Active Learning. In: Proceedings of the Conference on Uncertainty and Artificial Intelligence (UAI) (2007)
22. Kapoor, A., Horvitz, E.: Experience Sampling for Building Predictive User Models: A Comparative Study. In: Proceedings of the Conference on Human Factors in Computing Systems (CHI) (2008)
23. Chapelle, O., Schölkopf, B., Zien, A.: Semi-Supervised Learning. MIT Press, Cambridge (2006)
24. Guan, D., Yuan, W., Lee, Y.K., Gavrilov, A., Lee, S.: Activity Recognition Based on Semi-supervised Learning. In: Proceedings of the 13th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA) (2007)
25. Subramanya, A., Raj, A., Blimes, J., Fox, D.: Recognizing Activities and Spatial Context Using Wearable Sensors. In: Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI) (2006)
26. Mahdaviani, M., Choudhury, T.: Fast and Scalable Training of Semi-Supervised CRFs with Application to Activity Recognition. In: Proceedings of the 21st Annual Conference on Neural Information Processing Systems (NIPS) (2007)
27. Hoey, J., Poupart, P., Boutilier, C., Mihailidis, A.: Semi-supervised learning of a pomdp model of patientcaregiver interactions. In: Proceedings of the IJCAI Workshop on Modeling Others from Observations (2005)
28. Zhou, Z.H.: Multi-Instance Learning: A Survey. Technical report, AI Lab, Department of Computer Science and Technology, Nanjing University (2004)
29. Andrews, S., Tschantaridis, I., Hofmann, T.: Support Vector Machines for Multiple-Instance Learning. In: Proceedings of the 17th Annual Conference on Neural Information Processing Systems (NIPS) (2003)
30. Vapnik, V.: Statistical learning theory. Wiley and Sons, NY (1998)
31. Partridge, K., Golle, P.: On Using Existing Time-Use Study Data for Ubiquitous Computing Applications. In: Proceedings of the 10th International Conference on Ubiquitous Computing (UbiComp) (2008)

A Framework for Creating and Using Maps of Privately Owned Spaces

Steven Shafer

Microsoft Corp., One Microsoft Way, Redmond WA 98052, USA
StevenSh@Microsoft.com

Abstract. The Facility Map Framework (FMF) is a new system of specifications and software for maps of privately owned spaces – campus, building, room detail, and shelf or rack. Based on a generic object model, FMF supports map authoring from CAD or other drawings, asset databases and location sensor systems, map access through browsers or applications, and development of application programs. FMF can integrate existing system components to support interoperability and extensibility, and has features to address a number of additional engineering problems in location systems for private spaces. FMF is a work in progress, with many key elements implemented and others currently under design. Several demos are now in place, representing a variety of different scenarios for using indoor maps. This paper gives an overview of the system components and a number of the engineering challenges and solutions; it represents a progress report on this evolving project.

Keywords: Indoor maps, Indoor location, Floorplans, Location-Awareness.

1 Introduction

One of the ironies of ubiquitous computing is the many wonderful research demos that never run outside of the lab in which they are created. There can be many reasons for this, and one is that real homes and businesses must address issues of deployment, maintenance, and training that are not the subject of the research demo. This work addresses precisely those issues, in a very specific domain that is fundamental to ubiquitous computing – the representation of location information in private spaces. Here, by *location information* we mean object-model maps along with semantic notations and rosters of assets that appear in the mapped areas; and by *private spaces* we mean any spaces that are below the level of public data (streets and political entities), i.e. the campus, the building, the room, and even the shelf or rack.

The Facility Map Framework was developed to try to overcome the barriers to deployment of ubiquitous computing, in a range of settings though emphasizing the particular needs of large enterprises, to support a broad range of applications, providing a high level of integration of location information from diverse sources.

Of course, there are many research and commercial systems that use floorplans, asset roster databases, and/or indoor location sensors. Such systems generally have one or more of the following shortcomings:

1. Map drawing is treated as a bitmap, with no ability to name spaces of interest;
2. System uses proprietary representation, with no extensibility capability; and/or
3. System has characteristics unique to a specific sensor or database.

There are some research systems that overcome some of these difficulties, notably [1] and [2]. The Facility Map Framework (FMF) includes many of the capabilities of these systems and adds:

1. A route network for driving, walking, or other transportation modalities;
2. Authoring tools for large-scale and third-party map suite development; and
3. Deployment tools for hosting integrated map suites.

There are also industry initiatives for standardizing Building Information Models (BIMs), e.g. [3] and [4]. These models are far more extensive than FMF, with much more verbose representation, and are therefore suitable for targeted use (such as construction and energy management), but are not aimed at ubiquitous computing scenarios such as mobile devices and end-user applications, as is FMF.

Other prior research in indoor location technology has primarily been in the development of new sensors [5] [6] and novel “closed” (non-extensible) applications [7] [8]. By contrast, FMF is an infrastructure that can absorb and publish sensor data, and can be used to create applications using extensible components. FMF aims to move toward a “marketplace” for interoperable private space location components and applications, particularly for indoors and campuses.

2 Facility Map Framework Overview

Figure 1 illustrates the vision of FMF. Location sources include drawings – typically CAD drawings or CAFM (Computer-Aided Facility Management) systems – as well as asset roster databases, location sensors in the facility infrastructure, and “own location” sensors on mobile devices. The consumers include customers, business users, and enterprise or facility operations staff, using servers, desktop systems, and mobile devices. The FMF approach is to provide a single forum for collecting information from all the sources, and distributing it to all the consumers through a common repository. FMF includes authoring tools, program development tools, and deployment tools, all unified by a common object model expressed in several forms, most notably a new XML schema called the Facility Map Modeling Language (FMMML).

FMF is aimed at supporting these generic capabilities:

1. Map Drawing and Map as Dashboard – Displaying places and assets, including real-time asset location and properties, and using the map display for interactive display and control of assets.
2. Routing and Wayfinding – Computing and displaying paths, and providing navigation information to the traveler.
3. Map Integration and Federation – Presenting maps with a uniform view, though they may include diverse types of data provided by diverse authorities.
4. Dedicated and Promiscuous Applications – Dedicated applications (designed to run at a specific site) should be efficient and powerful; while promiscuous applications (designed to operate at any site with an FMF model) should support common map capabilities.

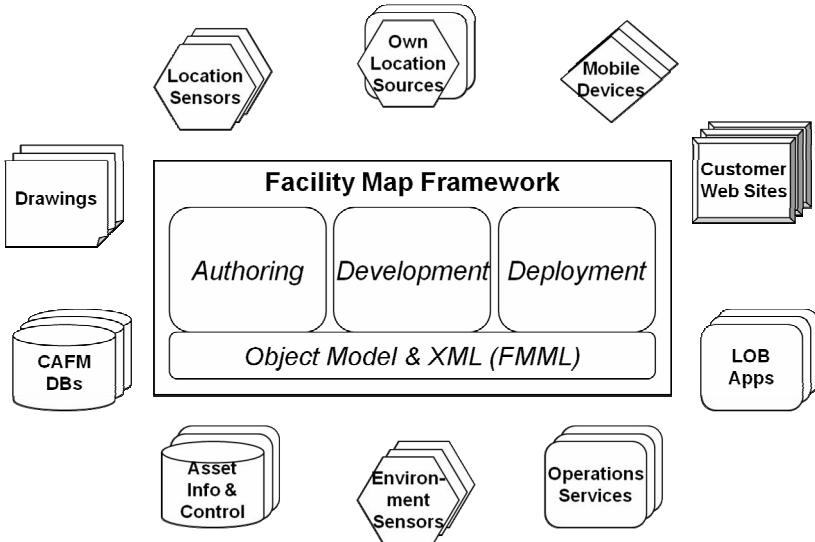


Fig. 1. FMF is a single repository integrating information from multiple sources and distributing it to multiple consumers, unified by a common object model

The vision of FMF is that by supporting all of these functions in a single framework that integrates all available location data sources, an enterprise can create and deploy a single FMF model of its facilities, to support all location-aware applications. This would be a great improvement over typical scenarios of today, in which an enterprise must deploy the same data in numerous variations, each of which supports only a single distinct application or operations group or business function.

3 The Facility Map Object Model

The Facility Map Object Model (FMOM) unifies all components of FMF. The key high-level objects are illustrated in Figure 2. A *Facility Map* is a loadable unit, for example a single FMML file. A Facility Map may contain one or more *Layouts*; each Layout represents a single “map sheet” or object roster. In typical use, a building will be a single Facility Map in an FMML file, defining a Layout for each floor; each type of asset might also have its own FMML file, with a single Layout for the roster. A Layout can represent a *plan* or *elevation* view defining a coordinate system (x-y or x-z, respectively); a *set* of objects such as a roster; a *template* containing prototypes (see below); or a *stack* defining a layered arrangement such as the floors of a building.

3.1 Map Objects

Map Objects defined in a Layout include *Places*, *Assets*, *Persons* (Asset with a privacy policy), and *Artworks* (no semantics, just a drawing annotation). Map Objects also include several types of *Authoring Mark*, that are used in the map authoring

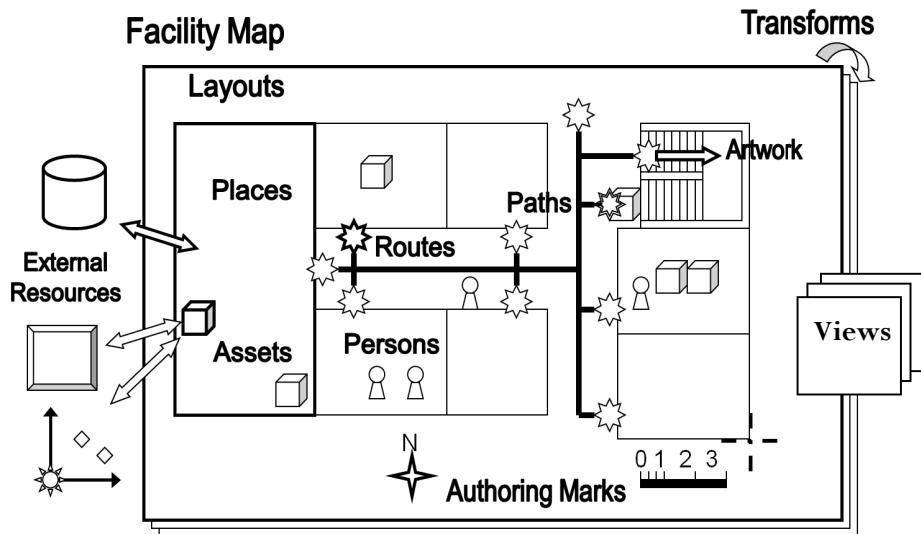


Fig. 2. Facility Maps, Layouts and Other Map Objects in the Facility Map Framework

process and/or routing; *Authoring Marks* include *Scale Marks*, *Compass Marks*, *Registration Marks*, *Calibration Marks* (for registration with external systems), *Cross-Section Marks* (e.g. for linking to a front view of a computer rack or retail shelf), *GeoMarks* (for alignment with latitude/longitude), and *Route Connectors* and *Route Segments* (for transportation networks). *Authoring Marks* typically hold metadata but are invisible in normal drawings. Layouts also define *Paths* (see below description of routing), *Views* (appearance stylesheets, most importantly the “default View” to be used in the absence of application-specific instructions), and *Transforms* between the coordinate systems of this Layout and other Layouts.

Prototypes. Many map and GIS systems use classes for map objects in which each class is defined by a schema; individual objects are instances of these classes, much like modern object-oriented programming languages. A deployed map must have both schema and object definitions. FMF works differently, using *Prototypes* as in prototype-based programming [9]. A prototype is a map object like any other. All map objects may have a pointer to a prototype object, and if a property is desired from an object and the object does not define it, then the prototype is consulted to see if it can supply a “default” value. Each prototype thus serves the role of a “class” definition by supplying default values for properties of the objects that use it. The prototype may itself have a prototype, so that there can be a hierarchy much like a class hierarchy. Prototypes offer important advantages over schemas for map systems by simplifying both the definition and object model semantics. A schema-based map system, for example, must deploy both the schema and the object definitions; a prototype system only deploys object definitions, which include the prototypes as well as the map objects in the same representation. Also, for example, a schema-based system would need two different primitives for “color all offices blue” and “color office 123 blue”, because the former is a class whereas the latter is an object; but in FMF,

“color all offices blue” is expressed as “color the office prototype object blue”, so the primitive “color <object> <color>” works for all offices or for a specific office.

A set of Map Objects may also be associated with a *Prototype Map*, which is another set of corresponding Map Objects used to supply missing values for the original objects. For example, a roster of Persons could provide static data for each individual Person; and it could also indicate a Prototype Map that is a real-time sensor system to associate a current location with each Person. If an application requests a static data element it would be supplied by the Person roster; and if it requested the Person’s location, that request would fall through the static data roster to be supplied by the associated Prototype Map.

3.2 Relationships

A Facility Map or Layout may define a namespace for Map Objects, called a Dictionary in FMF. A Dictionary may indicate that names that start and/or end with a given prefix or suffix should be referred to a Dictionary in a different Facility Map or Layout – in this way, for example, a Dictionary for a company could indicate that names beginning with “A-“ should be referred to the Facility Map for Building A. Similarly, the Dictionary for Building A could indicate that all of its Map Objects’ names can be prefixed with “A-“ to create a name within the namespace of the company. A pointer to a Map Object is called a Reference in FMF. References indicate the name of the Map Object, and if it is not in the same Facility Map or Layout as the Reference itself, then the Reference includes information about the Facility Map or Layout whose Dictionary defines the name. Information about a Facility Map includes not only the source (e.g. filename or URL), but also the type of source and optionally additional parameters needed to access that source (such as authorization parameters). FMF has the following map types:

1. FMML files (specified by the filename). FMML files can also be used to access data stored in a SQL database, a spreadsheet, or an LDAP directory, as described later.
2. SOAP web services [10] conforming to an interface specification called Facility Map Over Web Service (FMOWS), described later in this paper.
3. A built-in Layout for the WGS84 latitude/longitude specification for the Earth (this is the coordinate system used by GPS and most modern GIS systems).
4. An interface is provided in the Facility Map API (described below) for an application programmer to define other types of source, though in most cases it would be wisest for other sources to either produce an FMML file (if the data is static) or to be accessed through an FMOWS web service (if dynamic).

The interchangeable representation of map drawings and rosters, combined with the uniform access paradigm for References to all types of maps, allows an FMF deployment to take the form of a suite of Facility Maps with pointers among them to indicate key relationships. This allows decentralized representation of enterprise data, addressing the “Jurisdiction Problem”. The Jurisdiction Problem is that enterprise data administered by many different units within the organization, and each claims sovereignty over its designated domain of data. For example, one group will own the CAD drawings and campus maps; another group may own the printer locations; another the

locations of wireless access points and network routing; and another the “seating chart” of office assignments. When the data is centralized, that central system becomes a bottleneck for updating information, and it is typically slow and perhaps inconvenient to access it. Work orders may be needed to update the data, and business units will develop workarounds such as unregistered local data stores. The jurisdiction problem is compounded because at the top level of the enterprise, jurisdiction is usually defined by type of data (printers, floorplans, etc.) across all spatial domains; whereas at lower levels of the enterprise, jurisdiction is defined by spatial domains (Lab A-1, Lab A-2, Meeting Rooms B-1 to B-6), including all the various assets found within each domain, including furniture, room partitions, and equipment. In today’s enterprises, it is virtually impossible to overlay the various data stores, or to cross between the enterprise-level and local-level data stores. FMF can unify these models, allowing each data authority to maintain its own data store.

FMF implements numerous relationships between Layouts (or between Layouts and other Map Objects). Campus maps can be georeferenced to latitude/longitude. Their buildings have a detail/context relationship with building maps, just as a room on a building floorplan map may have a detail/context relationship with a room detail map. The campus maps are coplanar. The floor Layouts for a building are registered but parallel; they may be grouped into an ordered stack for the building; and they may be grouped into an exploded view of the entire building. The room detail may define a number of rack objects, each with a cross-section link to a front view of that rack. The front view is derived from a template for the type of rack. The asset roster for the building can be viewed as an overlay; each asset has a prototype and may belong to a person. The asset locations come from a real-time sensor system, and the locations themselves indicate places on the various racks. Each asset’s description is a union of the static data in the asset roster, with the dynamic location data from the sensor system. All of these relationships can be embodied in various tools or models in the Facility Map Framework, as appropriate.

3.3 Location and Geometry

Each Map Object in FMF may specify a *Location*, which consists of two parts: (1) zero or more *InPlace* References to Places in which this object is located; and/or (2) zero or more *Polygons*. An *InPlace* Reference indicates that this object is known to be within the indicated Place, but perhaps at an indeterminate or unknown position within that Place; it indicates topological containment only. A *Polygon* specifies one or more XY points within a given Layout’s coordinate system; the Layout may be specified by an explicit Reference, or it is assumed to be the Layout that defines the Map Object whose Location this is. Thus, for example, a Facility Map defining a roster of printers would indicate the Location of each printer object, either by giving an *InPlace* Reference (such as “In Place B-32” or “In Place 32 in FMML map B.xml”), or by giving a *Polygon* (or point) (such as “32,14” or “32,14 in Layout 2 of FMML map B.xml”).

Geometry in FMF is given in 2D rather than 3D. For the vast majority of business and wayfinding applications, 3D is not computationally necessary, and the availability in FMF of cross-sections, building stacks, and exploded views covers many of the most common uses of the third dimension for facility maps. 3D is much more

complex to represent, slower in computations, and the models are much more expensive in size and labor cost. Therefore, at present, FMF is a 2D system. There is limited representation of a third dimension, enough to interpret an XYZ point within a building as an element of a specific floor, but FMOM is not a full 3D model.

In FMF, the geometric coordinate system used for a given Layout, say a floorplan, will usually be derived from the CAD drawing used as a basis for the Layout definition. Each (plan or elevation) Layout thus defines its own XY coordinate universe, related to other Layouts only through explicit Transforms. In contrast, many GIS-based systems use latitude/longitude throughout, including for coordinates within buildings. The lat/lon approach has the advantage that all computations and representations occur within a single XYZ universe, eliminating the need to explicitly discover and apply local coordinate transforms. Indeed, many GIS systems can only represent lat/lon and cannot represent local “engineering” coordinate systems. This approach was rejected in FMF because indoor coordinates and distances typically will reside in the 9th to 10th significant digit of lat/lon coordinates, resulting in numerical instability when double-precision transformations and calculations are applied; and because of limited availability of georeferences for many CAD and similar drawings. Instead, FMF takes a more robotics-like approach of defining local coordinate systems, associated with Layouts, rather than relying on a single universal coordinate system. The relevant Transforms can be easily computed during map authoring, and stored in the Layout itself.

At present, we model the polygonal outlines of Places, but not the architectural features present in the CAD data, such as actual door swings and widths. Thus, we would be unable at this time to support applications such as [11] requiring such detail.

3.4 Other FMOM Features

Each Map Object can define a set of *Properties* whose semantics are up to the system designer; they are not interpreted specifically by FMF. The value of a Property can be a string, a Reference to a Map Object, or a list of nested Properties (such as “a.x”, “a.y”, and “a.z”). Properties can be used by applications, and can be used to determine appearance specifications and/or text strings to display on a map drawing.

FMF uses the approach of a predefined route network, drawn onto the CAD data during the map authoring process. This is in contrast to automatically computed routes, as in the system [2]. Drawing the route network allows the author to decide where the travel lines should be, how to model unusual-shaped spaces, and where doors and other access points should be defined. The author also indicates metadata, including vertical connectors (stairs, elevators, etc.), external connectors (e.g. building external door connecting to the campus map route network), and modes of transport available on a given segment (e.g. walking, wheelchair, bike, auto).

When a map is used as a “dashboard”, clicking on an object’s name or icon should provide information, commands, and possibly a window for querying or controlling the physical object. FMF accomplishes this through *Proxies*. Each Map Object or list of Map Objects can specify a Proxy; when this object is displayed with the FMF UI (see below), and its name or icon is clicked by the user, FMF will query the indicated Proxy for a list of commands to provide. Typically, the Proxy will be a SOAP Web Service implementing the Facility Map Object Proxy (FMOP) interface. When the

user then selects a command from the list, the Proxy indicates whether the action should be to display a set of Properties, open a browser window with a given URL, or issue a command back to the Proxy.

4 The Facility Map API

The Facility Map Object Model defines an abstract data representation. The Facility Map API (FMAPI) is the programming model of FMF, and is illustrated in Figure 3. FMF is a client-side map system – the application retrieves complete Facility Maps, and performs all computations internally. Thus, there is no means for retrieving a specific Map Object; instead, the client retrieves an entire Facility Map using FMAPI, and applies the computational capabilities of FMAPI to analyze or display it.

4.1 Map Providers

The lowest level of FMAPI is a set of Map Providers. Actually, there are only three providers – an FMML file reader/writer, an FMOWS web service client, and the built-in map representing the Earth in WGS84 lat/lon. However, an FMML file called a Pointer can be used to provide data from another source. A Pointer is an FMML file that contains all the metadata about a set of objects (e.g. what icon to use in displaying them); but, instead of defining the actual roster within the FMML file, it has instead a description of how to obtain the data from an external source. FMAPI defines Pointers for SQL Server databases (each row of the indicated table defines one Map Object), MS Excel spreadsheets (each row of the indicated table defines one Map Object), and LDAP directories (each object defines one Map Object). When using a Pointer, the name or ID of each Map Object is obtained from the external source, and associated values in the external data can be assigned to the Location and Properties of the created Map Objects. Pointers are presently read-only; FMAPI does not define a mechanism for writing data back to the external source (of course, an application can do so through its own code).

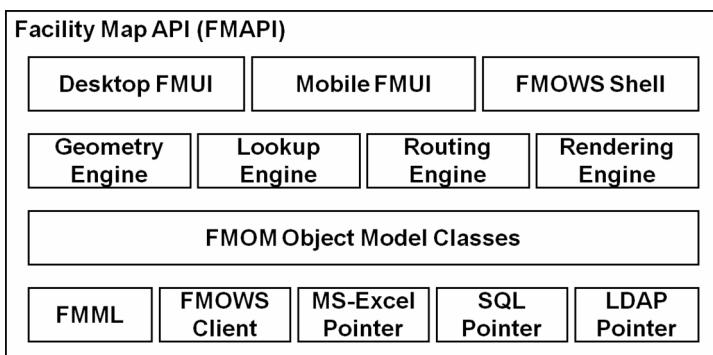


Fig. 3. The Facility Map API provides four layers of function: (from the bottom) Map provisioning, the FMOM model of the map data, computational engines, and exposure to the user and to other applications

4.2 The FMOM Object Model in FMAPI

The next level of FMAPI is the FMOM object model, which is a straightforward expression of FMOM as program objects (FMAPI is written in C# for .Net). While the data itself is a direct implementation of the FMOM specification, the code for accessing a field of a Map Object is somewhat complex. First, the Map Object is consulted; if the property is not found, the Prototype object is consulted; if still not found, then each Prototype Map defined for the list containing the Map Object is consulted to see if the Prototype Map defines a corresponding Map Object with the desired property. FMOM includes some features for factoring out common elements from lists of Map Objects, and one of these is the ability to associate a Prototype with all the Map Objects in a list.

In addition to all these lookups, a Map Object can specify an *ItIs* Reference to another Map Object, indicating that all operations upon this object should in fact be carried out on that other object. *ItIs* References are used to implement a roster of aliases for Map Objects defined elsewhere, for example associating Global Location Numbers (GLNs) [12] with corresponding Places in the enterprise. If a Map Object defines *ItIs*, then all accesses to its members are redirected as indicated.

4.3 Computational Engines

FMAPI provides a number of “engines” for computing important things about the map. Map semantics can be rather complex, so these computational engines are key to exposing the map functionally to application programmers.

The *Geometry Engine* provides a library for performing computations on points and Polygons (including line segments). The primary functions are distance, containment, projection, intersection, and coordinate system transformation. The lower level of the Geometry Engine assumes that the primitives exist within the same coordinate system; the higher level accepts Layouts as parameters along with the coordinates, and performs transformations when required.

The *Lookup Engine* performs the mapping of name (or ID) strings to Map Objects and vice versa. The Lookup Engine has a mechanism for well-known Facility Maps, for example “Acme:A-113” could mean the Acme Corp., building A, room 113. The Acme Facility Map would probably delegate the string “113” in turn to the Facility Map for Building A, so the application resolving this name would end up with the Map Object “113” within Building A’s FMML definition. Given a Map Object, the Lookup Engine can also provide a Reference to it, or it can provide the object name as a string, including reverse dictionary delegation and alias resolution. From the Map Object above, the Lookup Engine could generate the name “Acme:A-113”.

The *Routing Engine* performs path planning, to determine a Path that conforms to the route network defined on the Layout. Each endpoint is specified by a *PathEndpoint*, which can be either a Reference or a Location, and can be converted to and from a simple string. Once both PathEndpoints have a connection established to the route network, a standard A* search is performed to find a path. This may include crossing building floors using vertical connectors, crossing maps (such as origin building to campus map to destination building), etc. The bits and pieces of the route network that form the discovered route are recorded in an object called a *Path*. Thus,

the route network expresses all the possible ways to get from anywhere to anywhere, whereas a Path is a specific way to get from one PathEndpoint to another. A *Path-Stepper* allows an application to traverse a Path in convenient pieces, for example for generating instructions or displaying travel steps to a traveler.

The *Rendering Engine* can draw a map onto a UI, an image file, or a printed page. The application first defines a Presentation, which is a specification of exactly what is to be drawn and how. A Presentation can be as simple as a single Facility Map or Layout, allowing the default View to define the appearance of everything; or it can specify other Layouts to overlay (such as object rosters or detail drawings), specific appearances for specific objects or object classes, Paths to draw, and geometric transformations to use to align different Layouts on the drawing. The Rendering Engine can adapt a drawing based on criteria such as screen size, so for example it can select an alternate View for a small screen (e.g. mobile device) if such a View is indicated in the Layout. FMAPI therefore uses the same renderer for both desktop and mobile applications.

4.4 Map Exposure

The top layer of FMAPI is the built-in ways for an application to expose map data to a user or to another program.

FMAPI includes the *Facility Map UI* (FMUI), a UI widget that renders a given Presentation and interprets mouse movements and clicks to implement a full-featured map dashboard. Through FMUI, the user can zoom and pan, shift the view to other related Layouts (such as up down the floors of a building, out to a campus map, or in to a cross-section view of a computer rack), draw a path from one place to another, turn overlays on and off, invoke object Proxies, and many other functions. The application programmer need merely place an FMUI control onto the program's UI design, and assigns an initial Layout to be displayed. The application program can also disable any or all of the FMUI built-in features, and can intercept mouse clicks and moves, and the right-click "context menu", to perform its own processing or to supplement the built-in functions of FMUI. There are desktop and mobile versions of FMUI.

FMAPI also includes a shell for the *Facility Map Over Web Service* (FMOWS) SOAP interface. This can be used for providing data that is derived from a sensor system, for example, or from another external source not accessible through an FMML Pointer (above). The FMOWS developer must publish the SOAP web service on the hosting system. FMOWS implements additional interface methods to support browser-based maps (described below).

5 Authoring Facility Maps

In many systems that use floorplan drawings, the drawing is represented simply as a bitmap, usually for display purposes, and features are then drawn on top of it without any semantic interpretation. In some research systems, researchers will draw polygons for the rooms to create an object model; similarly, in some commercial systems, their own staff will do this as a service with an hourly labor charge. FMF has a different model, based on standard polylines.

*Polyline*s are polygons drawn onto a CAD file, usually as a layer of the file. They are also known as BOMA lines in the US in honor of the organization whose specifications they are generally required to follow [13] [14]. Polylining is generally performed either in an architect's firm, or by a firm that manages architectural drawings, using tools such as add-ins to CAD programs. There are usually at least several such firms in any major city, and they are contracted to manage the drawings, CAFM (Computer Assisted Facility Management), and IWMS (Integrated Workplace Management Systems) for larger business enterprises and public facilities.

The approach in FMF for authoring maps from drawings begins with such polylines, as a CAD layer, or as data in a CAFM or IWMS system that can be exported into a CAD file or a VISIO drawing. The map author then applies the *Facility Map Editor* (FMED) in FMF to create the FMML file from this drawing. FMED is a VISIO stencil, including a number of master shapes as well as a body of code behind those shapes. The map author creates a VISIO file, imports the source map drawing (as CAD or VISIO), and opens the FMED stencil. The FMED master shapes on the stencil include *Information Blocks*, *Authoring Marks*, and *Route Marks*. Information Blocks store metadata about Map Objects; Authoring Marks store metadata about the drawing itself; and Route Marks define the route network. Finally, FMED provides a command to export the model into an FMML file. The VISIO file with metadata annotations can be saved as a standard VISIO file (we call it an *FM-VISIO* file to indicate that it has the FMED stencil and associated metadata for the map).

6 Deploying Facility Maps

A complete FMF map deployment includes FMML files from map drawings, FMML Pointer files and associated external data stores, and FMOWS web services for internally generated data (such as from sensor systems). All this data can be integrated on demand by FMAPI. To provide a single point of access for users, a Directory Pointer can be used. This would be an FMML file for an enterprise that implements the top-level Facility Map; rather than defining the campus and building maps, it simply points to the directories containing each type of FMML map files. The Directory Pointer can automatically create all the implied links to those maps, such as name dictionary delegation, based on the filenames of the files in the directory. A complete deployment is illustrated in Figure 4.

Internal users on the enterprise network can access the individual components of the map suite directly. The References in each FMML component will contain all necessary details.

External partner applications are not on the enterprise network, so they cannot access the resources directly. Instead, the enterprise deploys an FMOWS web service to supply the individual Facility Maps comprised in the map suite. This FMOWS web supplies maps in FMML to the requesting application. Of course, the Facility Maps contain pointers to the other maps (etc.) on the enterprise network, and these pointers are unusable outside the hosting enterprise; therefore, FMOWS has a mechanism to translate from the enterprise resource descriptions to a standardized externally usable form. The client application thus sees everything in the map suite as components of the FMOWS server for the enterprise.

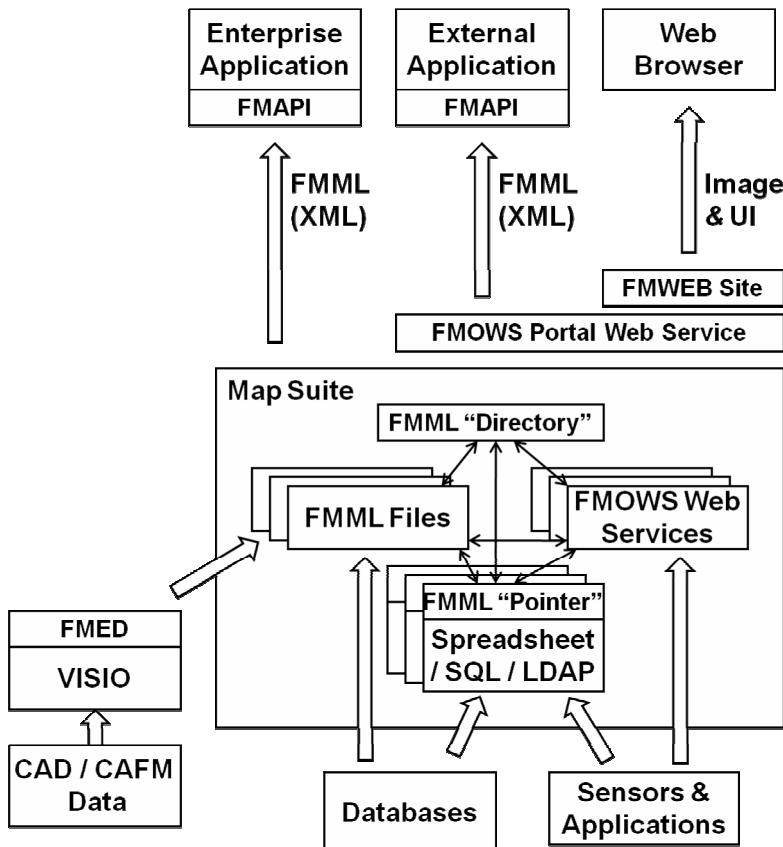


Fig. 4. A complete deployment of an FMF map suite may include FMMML maps from drawings, FMMML Pointers to databases, FMOWS web services for sensor systems, an FMMML Directory Pointer, and FMOWS and FMWEB portals

Web browsers require additional mechanism. The FMMML representations are too complex to parse and process in a web page script, and FMAPI is not available (yet!) in any web scripting language. Therefore, FMF includes a *Facility Map Web Application* (FMWEB) to support browsing of maps. A browser requests a map drawing from FMWEB, where the parameters attached to the URL specify exactly what is desired. FMWEB will create the specified drawing, and it returns to the browser the drawing and information for the map UI in the browser. FMWEB uses the underlying FMOWS web service to create the drawing; FMOWS creates the drawing as an image file, and returns its URL and the associated UI information to FMWEB. FMWEB then constructs the web page for the browser to implement an interactive map. The functions of this map are much more limited than, for example, the FMUI interface in FMAPI; however, it is still quite a useful map.

The FMOWS + FMWEB combination is referred to as a *Portal* for the FMF map suite. The Portal requires an externally available directory for the map image files.

7 FMF Examples and Demonstrations

FMF is an evolving research prototype; it also serves as a proposal-by-example for a ubiquitous computing infrastructure component. As such, it should meet numerous criteria, including: Technical feasibility, Business feasibility, Generality, and Usefulness. If any of these criteria cannot be met, then the infrastructure may not be broadly adopted. By presenting some examples and demonstrations with FMF, we will attempt to portray how it performs in these regards.

7.1 Example Facility Models

Facility Sampler. The first example model we call the Facility Sampler. This began as a small floorplan to illustrate many cases for designing a route network, and it has grown to include a tiny campus, one building map with two floors, a few “open spaces”, a stairway and an elevator, a lab room with a detail map, and several computer racks in the lab. The model comprises four Facility Map FMML files (campus, building, lab, and rack template). In a real enterprise, the campus and building maps might be administered by the facility management group, the lab by a lab manager, and the rack template by a manufacturer or standards body. In addition, there are overlay maps for people’s office assignments, people’s current locations, and office telephones. These maps, as drawn by FMAPI, are illustrated in Figure 5.

The Facility Sampler deployment includes directories for FM-VISIO files, FMML files, icon image files, and FMWEB-generated image files. There are also an FMOWS web service and an FMWEB web application serving as a Portal for the Facility Sampler, and the overlay data (object names and locations) is stored in spreadsheet files. Statistics such as file size and authoring time are not especially relevant for the Facility Sampler because the data is all artificially simple, there were no CAD files to begin with, and the Facility Sampler was developed a bit at a time.

Microsoft Buildings. Our most comprehensive examples to date have been two Microsoft buildings, which we will refer to as Building A and Building B. Table 1 presents some of the key statistics for these two buildings and their models, which we consider representative for commercial office space.

Several remarks are in order concerning Table 1:

1. At Microsoft, Building A is typical for moderate-to-small buildings, and Building B is representative of the larger buildings.
2. The FMML file size is critical, since this file must be sent to every application that requests the building map. The design goal for FMF was for a maximum size of roughly 1MB to keep transmission time and bandwidth requirements modest, particularly when sending the maps to mobile devices. FMML for details of all meeting rooms at Microsoft is about 0.5 MB, and for all printers is currently about 1.6 MB, a bit larger than we would like.
3. Microsoft has clean, up-to-date CAD drawings, and uses a CAFM system, so polylining has already been professionally performed and the polylines are available; polylining time is therefore excluded from Table 1. The authoring time for each building represents about 1.5 hours of routine tasks (such as file



Fig. 5. The Facility Sampler comprises a campus map, a building map with two floors, a lab detail map, and a set of computer racks with templates. Map appearances can be tailored for a small screen such as a mobile device, as shown here for the first floor map.

Table 1. Key statistics for the FMF models of Buildings A and B at Microsoft Corp

Building	Size (ft ²)	Number of offices	CAD file size	VISIO file size	FMML file size	FMF authoring time
A	114,000	391	5.4 MB	1.6 MB	0.5 MB	4 hours
B	278,000	738	7.5 MB	3.0 MB	1.1 MB	11 hours

copying, generating spreadsheets from databases, entering metadata into FMED, etc.); and the remainder of the time is for drawing the route network with FMED.

4. Our goal is to reduce the authoring process to a commercially available service that can be provided by architectural firms and by CAFM and IWMS companies. Based on our very limited experience to date, we are aiming for a price point of approximately \$2/office (hopefully even less in the future, we would like to achieve \$1/office). For one project, the building CAD is very poor, and in addition, there are no existing polylines – the total cost to update the CAD and add polylines is roughly equal to the cost of the FMF authoring, so for that project the total cost may be roughly \$4/office. Campus map models are relatively more expensive, as they can contain quite complex route networks, and the source data may be very limited.
5. It is difficult to find comparable authoring statistics for other building-model systems in the research literature, but [2] does provide some relevant information. There, defining a building model of approx. 120 rooms required “a few hours” for the ground floor and “less time” on the second and third floors. The bulk of the time appears to have been for drawing polygons (similar to polylines); that system does not store a predefined route network.

7.2 Demos and Applications

So far, we have a small number of diverse demos and applications of FMF. These are all works in progress.

GoMap. The GoMap technology uses a stationary tag, sensed by a mobile phone, to bring up a web browser on the phone viewing the mobile map produced by FMWEB. The system allows the user to specify a path, and since the tag’s location is fixed, it is initially used as the starting point for the path. The functions supported include: (1) zoom and pan of the map; (2) shifting to related maps such as campus map; (3) selecting an object on the current map as the path start or end; (4) stepping along the path; (5) turning overlays of objects on or off; and (6) lookup of a map object by name in the current map, a currently displayed overlay, or the enterprise map suite. Figure 6

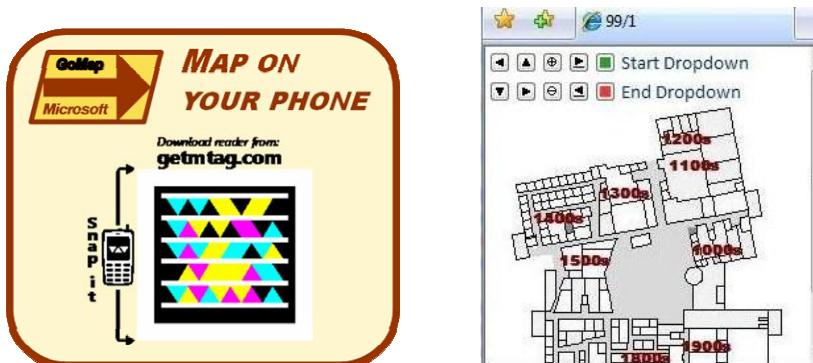


Fig. 6. The GoMap demo allows a mobile phone user to snap a photo of a Microsoft Tag to bring up a building map in the phone’s web browser. (The screen shot here is from a desktop browser for ease of production.)

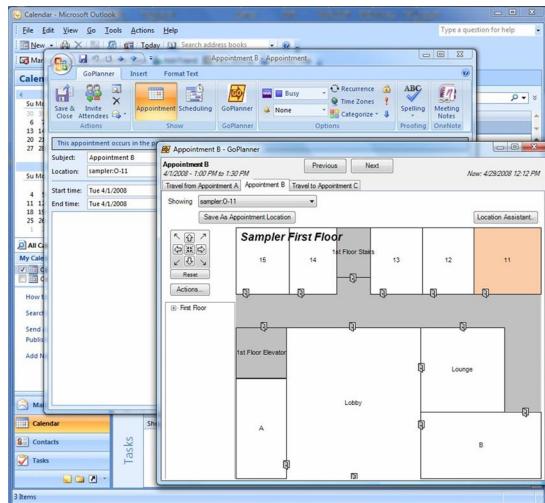


Fig. 7. The GoPlanner aids in meeting location selection, in travel planning, and in map generation for wayfinding

depicts the Microsoft Tag [15] that the phone senses in the current demo (NFC or other technologies could be used), and the resulting map. The current GoMap is browser-based; we are developing a native phone application as well, that will fetch maps to the phone, performing all computations on the phone itself.

GoPlanner. The GoPlanner (Figure 7) is an add-in for Microsoft Outlook that aids in selecting a meeting or event location, as well as displaying the map for travel between events. The functions are being expanded to include planning for a day's travel and for selecting an available meeting room given numerous constraints on the attendees' calendars and travel (even looking for a room in a nearby building on campus if no rooms are available in the attendees' buildings).

Hospital. In conjunction with a developer of hospital information systems, we are prototyping a number of location-aware applications using the Facility Map Framework. We have a demo now in which the nurse or administrator in the Emergency Department can begin with a spreadsheet-like display of patient information, including medical diagnosis and treatment (along with other data) – then, by clicking a “Map” button, the view shifts to a map drawing of the Emergency Department showing all patient beds, along with key information about each patient. Future applications include locating equipment and staff through the map, and planning patient travel paths to minimize risk of infection. We have contracted with an IWMS partner to create an FMF model of the entire hospital.

8 Status and Future Work

The Facility Map Framework is a work in progress. The current design is the result of years of experimentation with various engineering approaches to the subtasks and

functions that comprise the system. But, the current approaches are not always adequate in all dimensions. The capabilities described above all are implemented and working in research demonstrations, but some key subsystems have been deferred for future investigation and development:

1. Real-time map update, and tracking and event subscriptions
2. Server-side queries and loading of partial Facility Maps
3. Map alignment and overlay authoring tools

When these capabilities are added, FMF will be a full-featured infrastructure for private space location systems. Beyond that point, attention can be given to alignment with other standards and industry initiatives, to create a marketplace for location-aware ubiquitous computing technology components in private spaces.

Acknowledgments. Special thanks to Bill Holt for much discussion and labor in support of FMF and its predecessor systems, to Donald Barnes for directing attention to many practical issues arising in the creation and use of facility maps, and to Renato Cazangi for collaboration in developing demonstrations. Thanks also to the reviewers for very helpful suggestions and comments.

References

1. Jiang, C., Steenkiste, P.: A Hybrid Location Model With a Computable Location Identifier for Ubiquitous Computing. In: Borriello, G., Holmquist, L.E. (eds.) UbiComp 2002. LNCS, vol. 2498, pp. 246–263. Springer, Heidelberg (2002)
2. Stahl, C., Haupert, J.: Taking Location Modelling to New Levels: A Map Modelling Toolkit for Intelligent Environments. In: Hazas, M., Krumm, J., Strang, T. (eds.) LoCA 2006. LNCS, vol. 3987, pp. 74–85. Springer, Heidelberg (2006)
3. Building Information Spatial Data Model committee, <http://bisdm.org>
4. The Green Building XML Schema gbXML, <http://www.gbxml.org>
5. Niu, W., Kay, J.: Location Conflict Resolution with an Ontology. In: Indulska, J., Patterson, D.J., Rodden, T., Ott, M. (eds.) PERVASIVE 2008. LNCS, vol. 5013, pp. 162–179. Springer, Heidelberg (2008)
6. Filho, J.O., Bunoza, A., Sommer, J., Rosenstiel, W.: Self-Localization in a Low Cost Bluetooth Environment. In: Sandnes, F.E., Zhang, Y., Rong, C., Yang, L.T., Ma, J. (eds.) UIC 2008. LNCS, vol. 5061, pp. 258–270. Springer, Heidelberg (2008)
7. Casas, R., Marco, A., Falco, J.L., Gracia, H., Artigas, J.I.: DALMA – Location Aware Alarm System for People with Disabilities. In: Miesenberger, K., Klaus, J., Zagler, W.L., Karshmer, A.I. (eds.) ICCHP 2006. LNCS, vol. 4061, pp. 744–751. Springer, Heidelberg (2006)
8. Papataxiarhis, V., Riga, V., Nomikos, V., Sekkas, O., Kolomvatsos, K., Tsetsos, V., Papa-georgas, P., Vourakis, S., Xouris, V., Hadjiefthymiades, S., Kouroupetroglou, G.: MNISIKLIS: Indoor Location Based Services for All. In: Gartner, G., Rehrl, K. (eds.) Location Based Services and TeleCartography II From Sensor Fusion to Context Models. Lecture Notes in Geoinformation and Cartography, pp. 26–282. Springer, Heidelberg (2009)
9. Prototype-based Programming at Wikipedia, Wikimedia Foundation, http://en.wikipedia.org/wiki/Prototype-based_programming

10. Simple Object Access Protocol, <http://www.w3.org/TR/soap/>
11. Hile, H., Borriello, G.: Information Overlay for Camera Phones in Indoor Environments. In: Hightower, J., Schiele, B., Strang, T. (eds.) LoCA 2007. LNCS, vol. 4718, pp. 68–85. Springer, Heidelberg (2007)
12. Global Location Numbers (GLN), GS1, http://www.gs1.org/docs/idkeys/GS1_Global_Location_Numbers.pdf
13. Unified Approach for Measuring Office Space; For Use in Facility and Property Management. Building Owners and Managers Association (BOMA) International, <http://www.boma.org>
14. Industrial Floor Measurement Standard 2004, Building Owners and Managers Association (BOMA) International (2004), <http://www.boma.org>
15. Microsoft Tag, <http://www.microsoft.com/tag/>

Daily Routine Recognition through Activity Spotting

Ulf Blanke and Bernt Schiele

Computer Science Department, TU Darmstadt, Germany
`blanke@cs.tu-darmstadt.de, schiele@cs.tu-darmstadt.de`

Abstract. This paper explores the possibility of using low-level activity spotting for daily routine recognition. Using occurrence statistics of low-level activities and simple classifiers based on their statistics allows to train a discriminative classifier for daily routine activities such as working and commuting. Using a recently published data set we find that the number of required low-level activities is surprisingly low, thus, enabling efficient algorithms for daily routine recognition through low-level activity spotting. More specifically we employ the JointBoosting-framework using low-level activity spotters as weak classifiers. By using certain low-level activities as support, we achieve an overall recall rate of over 90% and precision rate of over 88%. Tuning down the weak classifiers using only 2.61% of the original data still yields recall and precision rates of 80% and 83%.

Keywords: Activity Recognition, Wearable Computing, Human Routines.

1 Introduction

Human activity is an important ingredient for context-aware systems. Its recognition has gained a lot of interest recently for diverse domains spanning from industrial applications to modeling of human behaviour in medical care. The different requirements of these domains have led to a multitude of approaches targeting recognition on different scales and complexities ranging from gesture recognition (happening within seconds) to complex daily routines (lasting often for hours and consisting of multiple activities).

Many state-of-the-art activity recognition approaches use and model all sensor data generated during the user's activity. For example a Hidden Markov Model can be used to model the sensor data of a particular activity. However, modeling and recognizing highly variable and in particular long-lasting activities such as daily routines (e.g., lunch or commuting activities) using such holistic approaches might be suboptimal and even infeasible in the presence of limited training data. In contrast, *activity spotting* aims to identify those activities that are most distinctive for a particular long-term activity. Identifying those distinctive parts of an activity and spotting them for recognition has at least two major advantages. First, the computational requirements can be reduced significantly

and second, even limited amounts of training data can suffice to obtain good recognition performance.

Another predominant aspect of most recent activity recognition approaches is that the recognition of complex (high-level) activities follows a bottom up approach. The classification of low-level activities is used as a basis to model and infer more complex activities. This work takes rather the perspective of a top down approach. Having the annotation of high level routines, the aim is to identify low level activities, which have a high descriptive power to distinguish the routines. For this we use a feature selection algorithm to reveal the low level activities which matter most for classifying a routine.

The main contribution of this paper is to investigate the feasibility of spotting distinctive low-level activities to recognize complex activities, in our case daily routine activities. We achieve this by an automatic discriminative analysis of the routines and spotting the discriminative part of the routine's data. Experimental results show that far less data is needed than one might intuitively expect and that only a small subset of lowlevel activities suffice to support the classification of daily routines. As large portions of the data are not used and therefore irrelevant for activity classification, the approach is highly efficient. As a result the approach lends itself to computationally inexpensive embedded activity recognition.

The paper is organized as follows. First we situate our work within related work (section 2). After presenting our approach in detail (section 3), we introduce a publicly available dataset (section 4), on which the evaluation is done (section 5). We complete this paper with a conclusion (section 6).

2 Related Work

In human activity recognition, many different methods have been proposed focusing on different types, scales and complexities of activities. However, surprisingly little literature exists on long-term and high-level activity recognition and therefore remains an open research challenge. The following summarizes the most related work even though we are not aware of any prior work to use low-level activity spotting to recognize complex or high-level activities.

2.1 Layered Activity Recognition

A popular approach to activity recognition is a layered inference of complex activities based on prior classification of simple activities as subcomponents. In [7] the authors show that Hidden Markov Models (HMM) have the ability to capture different levels of abstraction with respect to time granularity. Constructing a cascade of HMMs, low-level video and audio data is used for a first HMM and its output is fed into a second set of HMMs. The evaluation is done on office activities, like *attending a presentation* or *making a phone call*.

In the work of Dong Zhang et al. [11] a two-level classifier is used on auditory and visual data to recognize human interactions patterns in meetings. In the first

layer individual actions are recognized such as *speaking*, *writing*, and *idle*. The output of these models is then used as input for the second layer, which models group interaction such as *presentation*, *discussion*, *monologue*, and *white-board*.

Clarkson and Pentland [1] propose a method to discover *events* and *scenes* on ambulatory audio and video data. The authors find that clustering with regular HMMs only separates specific events (e.g., cashier beeps, supermarket music, walking through aisles), but does not capture the fact, that these events occur together in a scene. Using a hierarchy of HMMs, capturing first the lowlevel events, scenes such as being at the supermarket, at a busy street or in the video store can be discovered.

In [5], an approach is used to separate classes such as walking, jogging, or driving a car on sample level discriminatively. The posterior probabilities of the classification are then fed into an HMM, which captures the temporal context and hereby improving the results of the discriminative inference.

2.2 Activity Spotting

Following the paradigm of activity spotting, the aim is to find subcomponents of activities that allow to distinguish different classes. The authors of [6] approach the discovery of activities by identifying motifs in multivariate time series. In their context, motifs mean reoccurring subsequences, which have a high intra-motif similarity and can be distinguished from other subsequences. They use unsupervised techniques to identify these motifs. To evaluate their approach, they record a set of different dumbbell exercises.

In [12], gesture recognition on continuous streams is approached by identifying segments of interest. These simple segments are then combined as subparts of a certain gesture.

2.3 Longterm Activity Recognition

Research on long-term activity recognition is still in its infancy. Besides the difficulty of the task, also practical reasons have hindered progress: recording of long-term data is a non trivial task and annotating the data tends to be cumbersome and time consuming. In [4], a probabilistic model (originated from text-document analysis) is used to discover daily routines. By discretizing low level sensor data into a 'word'-type of representation, routines can be interpreted as 'documents'. Using topic models, the authors can discover daily routines such as going for lunch or commuting.

Others investigate the recognition of activities on a larger timescale such as shopping or doing housework without having low level activities as a prerequisite and using standard algorithms to infer labeled activities on larger timescales [3]. Van Laerhoven et al. [9] use a model of the user's rhythms to improve the recognition on a dataset of 27 days and compare their approach to the exclusive use of sensor data, such as motion, light and temperature. The model is constructed by a daily probability distribution of the user's annotated activities.

3 Activity Spotting for Daily Routine Recognition

The main goal of this work is to use low-level activity spotting to recognize high-level activities such as daily routines. Given a particular set of such high-level activities we therefore need to identify parts of the sensor data that enable reliable discrimination of the activities. For this we employ a boosting framework using low-level activity spotters as weak classifiers.

In this paper we use the occurrence statistics of low-level sensor data as the basic representation of sensor data. This representation closely follows the representation previously used for daily routine modeling and recognition [4]. We first perform unsupervised clustering of the low-level sensor data. For a given time-window we then compute the occurrence statistics of the cluster centers. This is the basic representation of the sensor and is used as input for the boosting framework. Figure 1 illustrates the steps in more detail. After extracting the features (1), distances to the cluster centers are computed (2) and the occurrence statistics are calculated (3). The result of boosting is a set of scores for the activity classes (4).

The following describes the major components in more detail. The next section describes the feature calculation (according to step (2) and (3) from Fig. 1). Then we introduce the feature selection method based on JointBoosting and describe how it is being used to automatically discover discriminative low-level activities.

3.1 Extracting Low Level Events

Given activity data in form of acceleration data, we first cluster its features using K-means clustering. The obtained cluster centers are then used to assign each data sample the n nearest cluster centers. In the experiments below we use $n = 1$

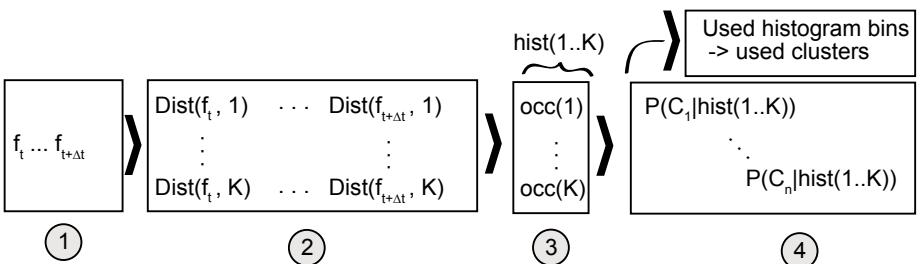


Fig. 1. (1) Feature extraction over a sliding window of raw sensor data. (2) All data points are (softly) assigned to cluster centers. (3) Occurrence statistics of the cluster centers are computed for the sliding window and stored in a histogram. (4) Each histogram is classified using JointBoosting resulting in scores, i.e., posterior probabilities for each activity. Note that boosting selects the most discriminative cluster centers (= histogram bins) for classification, thereby reducing the required sensor data substantially.

for hard assignments as well as $n > 1$ for soft assignments. In this work we set $n = 3$. For soft assignment we use the standard softmax function:

$$w_i = \frac{\exp\{-\frac{d_i}{\sigma}\}}{\sum_{j=1}^K \exp\{-\frac{d_j}{\sigma}\}} \quad (1)$$

with d_i being the distance to the cluster and σ the standard deviation of all distances.

In the second step we calculate the occurrence statistics, by summing the soft assignment over a certain window length. This way we include the duration of a specific activity. Alternatively, to get the distribution of hard assigned clusters, we count the occurrence of each assigned cluster within the window. As time has proven to be a powerful cue for daily routine recognition, we added a time-of-day timestamp to the feature vector.

We use occurrence statistics and timestamps as input for the JointBoosting. v_0 corresponds to the timestamp and all other entries v_i correspond to one bin of the occurrence statistics, containing the sum of the weighted assignments of cluster center i within a window. Hence, the size of the feature dimension $|v|$ corresponds to the number of centers K plus one dimension for the timestamp. These are then labeled with the highlevel daily routines.

3.2 JointBoosting

In general, boosting [2] is used to combine a pool of weak classifiers to form a single strong classifier. The mechanism of boosting is interesting in many ways. Besides the classification improvement compared to individual weak classifiers, boosting can be used to find the most discriminant features [10]. In this work we employ regression stumps of the following form as weak classifiers:

$$h_m(v) = a \cdot \delta(v^f > \theta) + b \cdot \delta(v^f \leq \theta) \quad (2)$$

given a feature vector $\{v^f : f = 1, \dots, N_{features}\}$. θ is the optimal threshold being automatically found, so that h_m is positive if $v^f > \theta$ or negative if $v^f \leq \theta$. The Kronecker- δ results in 1 or 0, depending on the condition being true or false. Weak classifiers are combined additively to a strong predictor as defined:

$$H(v) = \sum_{m=1}^M h_m(v) \quad (3)$$

The number of weak classifiers M is also referred to as *rounds*.

The regression stump parameters

$$a = \frac{\sum_i w_i z_i \delta(v_i^f > \theta)}{\sum_i w_i \delta(v_i^f > \theta)} \quad (4)$$

$$b = \frac{\sum_i w_i z_i \delta(v_i^f \leq \theta)}{\sum_i w_i \delta(v_i^f \leq \theta)} \quad (5)$$

are computed from the weighted square error of the training data and can be seen as a weak classifier voting for or against a class. w_i denotes the weight for each training sample. In each round these weights are updated and increased for samples which are misclassified and decreased for samples which are correctly classified. This makes the training focus on harder training samples in future rounds. $z_i = \{+1, -1\}$ denotes the binary class membership label. Intuitively, a is the confidence in judging a sample positively, if the feature is greater than θ . b is the confidence this sample not being part of the class.

To analyze how well a weak classifier separates one class from the remaining classes, we take the difference of the regression stump parameters a and b :

$$vote_{confidence} = sign(a) \cdot |(a - b)| \quad (6)$$

The sign of a indicates whether this classifier votes for (when positive) or against (when negative) a specific class.

In [8], Boosting is extended by the ability to share features across different classes. The basic idea is to not only to separate between two classes but to find subsets of classes (for each weak classifier) that are best separated. This increases the computational cost during training and typically greedy a search is used to reduce training times [8]. During testing however, this extension allows to reduce the computational costs significantly as weak classifiers are shared across classes.

Besides the computational advantage, JointBoosting also allows the analysis which low-level activities can be used jointly to discriminate between multiple high-level activities. How the parameters of the JointBoosting can be interpreted to allow such an analysis is described in section 5.2.

4 Dataset

To test our approach of selecting the most significant low level subcomponents of an activity, we evaluate it on a realistic and representative dataset of activities and routines of daily living. We briefly describe the dataset, and refer to [4] for further details on the recording.

Two 3-axis-acceleration sensors, sampling at a rate of 100Hz are worn, one at the wrist and one in the right pocket. As features, mean and variance are calculated over a window of 0.4s, resulting in approximately 2.5Hz. The dataset contains 7 days of continuous data leaving out the sleeping phases. Table 1

Table 1. Daily routines observed over seven days

Routine	Duration
Dinner	217.5 min
Commuting	289.0 min
Lunch	391.3 min
Office Work	2814.7 min

Table 2. Low level activities occuring during the routines

Activity	Average duration	Occurrences	Total
sitting / desk activities	49.41 min	54	3016.0 min
unlabeled	1.35 min	239	931.3 min
having dinner	17.62 min	6	125.3 min
walking freely	2.86 min	38	124.2 min
driving car	10.37 min	10	120.3 min
having lunch	10.95 min	7	75.1 min
discussing at whiteboard	12.80 min	5	62.7 min
attending a presentation	48.9 min	1	48.9 min
driving bike	11.82 min	4	46.3 min
walking while carrying something	1.43 min	10	23.1 min
walking	2.71 min	7	23.0 min
picking up mensa food	3.30 min	7	22.6 min
sitting / having a coffee	5.56 min	4	21.8 min
queuing in line	2.89 min	7	19.8 min
using the toilet	1.95 min	2	16.7 min
washing dishes	3.37 min	3	12.8 min
standing / having a coffee	6.7 min	1	6.7 min
preparing food	4.6 min	1	4.6 min
washing hands	0.32 min	3	2.2 min
running	1.0 min	1	1.0 min
wiping the whiteboard	0.8 min	1	0.8 min

shows the four annotated daily routines. These contain different types of low-level activities. The routine *dinner* for instance groups low-level activities such as *preparing food*, *having dinner (eating)*, and *washing dishes*. In total the data set contains 34 labeled low-level activities of which a subset of 24 activities occurred during the routines. A complete list of the low level activities occurring during these routines is provided in Table 2. The total duration of each activity is given as well as the mean duration for each instant of one activity.

5 Experimental Results

This section reports on experimental results for the dataset introduced in section 4. We begin with a quantitative analysis of the algorithm's performance and compare the results with [4] (section 5.1). Then we discuss in more detail the classifiers obtained with JointBoosting (section 5.2). In particular we give an interpretation which part of the low-level activities are characteristic and therefore chosen by JointBoosting to recognize daily routines.

5.1 Quantitative Results

To make our results comparable to published results on the same dataset we aimed to follow the original parameter settings as closely as possible [4]. Therefore we used the following parameters: for K-means clustering we set K to 60

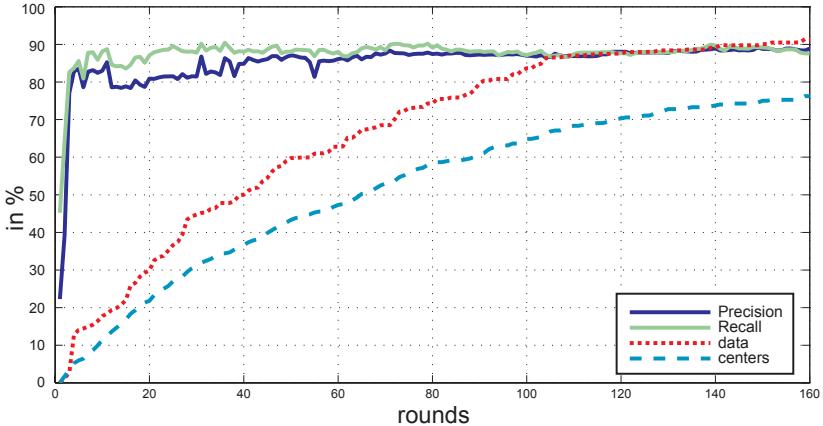


Fig. 2. Overall recognition results, amount of used data using different numbers of JointBoosting rounds and soft assignments of K-means clusters. *Centers* denotes the percentage of selected features of all given features (= K-mean-centers) and *data*, the percentage of their assignments in the data.

and we calculated histograms over a window of 30 min with an overlap of 5 min. As mentioned previously we also added a dimension for the time-stamp resulting in a feature vector with dimension 61. We evaluated the approach on a seven-fold crossvalidation, training a classifier on 6 days and leaving one day out for testing.

Overall Performance. The overall recognition results for different numbers of rounds are given in Fig. 2. The highest precision and recall rates are 87.81% and 90.17% at 80 rounds. Using more rounds results in a slight decrease in recall and a small increase in precision. Table 3 shows several results in more detail.

Using 4 rounds, the classifier yields a precision of 82.85% and a recall of 83.67% by using 12.78% of the given data. Applying boosting with 10 rounds increases the recall by about 5%, but has no noticeable effect on precision. Increasing the rounds to 80 improves the precision about 5% and recall about 6.5%. Here more than half of the clusters (57.93%) and nearly 3/4 of the data (74.30%) are used.

Table 3 also shows a comparison of our results to the probabilistic approach given in [4]. It can be seen that our discriminant approach yields clearly better results. What is quite surprising though is that already using but four boosting rounds (i.e. four low-level activity spotter) allows to achieve a clear gain in overall performance. Table 4 gives a more detailed comparison by not only showing the overall performance but also showing the performance for each routine separately. Here we compare the results of [4] with the results obtained with 4 and 80 boosting rounds. Using 80 rounds always outperforms the approach of [4] with one exception: precision for *commuting*. Again it is surprising that using 4 boosting rounds outperforms the approach of [4] for three out of four daily routine activities and only obtains lower precision and recall for *commuting*. This clearly

Table 3. Overall recognition results for different numbers (4, 10, 20, 80 and 160) of rounds of the JointBoosting algorithm using soft- (top) and hard-assigments (bottom) to the K-means cluster centers

Soft Assigments to K-means centers						
Rounds	4	10	20	80	160	Huyhn et al [4]
Precision	82.85%	82.98%	80.91%	87.81%	88.87%	76.90%
Recall	83.67%	88.12%	87.09%	90.17%	87.16%	65.80%
Used Centers	5.20%	11.39%	15.70%	57.93%	76.25%	-
Used Data	12.78%	17.74%	21.81%	74.30%	91.41%	-

Hard Assigments to K-means centers					
Rounds	4	10	20	80	160
Precision	72.71%	77.34%	82.19%	86.40%	88.37%
Recall	82.67%	82.39%	87.09%	90.32%	89.18%
Used Centers	5.19%	12.14%	22.26%	50.82%	56.13%
Used Data	2.11%	4.94%	14.27%	45.42%	49.75%

Table 4. Results per Routine using soft assignment of K-means cluster centers

Routine	4 rounds		80 rounds		Huyhn et al [4]	
	Precision	Recall	Precision	Recall	Precision	Recall
Dinner	84.31%	100.0%	85.27%	90.48%	56.90%	40.20%
Commuting	70.04%	60.27%	81.77%	82.36%	83.50%	71.10%
Lunch	78.85%	81.79%	84.56%	90.04%	73.80%	70.20%
Office Work	97.86%	92.61%	98.12%	93.63%	93.40%	81.80%

shows the applicability of the proposed approach of low-level activity spotting for daily routine recognition.

Fig. 3 illustrates the classification on one day of the seven day dataset using the same number of boosting rounds as in Table 3. It can be seen that the borders of two routines are seldomly precise. For instance, using 10 boosting rounds the *lunch* routine is predicted before it actually happens (which is reflected in a lower precision). However, the transitions from one routine to another happens smoothly, that is, there is often no exact start and end point. As the obtained classifiers also show the same smooth transitions between the routines we think that the results are indeed sufficient for many applications.

So far we reported results for soft assignments of the clusters. As hard assignments have the potential to reduce the amount of data that needs to be considered for classification we also evaluated hard assignments of the K-means centers(, i.e., each sample is assigned to exactly one cluster). The overall results using 1 to 160 rounds are depicted in Fig 4.

We note the maximum at 40 rounds with 87% precision and 93% recall using 30% of the data and 37% of the centers. Table 3 shows the results in more detail. At 4 rounds we obtain a precision of 72.71% and a recall of 82.67%. This is a decline of 15% of precision and about 11% recall compared to the maximum.

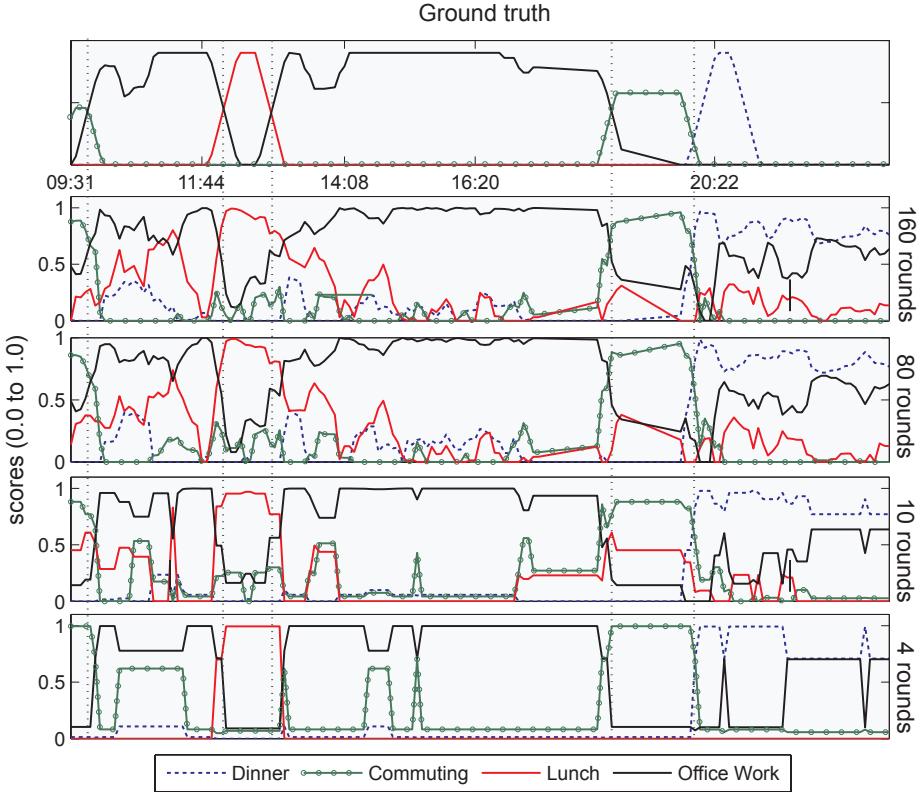


Fig. 3. Top: Ground truth for one day. Below the classification by the JointBoosting algorithm using 160, 80, 10 and 4 rounds.

However, the usage of data with 2.11% is minimal and considerably lower than for soft assignments. Adding one round, already improves the result to a precision of 79.75% and a recall of 83.40% using less than 1% more data (2.6%). This is due to the fact that only the major vote, that is the winning K-means centers can be observed even if some others have almost the same distance. Preserving the information of distances to a few centers using a soft assignment seems to be beneficial for lower number of classifiers.

We also have started to experiment with smaller window sizes. E.g. using a window length of 15 min results in a reduced precision (for 4 rounds) of 71.06% but with only a small decrease in recall (80.07%). Such a decrease in performance is in agreement with previous results [4]. As this is an important and interesting direction to pursue we plan to investigate this further.

5.2 JointBoosting Classifier Discussion and Visualization

Moving from activity recognition into the paradigm of activity spotting our goal is to analyze which and how much data needs to be observed to reliably recognize

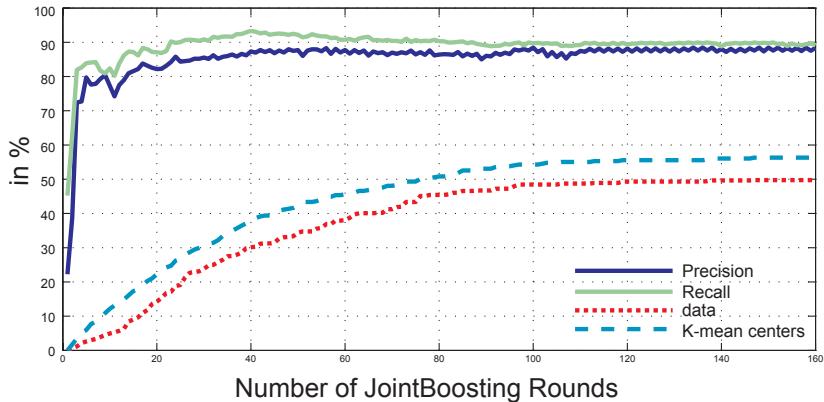


Fig. 4. Overall recognition results and amount of used data using different numbers of JointBoosting rounds and hard assignments of K-means clusters. *K-mean centers* denotes the percentage of selected features of all given features (= K-mean-centers) and *data*, the percentage of their assignments in the data.

daily routines such as the ones used above. Applying JointBoosting, we get a selection of the most important features to discriminate the given classes. As JointBoosting selects the most important (= discriminative) occurrences of a specific cluster, we can identify which parts of low-level data is important for the daily routine classification task.

To identify the amount of data used for classification, we count how often these clusters occur during the total length of the data. For soft assignments we look if the selected clusters appear in the top 3 assignments. While soft-assignment increases the amount of data used it also has the potential to reduce the number of weak classifiers required for good performance.

For a better understanding which data is selected we use the provided low-level activity annotation and assign to each cluster a set of low-level activity labels. In the following we use the distribution of the top five assigned labels to each cluster center.

Fig. 5 visualizes the first ten weak classifiers chosen by boosting. The color indicates the confidence of the weak classifier, predicting the sample to be from a specific class. The corresponding thresholds are given inside the boxes. Positive weights are colored green, respectively marked with a plus-symbol, negative weights are colored red and marked with a minus-symbol. Intuitively the reader can interpret the color as a voting for a specific routine when colored green or against it when colored red. The intensity of the color represents the absolute difference between the regression stump parameters a and b , given in equation 6 (Section 3.2).

Starting with round one from the top (one weak classifier), it separates *dinner* from the rest using time as feature. In the second round, the weak classifier separates *lunch* from the rest by observing the occurrence of cluster 36 (= feature 36). This cluster corresponds mostly to *walking freely* which turns out to be

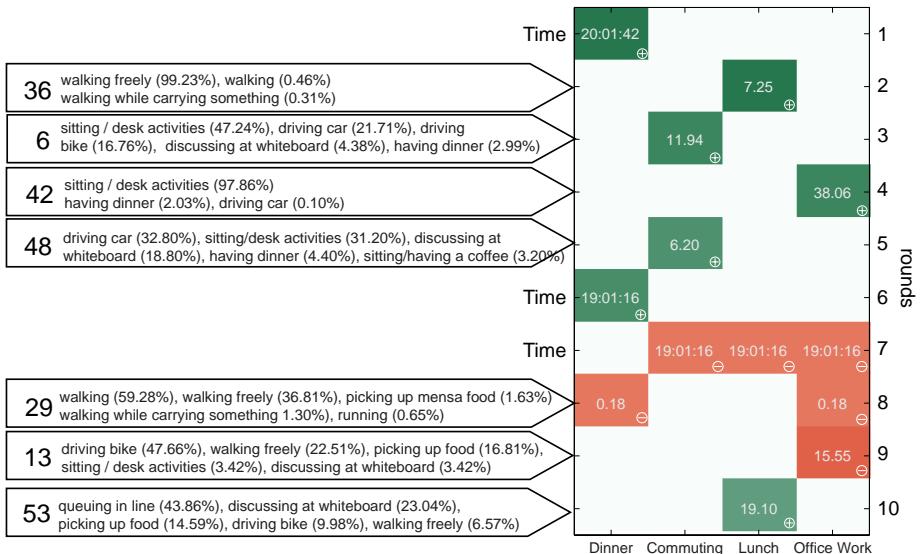


Fig. 5. The used features (occurrences of K-means centers and time) per round and per class, starting from the top with the first round. For each K-means center the label distribution of low-level activities are denoted. The threshold of the weak classifier is given in the colored boxes. Green (/plus) marked classifiers indicate a positive voting for the routine, red (/minus) a negative voting.

more discriminant than other low-level activities such as *having lunch*. This can be easily explained by the fact that *having lunch* is too similar to other low-level activities such as *having dinner* or *sitting/desk activities* and is therefore not chosen by boosting. The third classifier separates *commuting* from the other classes using cluster 6. This cluster is dominated by activities occurring during commuting: *sitting/desk activities*, *driving car* and *driving bike*. In the forth round cluster 42 is chosen to classify routine *office work* which mostly contains *sitting/desk activities*. Note that the threshold shows a comparatively high value, which means the activities of this cluster have to appear often.

In the eighth round cluster 29 is used to separate *dinner* and *office work* from *commuting* and *lunch*. More specifically the classifier votes against *dinner* and *office work*, if the given data sample is bigger than the threshold. The threshold of 0.18 indicates that the occurrence of this activity is fairly low, but enough to discriminate between classes. For each feature, i.e. the cluster center the top five low level-activities (Table 2) are given to get an intuition why the cluster is used. It can be seen that cluster 29 shares the labels *walking*, *walking freely*, *picking up mensa food* etc. Due to the fact that these activities only happen during *lunch* and *commuting* it is a good cue that *dinner* and *office work* are not happening.

An interesting observation is that the first 4 rounds separate each routine without sharing any features. Time turns out to be sufficient as feature to separate *dinner* from the others. After 8:00pm the confidence that *dinner* is the

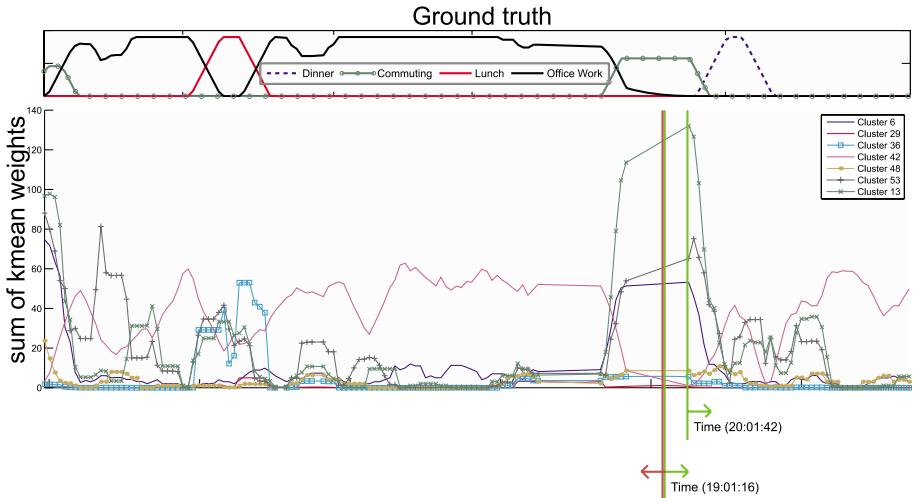


Fig. 6. The top feature occurrences during one day. (Top) The groundtruth for one day. (Bottom) The sum of weights of the soft assigned kmean-centers for each sample. The ranking of the selected features is as follows: 20:01:42 (time), 36, 6, 42, 48, 19:01:16 (time), 29, 13, 53.

current routine is very high. On the other hand, looking at round 7, *commuting*, *lunch* and *office work* is less likely to happen after 7pm and added as another weak classifier.

We did not compare JointBoosting with AdaBoost quantitatively. However, looking at the first 10 rounds depicted in Fig. 5, note that fewer features are shared as expected. Given this dataset, jointBoosting ranks those features first, which are discriminative enough for each class individually. This leads to the assumption that jointBoosting might yield similar results compared to AdaBoost for a low number of weak classifiers. However we observe at higher rounds that more features are shared. E.g., in rounds 10 to 20 seven features are jointly used, which is not possible with AdaBoost. JointBoosting therefore leads to a computationally more efficient solution.

Low Level Activities. Figure 6 illustrates the occurrences over time of the selected features in the tenth round. We encourage the reader to use Fig. 5 as reference to lookup the low level activities per cluster. Given the routine *office work* one can observe that the occurrence of cluster 42, which is selected as a weak classifier for this routine, is most of the time above the threshold of 38.06 to vote for this routine. One can see that cluster 53 and 42, which classify *lunch* and *office work* happens also during the dinner routine. However none of those routines happen after 19:01:16, whereas dinner is likely to happen after 20:01:42 and the combination of each of these classifiers leads to a higher posterior probability of the routine *dinner*. Cluster 53 is used by a weak classifier voting for the *lunch*

routine. It is mainly assigned to the low level activity *queueing in line*. Although this cluster has a high occurrence during the *office work* in the morning before lunch, its confidence is weaker than the confidence of cluster representing sitting at the desk, yielding a higher score for *office work* than lunch.

Observing the assigned labels to the used clusters, it can be noticed that, e.g, for the routine *lunch* it is not the activity *having lunch* discriminant. But activities which surround the actual activity of this routine, *which are walking (freely), queueing in line and picking up cafeteria food*. The explanation is simple - as *having lunch* is virtually the same as *having dinner* or similar to *sitting at the desk* and can be easily confused. Whereas walking or standing in line at a certain time can be very discriminative.

5.3 Discussion

The previous sections showed that discriminative classification and spotting of low-level activities yields promising results when applied to complex and highly variable daily routines. The proposed approach achieves good overall results with recall and precision above 80% in general to recognize the presented daily routines. Using a small number of classifiers (that is a small number of JointBoosting rounds) already achieves good performance. It is important to note that by using small numbers of classifiers, it is possible to reduce the required data substantially. Using only 2.6% of the data (i.e., 5 rounds and hard assignment to K-mean centers), it is possible to distinguish between different routines with a recall of 79.75% and a precision of 83.40%. We can determine a tradeoff between used data and performance, which favors using less data, as performance decrease is marginal.

6 Conclusion

This paper investigates the possibilities of *activity spotting* to recognize complex activities, like daily routines consisting of a series of low level activities. A top-down perspective is taken using a feature selection algorithm and the annotation of those higher-level routines to automatically spot discriminant low level data, respectively activities.

Our findings show that the approach of *activity spotting* leads to viable results. It could be seen that a surprisingly small amount of data is discriminant enough to differentiate routines such as *lunch* or *commuting* and yield to results that are better than those of current research. Our algorithm discovers successfully meaningful low level activities, which can be intuitively connected to the high level routines.

Applying this approach we are able to filter insignificant data, which does not support recognition. Once the training is done offline, the most discriminant spots in the data are obtained, which the classifier has to process. This allows to reduce the computational costs in the classification task, while the distance to only a few K-means centers needs to be calculated. Thresholding the distance

filters the unimportant data. Additionally, we reduce the amount of data which has to be stored. Implementing such classification on embedded systems will greatly benefit from this reduced complexity.

Acknowledgments. This work has been supported by the DFG Graduate School Topology of Technology. The authors of this paper gratefully acknowledges Christian Wojek for providing them with a JointBoosting implementation.

References

1. Clarkson, B., Pentland, A.: Unsupervised clustering of ambulatory audio and video. In: ICASSP (1999)
2. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors). *Ann. Statist.* 28(2), 337–407 (2000)
3. Huynh, T., Blanke, U., Schiele, B.: Scalable recognition of daily activities with wearable sensors. In: Hightower, J., Schiele, B., Strang, T. (eds.) LoCA 2007. LNCS, vol. 4718, pp. 50–67. Springer, Heidelberg (2007)
4. Huynh, T., Fritz, M., Schiele, B.: Discovery of activity patterns using topic models. In: Proceedings of the 10th International Conference on Ubiquitous Computing (Ubicomp) (2008)
5. Lester, J., Choudhury, T., Kern, N., Borriello, G., Hannaford, B.: A hybrid discriminative/generative approach for modeling human activities. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI) (2005)
6. Minnen, D., Starner, T., Essa, I., Isbell, C.: Discovering characteristic actions from on-body sensor data. In: Proceedings of the 10th IEEE International Symposium on Wearable Computers (ISWC) (2006)
7. Oliver, N., Horvitz, E., Garg, A.: Layered representations for human activity recognition. In: Proceedings of the 4th IEEE International Conference on Multimodal Interfaces (2002)
8. Torralba, A., Murphy, K., Freeman, W.: Sharing visual features for multiclass and multiview object detection. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2004)
9. Van Laerhoven, K., Kilian, D., Schiele, B.: Using rhythm awareness in long-term activity recognition. In: Proceedings of the 12th IEEE International Symposium on Wearable Computers (ISWC) (2008)
10. Viola, P., Viola, P., Jones, M., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Computer Vision and Pattern Recognition (CVPR) (2001)
11. Zhang, D., Gatica-Perez, D., Bengio, S., McCowan, I., Lathoud, G.: Modelling individual and group actions in meetings: a two-layer hmm framework. In: Computer Vision and Pattern Recognition Workshop (2004)
12. Zinnen, A., Schiele, B.: A new Approach to Enable Gesture Recognition in Continuous Data Streams. In: Proceedings of the 12th IEEE International Symposium on Wearable Computers (2008)

SofTOA: Software Ranging for TOA-Based Positioning of WLAN Terminals

Marc Ciurana, David López, and Francisco Barceló-Arroyo

Department of Telematic Engineering

Universitat Politècnica de Catalunya

Jordi Girona 1-3, 08034 Barcelona, Spain

{mciurana, david.lopez, barcelo}@entel.upc.edu

Abstract. TOA-based trilateration constitutes an interesting choice to locate terminals employing WLAN networks. A major limitation of this technique is the requirement for hardware modifications in the WLAN device in order to achieve accurate ranging. This paper presents an approach that overcomes this limitation. It is based on RTT measurements performed through time-stamping the transmission and reception of IEEE 802.11 MAC frames from the WLAN driver's code, employing the CPU clock as time-base. Some statistical processing is needed in order to mitigate the noise of the measurements. This paper presents experimental results obtained with a prototype showing ranging errors of a few meters when applied to estimate distances up to 25 meters in both indoor and outdoor environments.

Keywords: IEEE 802.11, WLAN location, positioning, ranging, RTT, Time of Arrival, TOA, tracking, trilateration, WLAN.

1 Introduction

1.1 TOA-Based Positioning in WLAN

Using deployed wireless communications infrastructures (e.g. IEEE 802.11, UWB, Bluetooth) for implementing location techniques offers advantages in terms of cost-efficiency, device-openness (i.e. non-proprietary devices compliant with the communication standard can be localized), service modularity and synergies (i.e. the infrastructure can also support communication services), while essential requirements for many location-based applications, such as good accuracy, availability and reliability, can likewise be reached. Location solutions employing IEEE 802.11 networks have become popular due to their deployment in many buildings, the maturity of the standard and the excellent relationship between coverage and needed infrastructure. In addition, current trends such as the inclusion of positioning capabilities in the upcoming IEEE 802.11v standard [1] and the increasing need for indoor locations show that location with WLAN is expected to continue playing a crucial role in positioning. Most of the existing WLAN-based location proposals are based on signal strength fingerprinting [2, 3, 4, 5]. They can achieve good positioning accuracy and are easily implementable, but the time-consuming offline training phase needed to construct the radio map database and the accuracy degradation

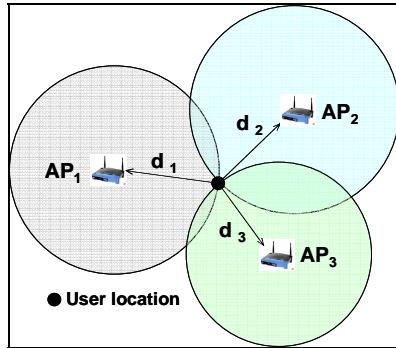


Fig. 1. Trilateration as intersection of circles

when environmental changes occur (e.g. furniture, people, temperature, engines, and so on) limit the suitability for certain applications. The alternative to overcome these limitations is trilateration-based positioning [6, 7]. With this method, the process of calculating the position of a mobile terminal (MT) can be summarized as follows: first, the distances between the MT and several reference points - Access Points (APs) in the case of WLAN - are calculated; then a trilateration [6] or tracking algorithm [7] takes as inputs the distance estimates and the known coordinates of the APs to calculate the MT's position. For 2D location, each geometrically obtained distance between the MT and an AP provides a circle centered at this AP (a sphere in 3D), on which the target device must lie. By using at least three APs, the MT's position in 2D can be obtained as shown in Fig. 1 [8]. In this figure d_1 , d_2 and d_3 represent the three necessary distances.

This paper deals with the key step of trilateration-based location methods for WLAN: the distance calculation – known as ranging – between the MT and an AP. The rest of the process to calculate the MT's position, i.e. the trilateration or tracking algorithm, is out of the scope of this work. The reader is directed to [6, 7, 9] for illustrative examples of contributions describing trilateration and tracking algorithms.

A simple method for WLAN ranging consists of measuring the received signal strength in the MT and relating the obtained value with the distance to the transmitter according to some attenuation pattern. However, this has been demonstrated to be impractical due to the high time-variability of the signal strength for a given distance and the lack of accurate signal propagation models, especially in an indoor environment [10]. A more stable alternative is to measure the Time of Arrival (TOA), which is the time that it takes the radio signal to travel from the transmitter (typically the MT to be located) to the receiver (typically an AP), and then calculate the distance by relying on the fact that electromagnetic waves propagate at constant speed (c):

$$d = c \cdot TOA . \quad (1)$$

While signal strength measurements are easily accessible from a standard WLAN interface, TOA measurements are not directly available, so new methods must be investigated for its provision as described below. However, TOA estimates have desirable properties – better accuracy and stability, less sensitive to the environment - that make it suitable for achieving high-performance ranging between nodes. Thus,

TOA-based trilateration constitutes a promising technique to reach flexible, robust and accurate location with WLAN. Several methods to estimate *TOA* in WLAN systems have been presented recently. Good accuracy has been reported when the hardware of the WLAN card is modified (i.e. hardware-based methods); however, methods that do not modify the hardware (i.e. software-based methods) are not accurate enough for most positioning applications.

1.2 Motivation and Objective

The goal of this contribution is to obtain a software-based solution for ranging with *TOA* in WLAN with precision close to the accuracy obtained by hardware-based methods. This study shows that this is feasible by taking the theoretical principles of existing *TOA* ranging proposals based on performing two-way ranging with frames of the IEEE 802.11 standard protocol, but designing a novel technique for its implementation from off-the-shelf WLAN terminals. The idea is to obtain the necessary metrics to calculate the *TOA* by taking the major benefits of the commercial MT's existing hardware and their programming interfaces. One of the main aspects of the pursued solution is that avoiding hardware modifications drastically reduces the implementation cost and complexity while easing the location system deployment.

2 Related Work

Two classes of approaches exist to obtain the *TOA* between two WLAN nodes: measuring the one-way trip time and measuring the Round Trip Time (*RTT*). In the former approach, the receiver determines the *TOA* based on its local clock, which is synchronized with the clock of the transmitter. The latter, also known as two-way ranging, measures the time that the signal spends traveling from a transmitter to a receiver and back to the transmitter again. This approach avoids the need for synchronization, which in general entails increases in complexity and cost. On the other hand, existing methods can also be classified depending on the layer at which the measurements are taken to estimate the *TOA*, as explained in the following section.

2.1 Estimating the TOA at the Physical Layer

Measuring the *TOA* at the physical layer in WLAN leads to very accurate distance estimates, but specific and complex hardware modules are needed, which makes the solution impractical for portable WLAN devices. One-way trip time ranging is usually employed. Most proposals are based on frequency-domain measurements of the channel response with super-resolution techniques, due to their suitability for improving the spectral efficiency of the measurement system. Some examples are the Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT), the Multiple Signal Classification (MUSIC) [11], the Matrix Pencil [12] or the more advanced Prony Algorithm [13]. Other methods are based on the correlation of the received IEEE 802.11 signal, such as [14], in which the received signal is correlated with a long-training symbol stored in the receiver and afterwards the channel frequency response is obtained to refine the initial *TOA* estimation.

2.2 Estimating the TOA at Upper Layers

The intention of measuring the *TOA* at upper layers of WLAN is to ease the feasibility of implementing ranging in portable WLAN devices, which takes advantage of the IEEE 802.11 standard protocol capabilities. The most common approach is performing two-way ranging (measuring the *RTT*) by employing exchanges of IEEE 802.11 MAC frames (e.g. Ready-to-send (RTS) - Clear-to-send (CTS), Data-Acknowledgement (ACK), or probe request - probe response) that can be induced in the MT. As an illustrative example, a *RTT* measurement with data-ACK can be expressed with the following components:

$$RTT = t_{tx_data} + 2 \cdot TOA + t_{proc_AP} + t_{rx_ACK} . \quad (2)$$

Fig. 2 illustrates the process. The propagation time of both frames (i.e. data and ACK) is assumed to be the same (i.e. the *TOA*); t_{tx_data} and t_{rx_ACK} are the transmission and reception times at the physical layer, respectively, and t_{proc_AP} is the time elapsed between the AP receives the data frame and the ACK is transmitted. This processing time in the AP is composed of the pure processing time in the AP, the Short Inter Frame Space (SIFS), and the time that the frame spends in the AP transmission queue (if the queue is not empty).

Once the *RTT* is estimated, the *TOA* can be extracted from Eq. (2) with the following expression:

$$TOA = \frac{RTT - t_{proc_AP} - t_{tx_data} - t_{rx_ACK}}{2} . \quad (3)$$

The *RTT* when the distance between MT and AP is zero is assumed not to contain propagation time, hence:

$$RTT_0 = t_{proc_AP} + t_{tx_data} + t_{rx_ACK} . \quad (4)$$

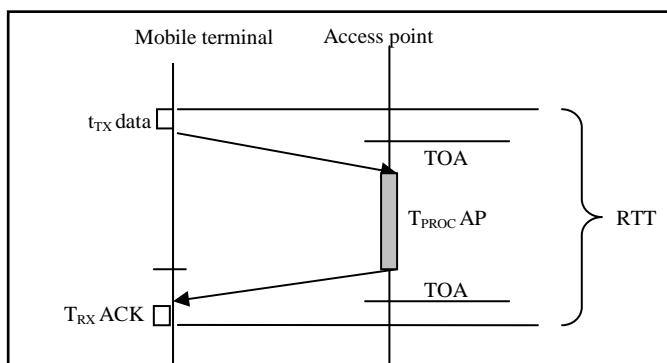


Fig. 2. Theoretical RTT measurement with IEEE 802.11 data-ACK frames

Then, combining Eq. (3) and (4), the *TOA* for a distance i different to zero can be obtained as:

$$TOA_i = \frac{RTT_i - RTT_0}{2} , \quad (5)$$

where RTT_i is the *RTT* for a distance i different to zero. In practice, RTT_0 is obtained by placing the MT and the AP together and then estimating the *RTT*. The upcoming IEEE 802.11v standard [1] is going to avoid this process because the processing time in the AP is calculated in real time and sent back to the MT.

It is important to accurately evaluate the *RTT* in the WLAN enabled node: since the signal propagates approximately at the speed of light, a time resolution of a few nanoseconds is needed to achieve accurate measurements. Currently neither the IEEE 802.11 standard nor the existing WLAN chipsets provide timestamps with the sufficient resolution in the frames; only a time base of 1 μ s corresponding to the temporal update of the IEEE 802.11 TSF (Timing Synchronization Function) can be accessed, which means 300 m in terms of distance. This is why most of the existing *TOA*-based ranging approaches for WLAN providing precise time measurements, even working at upper layers, entail modifications of the off-the-shelf WLAN chipsets (i.e. they are hardware-based solutions).

2.2.1 Hardware-Based Solutions

Most of these hardware-based solutions try to keep the hardware changes to a minimum, so that its implementation in practice is more feasible than in the case of the aforementioned methods working at the physical layer. The internal delay calibration both at transmitter and receiver is employed in [15], using the RTS-CTS frames exchange. In [16] an accurate time-stamp on transmission and reception is obtained by capturing a segment of the waveform and then performing a matched filter using the probe request – probe response exchange. In [9], the authors proposed connecting a counter module to the WLAN card and using the clock of the card as the time base for time-stamping the frame transmissions and receptions. The data-ACK frame exchange is employed; multiple *RTT* measurements are performed and merged over time to mitigate the impact of multi-path and enhance the time resolution. Ranging accuracy is close to 1 meter.

2.2.2 Software-Based Solutions

Software-only solutions have been seldom explored. A representative approach can be found in [17], in which *RTT* measurements are collected with time-stamps of 1 μ s using *tcpdump* and an additional monitoring node. The problem is that the average ranging error after some processing is around 8 m, which means noticeably worse accuracy than that in hardware-based approaches. As an illustrative example of the lack of maturity of *TOA*-based ranging for WLAN, an analysis of the feasibility of such ranging methods was provided in [18], and the conclusion was that it did not work due to the limitations of current hardware and protocols.

3 Description of the Ranging Technique

The ranging technique proposed in this work follows the theoretical idea of the two-way ranging approach explained above in Section 2.2. Thus, once the *RTT* between the MT and the AP has been estimated, the *TOA* can be calculated with Eq. (5) (it is assumed that the *RTT* at distance zero has been previously obtained and its value is stored in the MT). Then, the distance between both WLAN nodes is obtained with Eq. (1). A *RTT* measurement is performed in the MT with the IEEE 802.11 data-ACK frame exchange. The MT sends a unicast data frame to the AP and counts the time until the corresponding ACK answer from the AP is received. The employed frames belong to the MAC layer because this is the lowest network layer accessible from the MT via software; the lowest possible layer is preferred in order to avoid extra delays due to processing between network layers. Since the main targeted characteristic of the technique is accurately ranging with a pure software solution, the *RTT* is measured via a software mechanism that makes use of the existing hardware of an off-the-shelf MT. This is the innovative aspect of the method, which differentiates it from the rest of the existing proposals.

3.1 Mechanism for the RTT Measurement

The first key feature that is exploited is the optimum time resolution that the CPU clock of the MT can provide as a time-base to timestamp the transmission and reception of frames to measure the *RTT* (a resolution of 1 ns could be achieved with a 1 GHz CPU). This was briefly mentioned in [19]. The other mechanism that we take advantage of is that some current WLAN drivers can be modified and rebuilt so that the necessary events to obtain a *RTT* measurement (i.e. the transmission and reception of a MAC layer frame) can be accessible via software.

The proposed mechanism to obtain the *RTT* is time-stamping the instant when the MAC data frame is transferred to the physical layer for transmission and the instant when the ACK frame is received at the MAC layer; then both instants are subtracted. The difference between the time stamps is an estimate of the *RTT*. Since hardware-based mechanisms are intentionally avoided, the adopted solution is accessing the above-mentioned events from the code of the driver that controls the WLAN interface. In our case, this is achieved by modifying the open-source Madwifi driver for Atheros-based WLAN interfaces with a Linux operating system (OS). The driver's code implements the IEEE 802.11b/g MAC protocol. The timestamps are performed inside the *ath_rx_capture()* and *ath_tx_capture()* functions, which correspond to the frame's reception and transmission, respectively. The frame type and subtype can be distinguished, and thus the targeted frames (data and ACK) can be filtered from the others.

It is important to notice that, although the WLAN card driver allows direct access to the WLAN MAC level events at the code level (see the architecture overview in Fig. 3), the OS of the MT interfaces with the WLAN card via hardware interruptions. The driver code is executed once the OS has handled the interruption and the OS process scheduler has allocated the CPU to execute the corresponding network interruption process. Despite the higher scheduling priority of the hardware interruptions with respect to most other processes, this means that a slight time delay exists for the

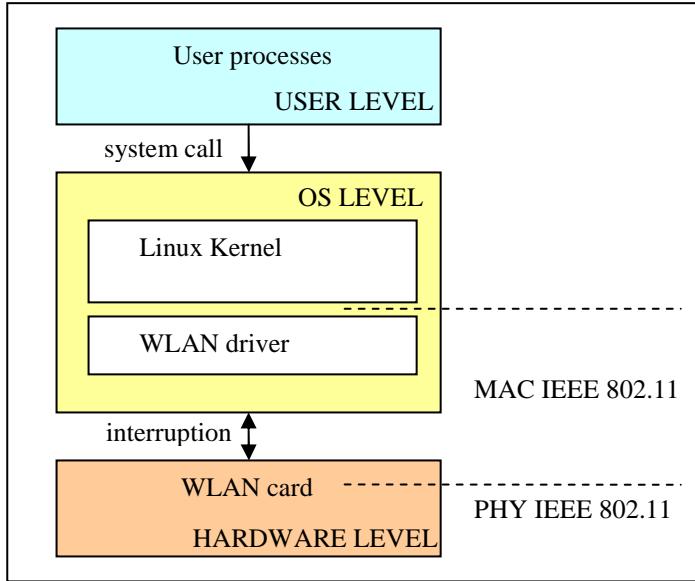


Fig. 3. Overview of the OS-IEEE 802.11 architecture

instant between the reception of an IEEE 802.11 frame at MAC level and the moment that this reception is time-stamped by the OS in the driver code. In practice, this entails non-negligible measurement error when measuring the *RTT*. Since this delay in the attention of the interruption is supposed to partially depend on the load in the OS, it is expected that it would have noticeable variability.

The employed time-base to time-stamp the frames corresponds to the CPU clock of the MT because it provides very good time resolution and it is accessible from the code of the WLAN card driver. Theoretically, this resolution would achieve ranging errors below one meter. The Time Stamp Counter (TSC) is a 64-bit register that counts cycles of the CPU oscillator, and it has proved to be reliable to count time differences [20]. It is stored in the EDX and EAX 32-bit registers and can be accessed by programmers with the *rdtsc()* (read TSC) function, which returns its value (the 32 least significant bits, stored in EAX) at that specific instant in the number of clock cycles. The call to *rdtsc()* is added to the code of the WLAN driver functions before mentioned. Once the returned values from *rdtsc()* for the data frame transmission and the ACK frame reception are subtracted to obtain the *RTT* measurement in clock cycles units, it is converted into time units with the CPU frequency.

3.2 The Ranging Process

An *RTT* measurement is affected by several noise sources that contribute to a positive bias error. This error has two primary origins. First, the *RTT* measurement system, including the WLAN interruptions' management by the OS, the possible fluctuations

of the CPU clock and the communication between the physical and MAC layer. Second, the propagation of the IEEE 802.11 signal through the radio-channel, which includes different adverse phenomena such as multi-path especially in indoor environments. It is expected that these errors cause non-negligible time variability (i.e. noise) in the *RTT*. Experimental measurements corroborate this conjecture, as shown below in Section 4. A natural approach to mitigate the influence of this noise is treating *RTT* as a random variable. A certain number of samples (*RTT* measurements) are collected together at a certain distance, and statistical processing is performed to estimate the *RTT* value as in [9]. Fig. 4 summarizes the complete ranging process using 1000 *RTT* measurements.

3.2.1 Statistical Processing

As depicted in Fig. 4, the statistical processing of the *RTT* measurements consists of two steps: 1) filtering and selecting the *RTT* samples that belong to a certain interval in order to discard spurious values (i.e. outliers); 2) applying a statistical estimator over the selected *RTT* samples in order to estimate the final *RTT* value.

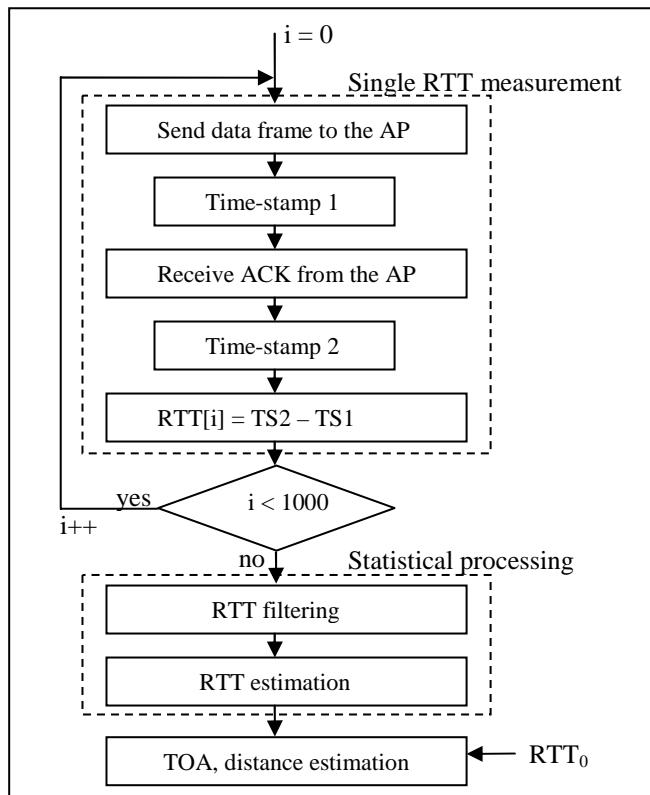


Fig. 4. Diagram of the complete ranging process

3.2.1.1 Filtering. The sample filtering can be understood as windowing over the RTT measurement histogram. The objective is to discard spurious values that would make the final RTT estimate more inaccurate. The interval for selecting the RTT samples is defined by its amplitude and its central point. Both parameters correspond to statistical parameters of the obtained RTTs, so that the interval depends on the RTT statistical distribution. Regarding the interval amplitude, the idea is that the more scattered the obtained RTT distribution, the longer the amplitude should be. Regarding the central point of the interval, the idea is that it is closer to the more representative RTT measurements.

The statistical parameters that are tested for the interval amplitude are the observed standard deviation (σ) and the standard deviation at distance zero (σ_0) and the parameters tested for the central interval point corresponds to the mode, the average (μ), the mode at distance zero (mode_0) and the average at distance zero (μ_0). Thus, the tested sample intervals are $[\mu-\sigma, \mu+\sigma]$, $[\text{mode}-\sigma, \text{mode}+\sigma]$, $[\mu_0-\sigma_0, \mu_0+\sigma_0]$ and $[\text{mode}_0-\sigma_0, \text{mode}_0+\sigma_0]$. While employing an interval defined by the observed *RTT* parameters makes the interval dependant on the distance between the MT and the AP (the central point of the interval increases with this distance), taking an interval with distance zero parameters means that the interval is constant regardless of the distance. The reason to consider the latter parameters is that they allow selecting smaller *RTT* values than in the former case, so the expected *RTT* positive error bias could be mitigated to a major degree.

3.2.1.2 RTT estimation. Regarding the RTT statistical estimators, the ones considered and studied in this work are the mode, the average and the estimator employed in [9] ($\mu-\sigma/3$). Notice that these mode or average values belonging to the RTT estimation are different from the mode or the average values explained previously for the central point of the filtering interval, because the former ones are obtained once the filtering has been performed and are therefore supposed to be more accurate. The mode is considered because the value that occurs more often is the one that provides the best estimate in a process with a high number of collected samples. The average is tested because it takes into account the information of all the selected samples, which can be especially suitable given the observed time variability of the performed measurements. The $\mu-\sigma/3$ is also considered because it provided good ranging accuracy in [9], and although the RTT measurement mechanism was noticeably different from the current one, measurements are taken in both cases with the MAC data-ACK exchange. The mentioned intervals and estimators are tested with real RTT samples in order to select the statistical processing that provides better accuracy (see Section 4). Once the RTT is estimated, formula (5) is employed to estimate the TOA, and formula (1) is used to obtain the distance between the MT and the AP.

4 Experimental Results

4.1 Test Setup

A prototype implementing the described ranging technique has been developed and tested. It carries out the *RTT* measurements and the statistical processing to estimate the distance with the AP. Each *RTT* measurement is initiated in the prototype inducing

the transmission of the MAC data frame sending an ICMP echo request to the AP. The *RTT* figure is calculated in the WLAN card driver's code as explained in section 3.1, and then is dumped to a file by means of the `printk()` function –which sends the data to the SO kernel– because it is not possible to use the typical functions from the `stdio.h` libraries (e.g. `print`, `scanf`) from the driver's code. The statistical processing is performed by a Java program, which obtains the *RTT* samples from the aforementioned file. The hardware platform is a laptop with an Intel Centrino Pentium M processor 760 (CPU 2 GHz), with an Ubuntu Edgy Eft 6.10 OS. The WLAN card is an IEEE 802.11b/g PCMCIA Netgear WG511T with Atheros chipset compatible with the Madwifi driver. In order to minimize the error due to the interruption management of the OS, the load of the CPU is restricted to the essential Linux OS kernel components. The employed AP is a laptop configured in AP mode in order to allow battery autonomy in the outdoor scenario.

Two measurements campaigns – one in an outdoor environment and the other indoors – are carried out in order to prove and define the *RTT* statistical processing and to test the feasibility and performance of the ranging technique. Both the MT and the AP are placed 1.5 m off the ground in order to preserve the Fresnel zone. Both the indoor and outdoor tests are performed in situations of Line-Of-Sight (LOS) between both nodes. The considered distances between MT and AP are 1, 4, 7, 10, 13, 16, 19, 22 and 25 m in both scenarios. A thousand *RTT* measurements are carried out for each *RTT* estimate.

According to equation (5), the *RTT* estimation at zero distance is used in all other distance estimations. Since it does not contain propagation time, it is assumed not to depend on the environment. Fig. 5 shows the *RTT* measurement histogram for zero distance outdoors. The x-axis corresponds to the number of CPU clock cycles and the y-axis to the number of each *RTT* measured value (i.e. the frequency). Each bar in the histogram represents a range of 100 clock cycles, which means 50 ns and 14.99 m in terms of distance. Thus, as expected, great time dispersion exists in the *RTT* measurements due to the noise produced by the described *RTT* measurements system.

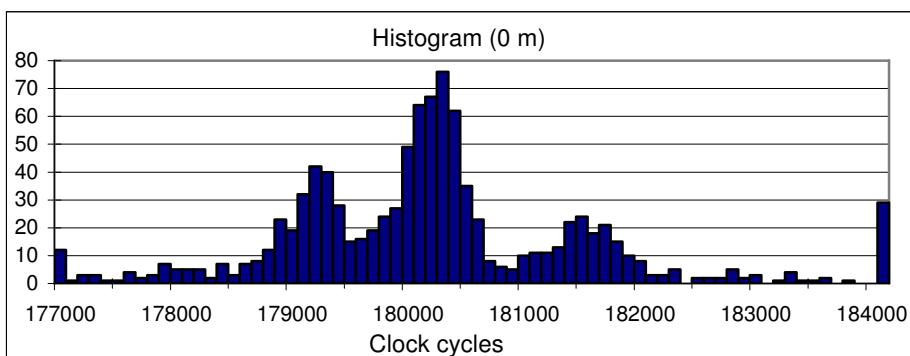


Fig. 5. RTT histogram at distance zero outdoors

Table 1. Ranging results outdoors with the average and the interval [mode₀- desv.est.₀, mode₀+desv.est.₀]

Distance (m.)	1	4	7	10	13	16	19	22	25
Estimated (m.)	1.08	0.32	4.75	8.73	14.7	14.3	15.9	21.26	22.85
Error (m.)	0.08	-3.67	-2.24	-1.26	1.74	-1.68	-3.01	-0.73	-2.14
Error (%)	8	91	32	12.6	13.4	10.5	15.8	3.31	8.56

4.2 Outdoor Tests

The chosen outdoor environment was an area far from buildings, so that the signal multi-path effect is expected to be low. Among the tested alternatives, the statistical processing that provides the best results consists of filtering the RTTs with the interval [mode₀-σ₀, mode₀+σ₀] and afterwards applying the average as the RTT estimator. This result corroborates the hypothesis given in section 3.2.2.1 about the major mitigation of the RTT positive error bias when employing a filtering interval with distance zero parameters. On the other hand, using this interval, 59.2% of the collected RTT samples are selected on average (considering all the tested distances). Fig. 6 depicts the RTT histogram at 13 m., with the filtering window darkened. Table 1 summarizes the obtained ranging results employing the mentioned statistical processing.

Taking into account all the distances, the average ranging error is 1.84 m. It is calculated with the following expression:

$$\text{average_absolute_error} = \sqrt{\frac{1}{N} \sum_{i=1}^N |a_i|} \quad (6)$$

where a_i is the absolute ranging error for the distance i and N is the number of considered distances. The average relative error (ARE) is 21.68 %. It is calculated as:

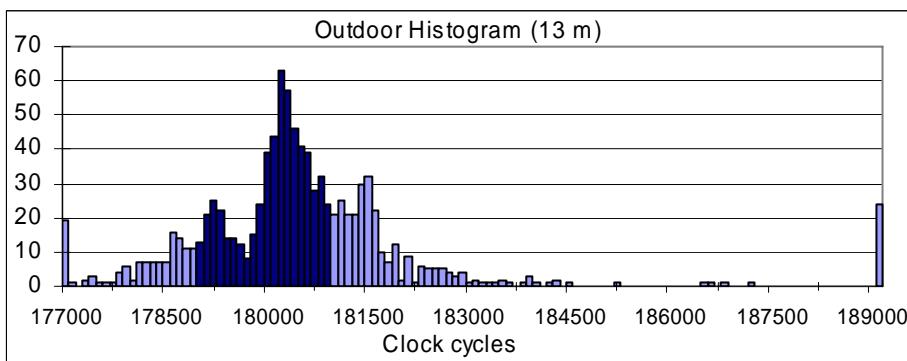


Fig. 6. RTT histogram at 13 m outdoors

$$\text{average_relative_error} = \sum_{i=1}^N r_i / N , \quad (7)$$

where r_i is the relative ranging error for the distance i and N is the number of considered distances. It must be observed that the ranging error does not increase with distance, which may lead to the conjecture that the measurement system errors have more impact than the ones due to the radio channel. This was expected because the signal propagation in an outdoor environment is hardly affected by multi-path.

4.3 Indoor Tests

Indoor tests were carried out in a corridor of the C3 building at the UPC in Barcelona (office environment). The estimator that provides better accuracy is again the average, but in this case filtering the RTT samples with the interval $[\mu_0 - \sigma_0, \mu_0 + \sigma_0]$. Again, the hypothesis given in section 3.2.2.1 is corroborated. In this case, 48.6% of the collected RTT samples are selected on average. The histograms for indoor cases are sparser than for outdoors, and this causes the number of RTT samples to be lower. It is conjectured that the sparser histograms are due to the impact of multi-path. Fig. 7 depicts the obtained RTT histogram at 13 m, with the window darkened. Table 2 summarizes the ranging results. An average absolute error of 1.70 m and an ARE of 37.24% are obtained.

Table 2. Ranging results indoors with the average and the interval [average₀- desv.est.₀, average₀+desv.est.₀]

Distance (m.)	1	4	7	10	13	16	19	22	25
Estimated (m.)	3.19	3.51	3.41	10.08	10.8	13.4	20.7	20.6	23.8
Error (m.)	2.19	-0.48	-3.58	0.08	2.17	2.51	1.74	-1.36	-1.15
Error (%)	219	12	51.14	0.8	16.7	15.7	9.1	6.18	4.6

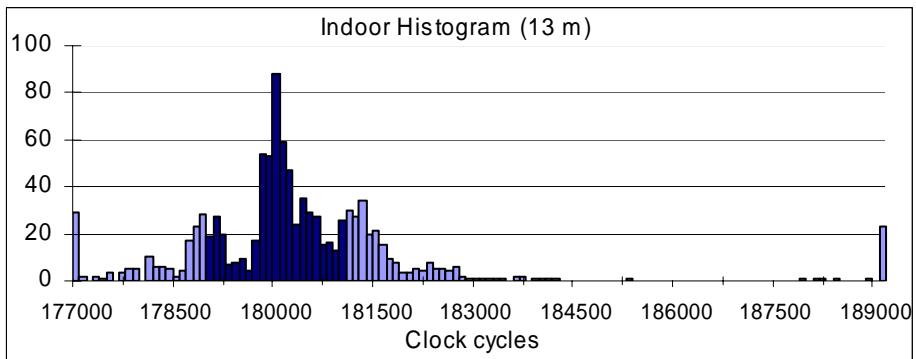


Fig. 7. RTT histogram at 13 m indoors

When comparing the results from both environments, we find that the ranging accuracy is similar for indoors and outdoors (the average relative error is bigger for indoors due to the big error for 1 m), but we have to be aware that, in both environments, tests are performed under LOS situations so that the only difference between them is the radio-channel multi-path. On the other hand, it is also important to observe that the error does not show an increasing tendency with distance. These two observations lead the authors to conjecture that the impact of the measurement system errors is more important than the impact of the propagation environment. This fact could explain why the best estimator found in [9] does not provide good results for the current technique. Considering that the *RTT* measurement mechanism is completely different in the two approaches, the related measurement noises are expected to follow different patterns, and thus the optimum estimators in the optimal statistical processing are completely different.

4.4 Ranging Latency

An important parameter for some applications is the latency: the time spent for one single distance estimation. This time basically corresponds to the time spent performing the required *RTT* measurements (data-ACK exchanges); the statistical processing duration is negligible. Table 3 shows the latency and average absolute error of the distance estimate for several amounts of required measurements.

As expected, the ranging latency is directly related to the number of *RTT* samples. It can be also observed that the ranging accuracy is also impacted by this parameter, mainly because a great number of samples are required to mitigate the measurement noise. Thus, for instance, reducing the number of samples 10 times (from 1000 to 100) gives 10 times better latency but makes the accuracy 5 times worse. Therefore, there is a trade-off between the latency (and CPU workload) and accuracy that must be tuned depending on the specific application.

Table 3. Ranging latency and accuracy depending on the number of RTT samples

Number of RTTs	1000	500	300	100
Latency (seconds)	1	0.5	0.3	0.1
Average ranging error indoors (m.)	1.70	4.72	6.64	9.03

4.5 Further Discussion

The experimental results shown above include only a reduced number of scenarios; however, we are aware that tests in more hostile scenarios are necessary in order to address problems that would appear when moving to a real deployable indoor positioning system that implements the presented ranging technique. For instance, in a true positioning situation at least three ranges must be estimated, and it is unlikely that all three involved APs are all in LOS; thus, it would be desirable to test the ranging technique under Non-Line-Of-Sight (NLOS) conditions (e.g. with walls, furniture, elevators obstructing the direct path between the MT and the AP). Another issue is that MTs

may run more processes apart from the positioning process; therefore, it may be interesting to asses the impact of the CPU load on the technique's performance.

On the other hand, a fact that must be taken into account is that the performance of the technique can suffer variations when other MTs are simultaneously injecting traffic to the AP, corresponding either to data-ACK exchanges for ranging or data transfers for communications over the WLAN. In that situation, collisions, retransmissions, busy medium detections and bigger contention periods may occur for the MTs trying to gain the shared medium, so that the latency and accuracy may be negatively affected. This problem of scalability has been explored in [21]. Related to this, it would also be interesting to study how to make compatible, in a single MT, the TOA-based ranging method with the WLAN data communications (e.g. file transfer, web browsing, video streaming), or at least to quantify the impact of that on the ranging performance. All these issues should be addressed as future work stemming from the presented research.

5 Conclusion and Future Work

A novel *TOA*-based technique to estimate the distance between a WLAN-enabled device and an AP has been presented. It has been demonstrated that it is feasible to achieve good ranging accuracy by means of a software-only solution for standard WLAN devices. The ranging process begins performing several *RTT* measurements with MAC IEEE 802.11 data-ACK frames exchanges; each *RTT* is calculated by time-stamping the frame transmission and reception from the WLAN driver's code while employing the CPU clock as a time-base. Since *RTT* measurements contain high time variability –mainly due to the OS WLAN interruption scheduling– a proper statistical processing method has been researched in order to mitigate this noise. Results obtained from measurements performed with a prototype show ranging errors below 2 m in indoor and outdoor environments. The results of this research can be applied to the design of WLAN indoor positioning systems with the accuracy of *TOA*-based techniques and the simplicity and cost-effectiveness of approaches based on received signal strength.

Acknowledgment

This research was funded by the Spanish Government and FEDER through the Plan Nacional de I+D (TEC2006-09466/TCM) and PROFIT FIT-330101-2007-7.

References

1. IEEE 802.11 WG Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Amendment v: Wireless Network management. IEEE P802.11v/D0.02. IEEE, New York (March 2006)
2. Youssef, M.: Horus: A WLAN-Based Indoor Location Determination System, Department of Computer Science, University of Maryland (2004)

3. Bahl, P., Padmanabhan, V.: Radar: An In-Building RF-based User Location and Tracking System. In: Proc. of 19th IEEE Infocom, pp. 775–784 (2000)
4. Ekahau Positioning Engine-Datasheet, <http://www.ekahau.com>
5. WhereNet, wireless solutions for tracking and managing assets, <http://www.wherenet.com>
6. Murphy, W., Hereman, W.: Determination of a Position in Three Dimensions Using Tri-lateration and Approximate Distances, Colorado School of Mines, Golden, CO tech. report MCS-95-07 (1995)
7. Ciurana, M., Barceló, F., Cugno, S.: Indoor tracking in WLAN location with TOA measurements. In: Proc. of 4th ACM Mobiwac, October 2006, pp. 121–125 (2006)
8. Caffery, J.J., Stüber, G.L.: Overview of radiolocation in CDMA systems. *IEEE Communications Magazine* 36(4), 38–45 (1998)
9. Ciurana, M., Barceló, F., Izquierdo, F.: A Ranging Method with IEEE 802.11 Data Frames for Indoor Localization. In: Proc. of Wireless Communications and Networking Conference, March 2007, pp. 2092–2096 (2007)
10. Velayos, H., Karlsson, G.: Limitations in range estimation for wireless LAN. In: Proc. of 1st Workshop on Positioning, Navigation and Communication (March 2004)
11. Li, X., Pahlavan, K.: Super-Resolution TOA Estimation With Diversity for Indoor Geolocation. *IEEE Trans. on Wireless Communications* 3(1), 224–234 (2004)
12. Aassie, A., Omar, A.S.: Time of Arrival estimation for WLAN indoor positioning systems using Matrix Pencil Super Resolution Algorithm. In: Proc. of 2nd Workshop on Positioning, Navigation and Communication, March 2005, pp. 11–20 (2005)
13. Ibraheem, A., Schoebel, J.: Time of Arrival Prediction for WLAN Systems Using Prony Algorithm. In: Proc. of 4th Workshop on Positioning, Navigation and Communication, March 2007, pp. 29–32 (2007)
14. Reddy, H., Chandra, G.: An improved Time-of-Arrival estimation for WLAN-based local positioning. In: Proc. of 2nd International Conference on Communication Systems software and middleware, January 2007, pp. 1–5 (2007)
15. McCrady, D.D., Doyle, L., Forstrom, H., Dempsey, T., Martorana, M.: Mobile ranging using low-accuracy clocks. *IEEE Trans. on Microwave Theory and Techniques* 48(6), 951–958 (2000)
16. Golden, S.A., Bateman, S.S.: Sensor Measurements for Wi-Fi Location with Emphasis on Time-of-Arrival Ranging. *IEEE Transactions on Mobile Computing* 6(10) (October 2007)
17. Günther, A., Hoene, C.: Measuring Round Trip Times to Determine the Distance Between WLAN Nodes. In: Boutaba, R., Almeroth, K.C., Puigjaner, R., Shen, S., Black, J.P. (eds.) *NETWORKING 2005. LNCS*, vol. 3462, pp. 768–779. Springer, Heidelberg (2005)
18. Muthukrishnan, K., Koprinkov, G., Meratnia, N., Ljding, M.: Using time-of-flight for WLAN localization: feasibility study, Centre for Telematics and Information Technology (CTIT) technical report, TR-CTIT-06-28 (June 2006)
19. Giustiniano, D., Lo Piccolo, F., Melazzi, N.B.: Relative localization in 802.11/GPS systems. In: Proc. of International Workshop on Satellite and Space Communications, September 2007, pp. 289–293 (2007)
20. Corell, E., Saxholm, P., Veitch, D.: A user friendly TSC clock. In: Proc. of Passive and Active Measurement Conference (March 2006)
21. Llombart, M., Ciurana, M., Barcelo-Arroyo, F.: On the scalability of a novel WLAN positioning system based on time of arrival measurements. In: Proc. of 5th Workshop on Positioning, Navigation and Communication, March 2008, pp. 15–21 (2008)

Position Estimation from UWB Pseudorange and Angle-of-Arrival: A Comparison of Non-linear Regression and Kalman Filtering

Kavitha Muthukrishnan¹ and Mike Hazas²

¹ Pervasive Systems Group, Department of Computer Science
University of Twente, The Netherlands

² Computing Department, Lancaster University, UK

Abstract. This paper presents two algorithms, non-linear regression and Kalman filtering, that fuse heterogeneous data (pseudorange and angle-of-arrival) from an ultra-wideband positioning system. The performance of both the algorithms is evaluated using real data from two deployments, for both static and dynamic scenarios. We also consider the effectiveness of the proposed algorithms for systems with reduced infrastructure (lower deployment density), and for lower-complexity sensing platforms which are only capable of providing either pseudorange or angle-of-arrival.

1 Introduction

While GPS is a common solution for outdoor localisation, indoor localisation still remains an open research problem. Location systems based on conventional radio technology have relatively coarse-grained performance indoors because their signals typically cannot be resolved accurately enough to produce quantities such as a time-of-arrival or angle-of-arrival. Instead, received signal strength indication (RSSI) or other metrics are used to estimate location via a technique called *fingerprinting*, but positioning errors of several metres often result because of indoor multipath fading. By contrast, ultra-wideband (UWB) systems can employ pulses of extremely short duration to achieve much finer signal resolution. This resolution also aids in the identification of erroneous measurements due to multipath [3].

UWB positioning systems can measure time-of-arrival, time-difference-of-arrival (TDOA) or angle-of-arrival (AOA), or some combination of them. A time-of-arrival can be converted into a range estimate, but measuring time-of-arrival directly is problematic, because the synchronisation signal (typically conventional radio) and the UWB positioning pulse both travel at near the speed of light.¹ Pseudoranging (based on TDOA measurements) is more attractive for some deployments, since there is no need for precise synchronisation between the transmitting and receiving entities. A network of

¹ If a wired tether were used to connect the receiver and transmitter, direct range measurement would be possible after calibration of the timing offset due to the tether. Direct range measurement using round-trip time-of-flight measurements are possible for devices capable of both transmitting and receiving UWB positioning pulses.

receivers can be precisely synchronised using stable clocks which are periodically corrected via a wired reference timing signal [14]. Producing AOA estimates requires a receiver equipped with an antenna array.

One UWB positioning system [2] estimates a tag's location using pseudorange data, with a stated accuracy of about 30 cm. Ubisense is a commercial UWB-based location system which performs measurement of both pseudorange and AOA (azimuth and elevation). The advantage of using AOAs as well as pseudoranges is that location can be determined with fewer sensors, compared to systems that use just pseudoranging. The reported accuracy (using their proprietary algorithms) is 15 cm.

This paper examines the results of two positioning algorithms operating on pseudorange and AOA data. The four primary contributions of this paper are as follows: (I) a brief characterisation of the raw observations reported by two different Ubisense deployments (Sect. 3); (II) formulation of two algorithms (regression using a non-linear model, and Kalman filtering; Sect. 4) which fuse the heterogeneous observations (pseudorange, azimuth, elevation) of a UWB system; (III) a characterisation of the static (Sect. 5) and dynamic (Sect. 6) tracking performance of the two algorithms; and (IV) an evaluation for reduced sensor densities and for pseudorange-only and AOA-only data, showing that the algorithms can support minimal deployments and less sophisticated UWB positioning sensors.

2 Related Work

Three approaches are generally used to calculate location using range, pseudorange, or AOA estimates. The first approach uses a simple geometric model to calculate intersection of circles (lateration), hyperbolas (hyperbolic localisation) or lines (angulation), depending on whether range, pseudorange, or AOA data is used. However, such simple algorithms typically do not take measurement error into account, and cannot make optimal use of redundant data (such as that gathered from a large number of receivers) which overspecify the solution.

By contrast, a second approach is to use optimisation algorithms which are specifically designed to find a solution which minimises the total error between the collected data and the location estimate (i.e. the *residual error*). These algorithms traverse the solution space and compute expected measurements for each estimate of the solution. The *gradient* method employs derivatives to observe the rate at which an area of the solution space converges towards an optimum. Some examples in this category are the method of steepest descent, Newton's method, and the Levenberg-Marquardt method [8]. Such methods require *model equations* which are used to express the measured values (such as pseudorange, or AOA) in terms of the position being solved. The Bat system [15], and other ultrasonic positioning systems [1] apply such model-based optimisation algorithms to compute location solutions from ranging data.

Scott et al. utilise a non-linear system of equations to compute location based on pseudoranges (in this case TDOAs of acoustic signals) [11]. Fontana comments on the performance differences between the steepest decent search and Davidon-Fletcher-Powell algorithm for estimating a tag's location using UWB pseudorange data [2]. It is

worth noting that these error minimisation algorithms do not make use of information obtained from prior location readings.

Algorithms that utilise a solution state (either current state, or current and past states) can be grouped under the third approach of *state-estimation algorithms*. State-estimation algorithms are used extensively in robot localisation. They operate by iteratively combining the previous estimate of the state (e.g. a position and orientation) with the observed measurements (range, AOA, etc.). Many state estimation algorithms exist [4], of which Kalman filtering [16] is most commonly used. The HiBall tracking system [17] employs a technique called single-constraint-at-a-time (SCAAT) tracking to model movement, and handles one observation at a time rather than obtaining multiple simultaneous measurements. The Constellation system [5] tracks a mobile unit consisting of a 3D inertial sensor and a number of ultrasonic sensors which report ranges that are fed into a SCAAT algorithm. The idea is to correct the positional drift in inertial tracking by incorporating ultrasonic range measurements in an extended Kalman filter. Smith et al. present a tracking algorithm using extended Kalman filtering of ultrasonic range data gathered from “Cricket” devices [12]. Smith et al. employ a combination of least squares minimisation, Kalman filtering and outlier rejection to predict the state of the Cricket device. While their approach is similar to our proposed Kalman filtering, we fuse heterogeneous data (pseudoranges and angles-of-arrival), and operate on UWB rather than ultrasound measurements.

Recent work by Renaudin and Kasser demonstrates the fusion of pseudoranges and AOAs from a Ubisense system together with inertial sensing data using an extended Kalman filter [9]. Curiously, their modelling equations are quite different to ours presented below (Eqns. 1–4) which we have verified through the analysis in this paper. Moreover, the goal of Renaudin and Kasser is the *augmentation* of inertial tracking with UWB sensing. There is no literature which explores in detail the fusion of heterogeneous data (pseudorange and AOA) through regression or Kalman filtering for UWB position sensing.

3 Deployments

In all our experiments, we used hardware and software procured from Ubisense. In brief, Ubisense hardware is comprised of two entities: a tag² which emits UWB pulses when triggered by the system; and receivers (“Ubisensors”) which are typically fixed at upper extremes of the measurement volume. The receivers are networked via CAT5 cabling and it can be assumed that they are tightly synchronised (once the cable timing offsets are estimated during calibration). A workstation PC is connected to the same network, and runs the Ubisense Location Engine (LE), which can be used to configure and calibrate the system, and which produces location estimates from receiver measurements, using proprietary algorithms.

The calibration of the receivers’ position and orientation are crucial in achieving accurate location estimates. The coordinates of a receiver’s position can be estimated

² Ubisense offer both a “slim” and a “compact” tag. We chose to use the compact tag for the data collection. It is advertised that the tag emits pulses in an omnidirectional fashion, and it performed as such during our initial, informal experiments.

with sufficient accuracy via manual methods, such as measuring the distance (using a tape measure or laser rangefinder) from the receiver centre to several known points in the environment. By comparison, accurate estimation of receiver orientation³ (yaw and pitch) is more difficult without special equipment, and additionally there can be small misalignments between the plastic casing of the Ubisensor, and the plane of the UWB receiver array inside. Thus, calibration of receiver pitch and yaw is normally undertaken using a series of measurements from a tag; the measurement logging and orientation estimation process is automated by the Ubisense Location Engine. To calibrate the receiver pitch and yaw for the systems at our sites, we used two different modes of the automated process, as described below.

3.1 Twente

Six receivers were deployed covering an area of approximately 15×9 m (Fig. 1(a)). Four of the receivers were deployed in an area which was relatively empty except for some desks along the corners of the room. The remaining two receivers were placed in the rooms along the other side of the corridor, in typical office spaces containing desks, metal shelving, and other furniture. We arbitrarily chose twenty-one test points across the measurement volume. Measurements were taken with tags placed at two different heights (75 cm and 151 cm above the floor). The positions of the sensors and the test points were surveyed using tape measures.

Automatic calibration. At Twente, we used the “automatic calibration” mode of the Ubisense LE. In automatic mode, a tag is placed at a position within the measurement volume. The height of the tag must be known and provided to the LE by the user, but the system then takes measurements to estimate the horizontal position of the tag, and the pitch and yaw of those receivers which can reliably detect the tag’s pulses. The tag is then moved to another position and this process is repeated until the pitch and yaw of all receivers is estimated.

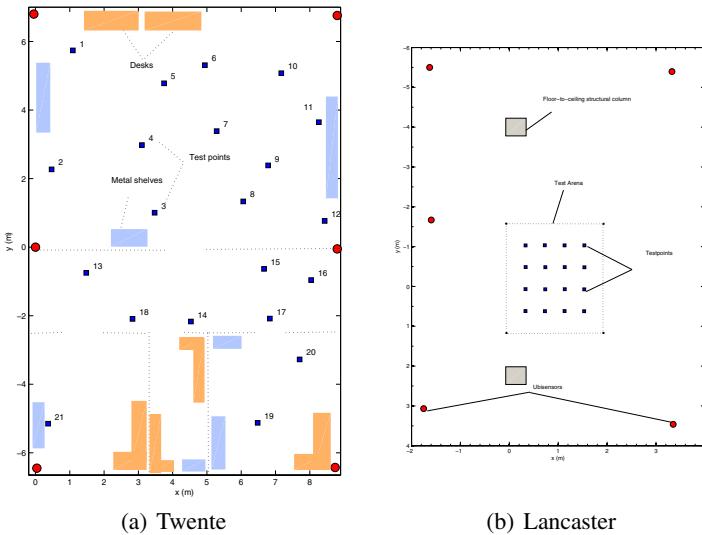
3.2 Lancaster

At another site, a smaller measurement area of 2.75×2 m was covered with five receivers. Readings were taken with a stationary compact tag placed at four heights (about 0, 7.5, 75, and 125 cm above the floor), at sixteen points across the measurement area (Fig. 1(b)). The positions of the sensors and the test points were surveyed using a Leica Total Station.⁴

Dual calibration. At Lancaster, the LE “dual calibration” mode was used to estimate receiver pitch and yaw. In this mode, a tag is placed at a known position in the environment, and its 3D coordinates are provided to the LE. Measurements of the tag’s pulses are then gathered from two receivers, and the receivers’ pitch and yaw are estimated for one or both of the receivers (selectable by the user). The dual calibration has the

³ Ubisense use the aeronautical terms *yaw*, *pitch* and *roll* to describe receiver orientation [13].

⁴ A Total Station is a professional surveying device which accurately measures the range, azimuth, and elevation between itself and a reflector, and which can thus be used to estimate the 3D coordinates of the surveyed point.

**Fig. 1.** Plan views of the deployment areas

potential to more accurately estimate pitch and yaw, since the estimation process does not depend upon the accuracy of the LE’s result for the tag’s horizontal position (as it does in the automatic calibration mode). Ubisense recommend that when performing calibration, the tag be placed as near as possible to the boresight of the receiver(s) being calibrated. At Lancaster, we ran the dual calibration for all five receivers at each of twelve known points (whose coordinates had been surveyed using the Total Station) in or near the test arena. The twelve calibration points (which are different from the test points used in our evaluation) were each selected for their favourable line-of-sight to at least four of the five receiver units. To remove poor pitch and yaw estimates (typically due to poor line-of-sight or environmental reflections), we took the median of the twelve pitch and yaw values for each receiver as our final calibration.

3.3 Characterisation of Raw Measurements

Fig. 2 shows the distributions of the receivers’ raw measurements for a static tag placed at the test points at each site (sixty-four locations at Lancaster, and forty-two locations at Twente). The accuracy of the Lancaster measurements is significantly better than that of the Twente measurements (Table 1). The Twente deployment covers a wider area, and not all receivers have line-of-sight to all the test points. In its measurement and location estimation process, the LE identifies receivers whose measurements were rejected for the purposes of computing a location; the rejected measurements typically correspond to receivers which had poor line-of-sight to the tag (and may therefore represent multipath readings). We plot the raw accuracy with these rejected measurements excluded. In total, the discarded measurements represent 25% of the Twente dataset.

Even with the LE-rejected measurements excluded, the data from the Twente deployment is significantly worse in accuracy than the Lancaster data. In general, there

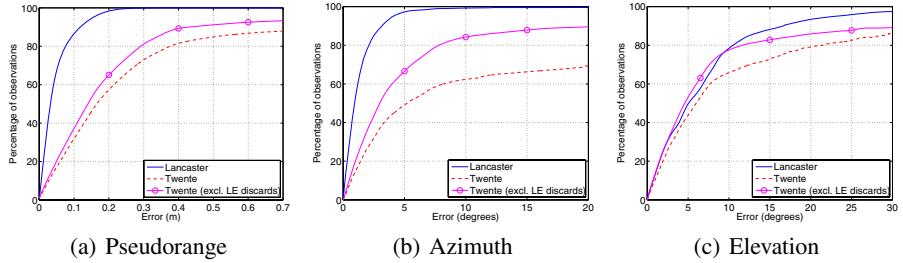


Fig. 2. Raw measurement error distributions

Table 1. Seventy-fifth percentile accuracy of the raw measurements of two Ubisense deployments

	Pseudorange	Azimuth	Elevation
Lancaster	6 cm	1.5°	9°
Twente	37 cm	22°	17°
Twente (excl. LE discards)	25 cm	6.8°	8°

are three sources of raw measurement error in any location system: (a) sensor inaccuracy; (b) calibration inaccuracy (sensor position, pitch and yaw), and (c) inaccuracies induced by environmental effects (attenuation and/or reflection of the UWB pulse). The Lancaster deployment was more carefully calibrated—a Total Station survey as opposed to hand measurements; and dual calibration of all receivers at twelve surveyed points, as opposed to automatic calibration at several arbitrary points. Moreover, the Lancaster deployment was specifically designed to accurately monitor the measurement volume; all five receivers have favourable line-of-sight to the majority of the sixty-four test points. By contrast, the Twente deployment has a much lower receiver density, covering a measurement volume containing office furniture and walls. For these reasons, we would argue that the inaccuracy of the Twente raw measurements is due to a poorer calibration (visible in the significant error offsets for some of the Twente receivers), and less favourable line-of-sight and multipath conditions caused by the environment.

In summary, the two sites represent very different kinds of deployment. The Lancaster deployment has been carefully calibrated using specialised equipment and system-specific knowledge, and has been designed to monitor a small volume using a high sensor density. The Twente installation covers a much larger volume, the environment was left unmodified (leading to a higher degree of unfavourable pulse propagation for some tag locations), and sensor position and orientation have been calibrated based on fewer measurements taken with non-specialised, less accurate equipment.

4 Overview of Algorithms

4.1 Non-linear Regression

We formulate a non-linear regression algorithm [8, section 15.5] which utilises modelling equations. The regression is an iterative process which finds an estimate of the

tag's location. The estimate can be seen as a “best fit” for the pseudoranges and/or angles-of-arrival and the surveyed receiver locations and orientations, since it minimises the sum of squares of the residual errors.

Pseudoranges. For a receiver i , the receiver's pseudorange estimate (based on the measured relative arrival time of the pulse emitted by the tag) \tilde{d}_i , the receiver's 3D location (x_i, y_i, z_i) , and the tag's location (u, v, w) can be related as follows:

$$\tilde{d}_i = \sqrt{(u - x_i)^2 + (v - y_i)^2 + (w - z_i)^2} - d_c, \quad (1)$$

where d_c is the distance offset common to all receiver pseudorange measurements, and arises from the tag's unknown clock offset from the system. It is assumed that the receivers are very tightly synchronised, or their reported pseudoranges have been otherwise appropriately adjusted (using for example deployment-specific calibration information) for any time offset which may exist between receiver units.

Angles-of-arrival. The receiver-reported azimuth ϕ_i of a tag is related to the tag's position (u_s, v_s, w_s) in the receiver's frame of reference.

$$\text{Quads. I \& IV: } \phi_i = \arctan\left(\frac{v_s}{u_s}\right) \quad \text{Quads. II \& III: } \phi_i = \left(\pi + \arctan\left(\frac{v_s}{u_s}\right)\right) \quad (2)$$

Similarly, the elevation θ_i of a tag measured by a receiver is defined as

$$\theta_i = \arctan\left(\frac{w_s}{\sqrt{u_s^2 + v_s^2}}\right) \quad (3)$$

Note that a four-quadrant definition of the elevation θ_i is unnecessary, since the denominator of the arctangent operand is always positive.

To compute the tag location in the receiver's frame of reference, first the coordinates of the tag $(u_{\text{rel}}, v_{\text{rel}}, w_{\text{rel}})$ relative to the receiver are computed by subtracting the receiver coordinates from the coordinates of the tag in the global frame of reference, i.e. $(u - x_i, v - y_i, w - z_i)$. This effectively translates the origin of the global coordinate system to the location of the receiver. These tag-to-receiver relative coordinates must be transformed to the receiver's frame of reference. This is first a rotation about the Z axis by an amount corresponding to the receiver's yaw φ_i , followed by a rotation about the Y axis by an amount corresponding to the receiver's pitch ϑ_i . The rotation matrix R_{GS} can thus be defined as:

$$R_{\text{GS}} = \begin{bmatrix} \cos(\vartheta_i)\cos(-\varphi_i) & -\cos(\vartheta_i)\sin(-\varphi_i) & \sin(\vartheta_i) \\ \sin(-\varphi_i) & \cos(-\varphi_i) & 0 \\ -\sin(\vartheta_i)\cos(-\varphi_i) & \sin(\vartheta_i)\sin(-\varphi_i) & \cos(\vartheta_i) \end{bmatrix} \quad (4)$$

Note that Ubisense yaw is defined as positive anticlockwise (looking at the XY plane from the Z+ direction), and Ubisense pitch is defined as negative anticlockwise (looking at the XZ plane from the Y+ direction). The position of the tag (u_s, v_s, w_s) in sensor's frame of reference is then computed by multiplying the tag's coordinates relative to the receiver $(u_{\text{rel}}, v_{\text{rel}}, w_{\text{rel}})$ by R_{GS} .

Outlier rejection. The number of observations required (pseudoranges and/or angles-of-arrival) depends upon the particular tag-to-receiver geometry. In typical Ubisense deployments, to estimate both tag location and the distance offset \hat{d}_c at minimum either two pseudoranges, one azimuth, and one elevation, or four pseudoranges are needed. To estimate only tag location, at least three angles-of-arrival (including at least one azimuth and one elevation) are needed. For locations estimated using more than the minimum number of observations, the standard error s of the estimate can be calculated from the residual errors e_i :

$$s = \sqrt{\frac{\sum_{i=1}^I e_i^2}{I - C}}, \quad (5)$$

where I is the number of observations being used in the non-linear regression, and C is the minimum number of observations required. C is set to 3 if only tag location is being estimated, or to 4 if the distance offset \hat{d}_c is also being estimated. The non-linear modelling process assumes the residual errors $e_{i=1\dots I}$ are normal, independent, and have equal variance and zero mean.

With heterogeneous data such as that reported by Ubisense receivers (pseudorange, azimuth, and elevation), the computed residuals have different units. Thus, before applying Eqn. 5 to estimate the standard error, each residual should be scaled by a typical magnitude of error which might be expected for each type of data. To perform scaling of residuals, we utilised divisors of 30 cm, 4° and 3° for pseudorange, azimuth, and elevation, respectively. These were chosen to work with both the Twente and Lancaster datasets, rather than being specifically tailored to either one.

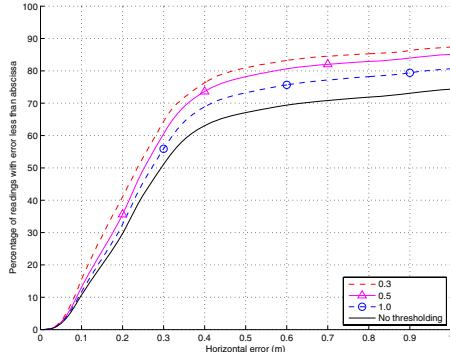
If the observations given to the regression do not corroborate one another, the result will typically have a high standard error estimate. Standardised residuals [6, ch. 4] can be used to identify the observations which agree the least with the solution. If the standard error estimate was greater than 0.30, our regression algorithm removes the observation with the highest standardised residual, and re-computes the solution. This process continues until the standard error of the estimate falls below 0.30, or the highest residual is less than 1.0.

Even after the outlying observations have been removed, the location accuracy can still be poor (Fig. 3(a)). To improve accuracy, one tactic is to simply reject location readings which have a high standard error estimate. For the regression results presented in the remainder of this paper, we have omitted estimates with a standard error higher than 0.5 (except where otherwise noted). Assuming both pseudorange and angle-of-arrival data is used, this corresponds to about 20% of the readings taken at Twente (Fig. 3(b)), and very few (less than 0.1%) of the static readings taken at Lancaster.

4.2 Extended Kalman Filtering (EKF)

We formulate an Extended Kalman Filter (EKF) using a state vector \hat{x}_k with seven variables, three position variables u, v, w , user clock offset d_c and three velocity variables v_u, v_v, v_w . After any discrete time step, which is approximately 108 ms⁵ the filter has an

⁵ The tag emits a UWB pulse once every four Ubisense time slots. With our particular version of Ubisense hardware, each time slot corresponds to 27.029 ms.



(a) Horizontal error distribution

Std. error threshold	75% horiz. error (cm)	Fraction accepted (%)
0.3	38.4	64.2
0.5	42.1	80.4
1.0	57.0	91.2
None	106.4	100

(b) Trade-off of accepting regression solutions based on standard error

Fig. 3. Horizontal accuracy at Twente for different standard error thresholding levels

idea of its state and how confident it is in that state (prediction). The filter then corrects the predicted state based on the most recent measurements (pseudoranges and angles) and its internal state. Note that we use an Iterative Extended Kalman Filter (IEKF) [7], which is an extension to the standard EKF, and is useful for reducing the errors that may occur due to large non-linearities in the system.

Initialisation. The filter is initialised with a posterior state estimate \hat{x}_k^- and uncertainty P_k^- . Kalman filter estimates rely heavily on these initial estimates. We set the initial state estimates based on averaging the first twenty measurements through non-linear regression (as described above), discarding any with high standard error. Since we have set the tag at the highest update rate, this corresponds to a wait of a few seconds to start the Kalman filter. We chose to use a small but non-zero value for P_k^- , meaning that there is a little uncertainty in the defined initial state.

System Model and Measurement Model. We use a *constant-velocity model*, i.e. it is assumed the tag moves at constant speed between time steps. Thus, the new state estimate \hat{x}_k will depend on the previous state estimate \hat{x}_{k-1} , constant velocity v_{xk} and a noise term w_k (as in, $\hat{x}_k = \hat{x}_{k-1} + v_{xk} dt + w_k$). In order to predict the state using the measurements, we will have to describe how the measurements are related to the state. The *measurement model* $\tilde{z}_k (= H\hat{x}_k + v_k)$ describes how measurements depend on the state estimates \hat{x}_k . H is the Jacobian matrix with partial derivatives of the measurement function with respect to the state \hat{x}_k . The measurement function here represents the pseudorange (\tilde{d}_i), azimuth (ϕ_i) and elevation (θ_i) as defined earlier in Sect. 4.1.

Prediction & Correction. The predicted error covariance (P_k^-) and the state estimate (\hat{x}_k^-) for a time-step is given by:

$$\begin{aligned}\hat{x}_k^- &= A\hat{x}_{k-1} + Bu_k \\ P_k^- &= AP_{k-1}A^T + Q.\end{aligned}\tag{6}$$

Here, A is the Jacobian matrix and Q is the process noise covariance. The process noise covariance Q for a position-velocity model includes the process covariance in position and velocity and the process covariance in the position.

The filter computes the posterior state estimate by taking the prior state estimate and combining with the Kalman gain K_k times the difference between the actual measurement (pseudoranges and angles) and a measurement prediction ($\hat{z}_k = H\hat{x}_k + v_k$), called the *innovation* or residual r . If the innovation is zero, then the predicted state estimate exactly reflects the real measurement. But if there is a difference between the predicted and the observed measurement, then the prior state estimate needs to be updated. In eq. 7, K_k determines to what extent the innovation should be used in the posterior state estimation. Based on the measurement noise R and the prior error covariance P_k^- , the gain can favour the innovations or the measurements more. The measurement noise R for pseudorange and angles (azimuth and elevation) is set to be 10 cm, 7° and 10° respectively.

$$\begin{aligned} K_k &= P_k^- H^T (H P_k^- H^T + R)^{-1} \\ \hat{x}_k^{i+1} &= \hat{x}_k^- + K_k(z_k - H(\hat{x}_k^- - \hat{x}_k^i)) \\ P_k &= (I - K_k H)P_{k-1}^- \end{aligned} \quad (7)$$

In Eqn. 7 the Jacobian matrix H is evaluated at the most recent intermediate state estimate \hat{x}_k^i (difference between the IEKF and EKF). After a number of iterations or when the intermediate state estimate does not differ with more than a certain threshold from \hat{x}_k^{i-1} , the filter sets the posterior state estimate and estimates its posterior uncertainty. It is important to note that the IEKF computes the uncertainty in the state only after it finds the most accurate intermediate state estimate. Though the computations involved in IEKF are larger than the standard EKF, the state estimates will be better because of re-evaluation of the measurement function and the Jacobian.

Validation gating. It is imperative to employ some form of outlier rejection (also known as validation gating in Kalman literature) as part of the filter, since noisy measurements can cause the filter arrive at a bad state estimate. In most cases, innovation or residual r is used to identify outliers. An approach to eliminate outliers is based on $r^2 S^{-1} > \gamma$, where S^{-1} is a scalar based on the state and γ is an empirically-chosen parameter [12]. Alternatively, the distribution of innovations can be used to detect innovations that are unlikely to occur. Recent work by Renaudin looks at the possibility of eliminating outlying measurements based on human body orientation and comparing the predicted state with the current measurements. Another interesting approach is to check the condition for *optimality* of the filter. It is reported [10] that through the usage of statistical methods it is possible to check if the innovation sequence is *white* (a sufficient and necessary condition for testing optimality). If the filter shows sub-optimality, R and Q can be adjusted in order to make the filter optimal, thereby making the Kalman filter *adaptive*. We however chose to use the simple strategy of using the innovation distribution (based on setting thresholds determined empirically) to identify outlying measurements.

5 Static Positioning

For the static positioning experiments at Lancaster, a tag was placed at the sixteen test points (Fig. 1(b)), at four different heights. At each of these sixty-four locations, receiver readings were gathered for about one thousand tag pulses and about 1000 readings were gathered. At Twente, a tag was placed at the twenty-one test points (Fig. 1(a)) at two different heights, for a total of forty-two locations, with about 1500 readings at each. For our experiments at both sites we set the tags to run at the highest update rate, emitting a UWB pulse once every four Ubisense time slots, which for our Ubisense hardware is equivalent to about ten times per second. All receiver observations were fed into our algorithms, even those rejected by the Ubisense LE (see Fig. 2).

5.1 Results from Heterogeneous Observations

Fig. 4 shows the accuracy of the regression and Kalman filtering algorithms when they are provided with the heterogeneous measurements (pseudoranges and angles-of-arrival) produced by the receivers. Note that the vertical accuracy at Lancaster is more than thirty centimetres worse than the horizontal accuracy. This might be explained in two ways. First, the geometry of the Ubisense deployment (receivers are approximately co-planar) and the types of measurements they take (pseudorange, azimuth, and elevation) mean that there is more information about the tag's horizontal position solution (heavily contributed to by pseudorange and azimuth), and less about the vertical position of the tag (primarily affected by the elevation readings). Second, referring to Fig. 2, one can see that the pseudoranging and azimuth accuracy tend to be better than the elevation accuracy. At Twente, the opposite is true: the vertical accuracy tends to be superior to the horizontal. However, about 30% of the Twente raw azimuth readings are off by more than 20°, whereas the elevation accuracy is comparable to that at Lancaster (Fig. 2(c)).

On the Lancaster data, the Kalman filter performs slightly better than the regression. This is because the Kalman filter utilises its predicted estimate for the position of the tag, in addition to new receiver observations, to produce the new tag estimate. By contrast, the regression algorithm utilises only the receiver observations from the current

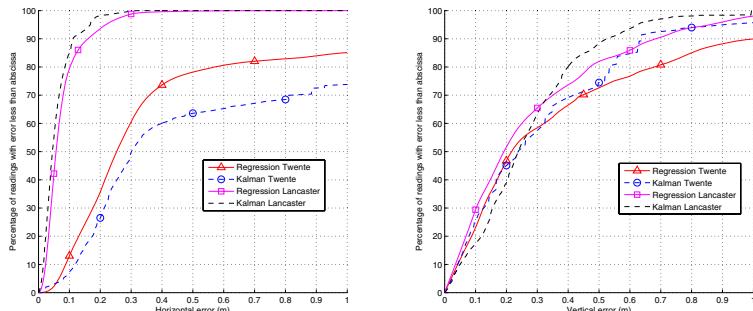


Fig. 4. Positioning static tags using heterogeneous measurements (pseudoranges and AOAs)

Table 2. Performance summary of static positioning using heterogeneous data

Algorithm	Site	75% confidence level (cm)		90% confidence level (cm)		Estimates accepted (%)
		Horizontal	Vertical	Horizontal	Vertical	
Regression	Twente	42.07	54.74	155.59	101.04	80.42
Kalman	Twente	107.82	51.86	276.95	63.72	N/A
Regression	Lancaster	8.83	42.13	16.02	68.46	99.98
Kalman	Lancaster	7.51	37.35	11.14	52.61	N/A

time slot to compute a solution. Thus the Kalman filter can be slightly more accurate, especially for static situations where the location estimate is stable for consecutive observations. Note that for the Twente data, the Kalman filter has much worse horizontal error than the regression. However, this is because the regression rejects results with a high standard error estimate. Rejecting no readings (Fig. 3), the regression's 75% horizontal accuracy of 106 cm is comparable to the Kalman filter's 108 cm.

Effect of reduced receivers. Decreasing the required infrastructure can reduce installation and calibration cost. To show how well our algorithms support reduced infrastructure, we compute the tag locations based on the observations from subsets of receivers. As expected, the accuracy of both algorithms decreases as infrastructure density decreases. However, the Kalman filter degrades much more gracefully; note in particular the difference in the vertical accuracy results for Kalman and regression. Moreover, the regression rejects more readings as deployment density decreases, whereas the Kalman filter consistently supplies reliable estimates regardless of density.

Table 3. Reducing deployment density (heterogeneous data, Lancaster static measurements)

	No. of receivers	75% confidence level (cm)		90% confidence level (cm)		Estimates accepted (%)
		Horizontal	Vertical	Horizontal	Vertical	
Non-linear regression	All	8.83	42.13	16.02	68.46	99.98
	4	11.38	47.59	18.25	75.05	99.66
	3	15.02	55.82	22.85	86.29	95.07
	2	24.24	64.76	36.29	108.11	52.78
Kalman filtering	All	7.51	37.35	11.14	52.61	Not applicable
	4	9.10	37.38	13.69	53.02	
	3	11.38	37.60	17.07	51.93	
	2	14.45	35.74	17.26	53.87	

5.2 Results from Homogeneous Observations

Both the algorithms we have presented are capable of computing location estimates using only pseudoranges or angles-of-arrival. It is interesting to consider their performance on homogenous data in order to judge our algorithms' application for UWB position-sensing platforms having fewer capabilities than Ubisense. For example,

receivers not equipped with array processing would not be capable of measuring AOA. Or, nodes in a distributed, wireless sensor network might only be able to measure angle-of-arrival, as the nanosecond-level node synchronisation required for pseudoranging is prohibitively difficult without a wired connection.

Pseudoranges only. Fig. 5 shows the accuracy of the two algorithms operating on pseudorange data only, at both sites. As in the heterogenous case, the regression performs significantly better than the Kalman filter on the Twente data; however note the ratio of readings rejected by the regression—over four-fifths for the Twente dataset (Table 4). Even for the Lancaster dataset, the regression rejects over 15% of readings. On the Lancaster data, the Kalman filter's accuracy is comparable, and it provides constant updates (no readings are rejected). Note that in general the vertical accuracy is inferior compared to the heterogeneous case. This is because the vertical solution is less constrained when only pseudoranges are used (no elevation data is present). The 3D accuracy of the Twente results are especially poor, since the accuracy of the underlying pseudoranges is much lower than in the Lancaster deployment (37 cm compared to 6 cm, at the 75% confidence level).

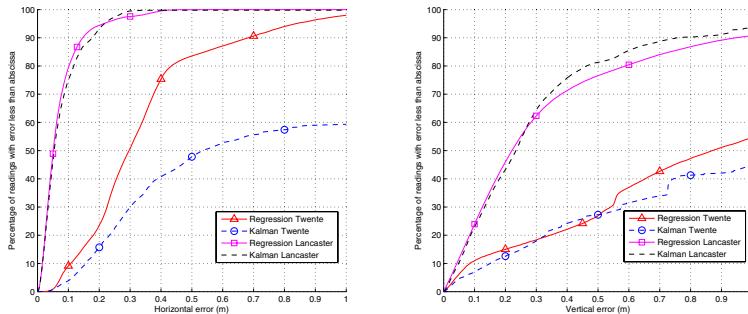


Fig. 5. Positioning static tags using pseudoranges only

Table 4. Performance summary of static positioning using homogeneous data

Data type	Algorithm	Site	75% conf. level (cm)		90% conf. level (cm)		Estimates accepted (%)
			Horizontal	Vertical	Horizontal	Vertical	
Pseudo-ranges	Regression	Twente	39.80	141.28	68.30	233.93	18.48
	Kalman	Twente	330.62	491.51	1983.08	1229.96	N/A
	Regression	Lancaster	8.85	46.48	14.66	94.44	84.14
	Kalman	Lancaster	10.03	39.17	17.75	75.80	N/A
AOAs	Regression	Twente	190.55	64.10	435.11	103.02	63.93
	Kalman	Twente	452.10	59.62	678.16	119.24	N/A
	Regression	Lancaster	19.56	38.95	30.55	70.40	98.1
	Kalman	Lancaster	17.18	37.15	26.89	54.92	N/A

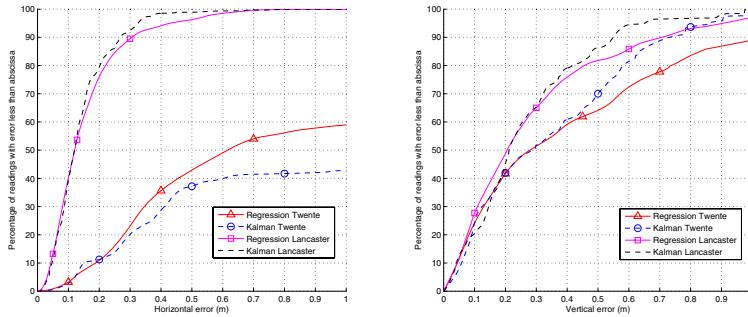
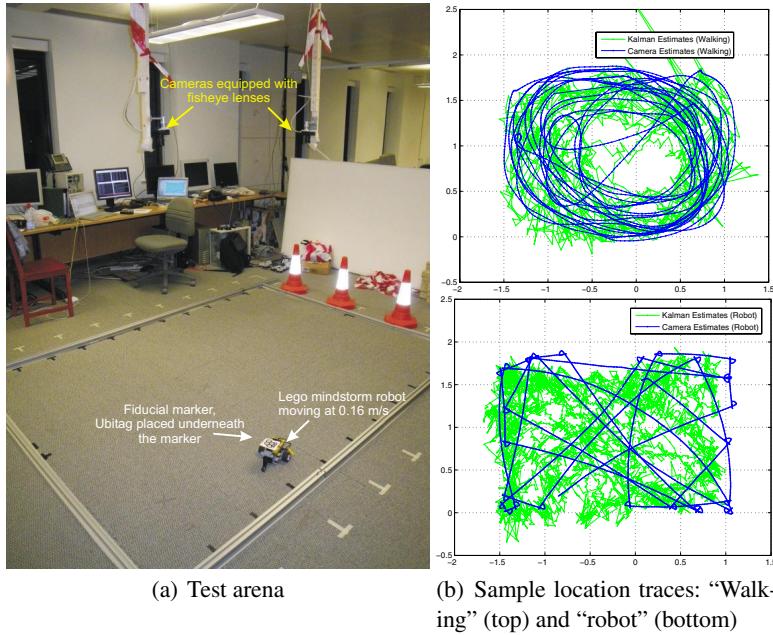


Fig. 6. Positioning static tags using angles-of-arrival (azimuth and elevation)

Angles-of-arrival only. Using only angles-of-arrival, the two algorithms perform comparably (Fig. 6 and Table 4). The accuracy for the Lancaster deployment is quite favourable (horizontal and vertical 75% confidence better than 20 and 40 cm, respectively). This is because for the majority of the Lancaster readings, a reliable azimuth and elevation were reported by all five receivers. The Twente AOA-only results are quite the opposite of the Twente pseudorange-only results. While the Twente vertical accuracy is about as good as the heterogeneous case (Fig. 4), around half of the readings exhibited a horizontal accuracy worse than one metre. Inspecting the per-receiver distributions reveals that three out of the six Twente receivers returned very poor azimuth estimates about 25% of the time. (As mentioned above, this is most likely due to environmental effects and an inaccurate calibration.) Thus, without the pseudorange contribution to the solution, the seventy-fifth percentile horizontal accuracy falls from several tens of centimetres in the heterogeneous case, to several metres in the AOA-only case.

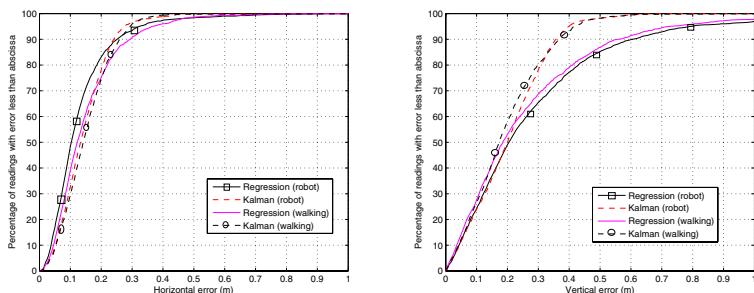
6 Dynamic Tracking

At Lancaster, dynamic data traces were taken in the test arena. Data was recorded (Fig. 7(a)) in traces lasting approximately seven minutes each, resulting in about 2500 pulse readings per trace. We gathered two types of dynamic data (Fig. 7(b)): “robot” data (four traces) was generated as a Lego Mindstorms robot roamed the arena (velocity 0.16 m/s); and “walking” data (three traces) was recorded as a person pulled the robot around in the arena using a tether (peak velocity of about 1 m/s). The ground truth positions of the tag in the dynamic experiments were recorded using computer vision-based localisation. Two cameras equipped with fisheye lenses were placed above the measurement area, a fiducial marker was rigidly attached to the top of the tag, and reactIVision software was used to perform accurate localisation in real time. Since there are slight differences in the timestamps between the logged camera estimates and the ubisense estimates, we use a weighted-average method to interpolate the camera data for the corresponding ubisense timestamp. Where there were large gaps in the camera timestamps, the interpolated values may not be valid, hence interpolation was done only if the surrounding data points are close in time (shorter than 300 ms). For gaps greater

**Fig. 7.** Lancaster dynamic experiment

than this, the Ubisense readings are discarded, to allow accurate comparison. For most traces, this method resulted in about 10% of receiver observations being dropped.

The location accuracy for the “robot” and “walking” traces is roughly equivalent for each algorithm, despite the large difference in the speed of the tag. This is likely due to the sufficient update rate of Ubisense readings (about 10 Hz) for both types of trace. Since the accuracy does not change appreciably between the two speeds, for the homogeneous dynamic case (Fig. 9 and Table 5), we consider only the three “walking”

**Fig. 8.** Dynamic tracking of tags using heterogeneous data (Lancaster)

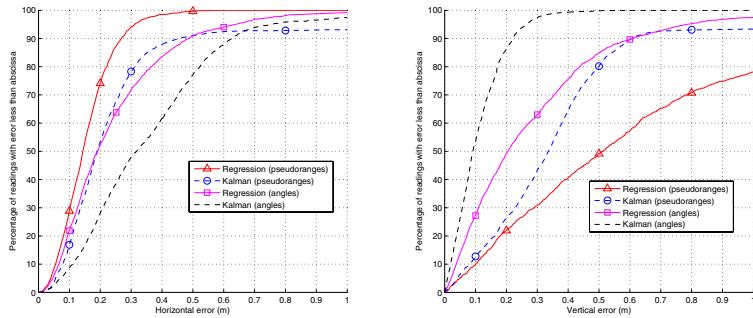


Fig. 9. “Walking” tracking accuracy using homogeneous data (Lancaster)

Table 5. Dynamic tracking performance summary (Lancaster). All values shown pertain to the location results of the three “walking” data traces.

Data type	Algorithm	75% conf. level (cm)		90% conf. level (cm)		Estimates accepted (%)
		Horizontal	Vertical	Horizontal	Vertical	
Pseudoranges and AOAs	Regression	18.31	35.31	27.46	55.43	99.66
	Kalman	21.02	26.07	31.42	36.12	N/A
Pseudoranges only	Regression	20.29	90.63	26.86	167.24	61.86
	Kalman	33.30	46.27	55.84	53.90	N/A
AOAs only	Regression	32.43	39.58	48.53	61.01	94.83
	Kalman	48.69	15.87	62.87	22.16	N/A

traces. The “walking” homogeneous results are comparable to the estimates for static data taken at similar height. For example, the regression operating on pseudoranges only yields a 75% horizontal dynamic accuracy of 20 cm, which is exactly the same accuracy as for the static readings taken at a height of 7.5 cm.

7 Ubisense Location Engine Estimates

Although the goal of this paper is to characterise our algorithms’ performance with different deployments and types of UWB positioning data (pseudorange and angle-of-arrival), we provide here the estimates produced by the Ubisense Location Engine for purposes of comparison. Fig. 10 shows the static tag positioning accuracy of the LE when configured to “default no filtering.” The regression and Kalman algorithm results for the same data are also shown. Like our regression algorithm, the LE produces a measure of standard error. But, because we do not know how the LE standard error is calculated, for this comparison we have not rejected any estimates from the LE or the regression results. With no filtering, the Ubisense LE performs quite comparably to our proposed algorithms for static tags. Likewise, the dynamic accuracy of the LE is similar to our results (Fig. 8) if “no filtering” is set.

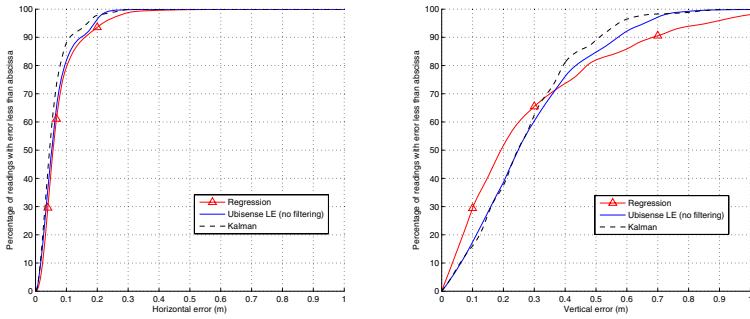


Fig. 10. Ubisense Location Engine accuracy (Lancaster, static readings)

8 Conclusion

This paper has presented a non-linear regression and a Kalman filtering algorithm designed specifically to process the heterogeneous data which UWB positioning systems are capable of producing. Both algorithms fuse the different types of raw data (pseudorange, azimuth, and elevation) effectively. For reliably accurate raw data (as produced by the Lancaster deployment), the algorithms exhibit similar performance, and we would select the Kalman filter since it provides a more consistent (if at times slightly less accurate) stream of location estimates. Supplied with reliable readings, the Kalman filter performs better than regression as deployment density decreases. For deployments with poor calibration and/or less reliable, “noisy” readings (as in the Twente data), we would select the non-linear regression algorithm for its accuracy, despite the high ratios (20–80%) of rejected readings.

We have shown that the algorithms can work well on homogeneous data (pseudoranges or AOAs), despite the reduction of the information contributing to the location solution. Under certain configurations, noisy, homogeneous data can be pathological, as in the Twente pseudorange-only vertical accuracy, or the Twente AOA-only horizontal accuracy. When working on homogeneous data with reliable accuracy, the algorithms continue to produce good location estimates, as seen in the Lancaster results.

In many of our results, the vertical location accuracy was worse than the horizontal location accuracy. As noted previously, this is very much a function of the geometry of the receivers in our deployments. One can envision receiver deployment strategies which correct the bias, such as fixing some receivers high on the wall (as in our test deployments), and some receivers to the ceiling with their boresight facing the floor. The ceiling-mounted receivers’ reported AOAs would contribute more heavily to the horizontal estimate, and the pseudoranges (in combination with pseudoranges received from wall-mounted receivers) would contribute much more to the vertical estimate.

As future work, we plan to explore *adaptive* Kalman filters. This is particularly interesting as Kalman performance is heavily influenced by the choice of the model parameters, and making the filter adaptive can automatically tune the model parameters based on the current measurement. Usage of other probabilistic algorithms, for example particle filtering, may also be a fruitful area for investigation. Another avenue of research

is to extend the presented algorithms to address *online auto-calibration* of reference nodes. Called simultaneous localisation and mapping (SLAM) in robotics, this can also be used for calibration-sensitive infrastructure to bring us a step closer in realising easily deployable yet accurate positioning systems.

Acknowledgements. We are grateful to Dr Andy Ward for providing insightful comments and suggestions on this work. We are deeply indebted to Carl Fischer for setting up the camera tracking system and collecting the dynamic data.

References

1. Duff, P., Muller, H.: Autocalibration algorithm for ultrasonic location systems. In: Proceedings of the Seventh IEEE International Symposium on Wearable Computers, White Plains, NY (October 2003)
2. Fontana, R.J., Richley, E., Barney, J.: Commercialization of an ultra wideband precision asset location system. In: Proceedings of the IEEE Conference on Ultra Wideband Systems and Technologies, Reston, Virginia, USA, November 2003, pp. 369–373 (2003)
3. Fontana, R.J.: Experimental results from an ultra wideband precision geolocation system. In: Ultra-Wideband, Short-Pulse Electromagnetics, May 2000, pp. 215–224 (2000)
4. Fox, D., Hightower, J., Liao, L., Schulz, D., Borriello, G.: Bayesian filtering for location estimation. *IEEE Pervasive Computing*, 24–33 (2003)
5. Foxlin, E., Harrington, M., Pfeifer, G.: Constellation: a wide-range wireless motion-tracking system for augmented reality and virtual set applications. In: Proceedings of SIGGRAPH 1998, pp. 371–378. Addison Wesley, Reading (1998)
6. Glantz, S.A., Slinker, B.K.: Primer of Applied Regression and Analysis of Variance, 2nd edn. McGraw-Hill, New York (2000)
7. Negenborn, R.: Robot localization and Kalman filters: On finding your position in a noisy world. Master's thesis, Utrecht University (2003)
8. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical Recipes in C: The Art of Scientific Computing. Cambridge University Press, Cambridge (1992)
9. Renaudin, V., Kasser, B.M.M.: Optimal data fusion for pedestrian navigation based on UWB and MEMS. In: Proceedings of PLANS (2008)
10. Mehra, R.: On the identification of variances and adaptive Kalman filtering. *IEEE transactions on Automatic Control* AC-15(2), 175–184 (1970)
11. Scott, J., Dragovic, B.: Audio location: Accurate low-cost location sensing. In: Gellersen, H.-W., Want, R., Schmidt, A. (eds.) PERVASIVE 2005. LNCS, vol. 3468, pp. 1–18. Springer, Heidelberg (2005)
12. Smith, A., Balakrishnan, H., Goraczko, M., Priyantha, N.B.: Tracking moving devices with the Cricket location system. In: Proceedings of the Second International Conference on Mobile Systems, Applications and Services (MobiSys) (June 2004)
13. Ubisense Ltd. Location Engine config manual (2007)
14. Ward, A.: Calibration of a location system. Patent PCT/GB2007/001425 (November 2007)
15. Ward, A.M.R.: Sensor driven Computing. PhD thesis, Corpus Christi College, University of Cambridge (1998)
16. Welch, G., Bishop, G.: An introduction to the Kalman filter. Technical Report 95-041, Dept. of Computer Science, University of North Carolina (2006)
17. Welch, G., Bishop, G., Vicci, L., Brumback, S., Keller, K., Colucci, D.: The HiBall Tracker: high-performance wide-area tracking for virtual and augmented environments. In: Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST) (1999)

An Ultrasonic 3D Positioning System Using a Single Compact Receiver Unit

Masanori Sugimoto¹, Kan Tulathimutte¹, Toshio Ito¹, Tetsuya Sato¹,
and Hiromichi Hashizume²

¹ University of Tokyo, 7-3-1 Hongo, Bunkyo-ku Tokyo, 113-8656, Japan
{sugi,kantula,toshio,sato}@itl.t.u-tokyo.ac.jp
<http://www.itl.t.u-tokyo.ac.jp/>

² NII, 2-1-2 Hitotsubashi, Chiyoda-ku Tokyo, 101-8430 Japan
has@nii.ac.jp
<http://www.nii.ac.jp/>

Abstract. We present a system that can identify positions of objects in three-dimensional space using ultrasonic communications. The proposed system uses the time of flight of ultrasonic waves to calculate the distance between a transmitter and a microphone. An innovative method called the Phase Accordance Method is used for accurate detection of the arrival time of the ultrasonic signal for distance measurements. The main benefit of the proposed system is that it can locate positions of static and moving objects in a three-dimensional space by using only one compact receiver unit mounting three ultrasonic microphones; this will reduce the users' deployment tasks. Experimental results prove that the system possesses sufficient accuracy levels and stable performance of position measurements in static and dynamic situations, even though the beacon geometry of the system is poor.

1 Introduction

In the past decade, the concept of location-aware computing has gained much attention from researchers in academia and industry all over the world (e.g., [1]). One of the important core technologies that drives possibilities of location-aware computing is the positioning system. While GPS provides good outdoor location information, its accuracy degrades indoors because buildings and walls block direct signals from the GPS satellites. For this reason, many research groups have attempted to create indoor location systems by using various approaches such as Wi-Fi, vision, infrared, and ultrasonic communications [2]. Active Bat [3] and Cricket [4] are examples of noteworthy studies on ultrasonic positioning systems.

One of the disadvantages that most existing ultrasonic positioning systems have in common is that they require at least three beacons separated by some distance. This placement is related to beacon geometry issues (The term “beacon geometry” usually refers to the placement of transmitters, but in this study, it also means that of receivers). The beacons must be placed close enough together so that they are all in the range of the transmitted ultrasonic signal but

separated enough to be suitable for trilateration. Therefore, many beacons must be deployed to cover the desired positioning or tracking area while retaining the desired level of accuracy. However, it can be difficult for users to install multiple beacons accurately.

We propose an ultrasonic positioning system that can provide accurate coordinates of objects in three-dimensional (3D) space. The system requires only one pair of a transmitter and a compact receiver unit. Therefore, the system has the advantage of easy deployment, while still providing accurate position information for objects in a 3D space.

We have conducted experiments to evaluate the performance of the proposed system in both static situations and dynamic situations in which a transmitter on a robot is tracked by using the proposed single receiver unit. The results prove that the system displays a sufficient accuracy level and stable performance of position measurements in static and dynamic situations, even though the beacon geometry of the system is poor.

This paper is organized as follows. In the next section, existing ultrasonic positioning systems are explained to clarify differences from the proposed system. In Section 3, features and implementation issues of the proposed system are described. Then, experiments conducted using the system in static and dynamic settings are discussed. Finally, conclusions are given and future work described.

2 Related Work

Several ultrasonic positioning systems have been proposed by a group at the University of Bristol [5][6][7]. These systems only use ultrasonic communications for the time-of-flight (TOF) distance measurement—they do not use RF communications, to reduce the energy consumption and make their ultrasonic receiver as small as possible. In [5], a receiver correctly identifies periodic patterns of received ultrasonic signals from multiple transmitters by using a particle filter-based method. In [6], four transmitters transmit ultrasonic packets sequentially in a given interval. The receiver’s position is estimated by identifying the correct interval time using the Extended Kalman Filter. In [7], an automatic calibration method for 3D positioning systems is proposed. This method allows nonexpert users to identify receivers’ positions simply by walking with a transmitter in the tracking space. The Dolphin system [8] developed by a group at the University of Tokyo proposes a novel distributed algorithm to automatically determine positions of objects with minimal manual configuration. Dolphin guides a user to conduct manual preconfigurations of only a few reference objects; it then estimates the positions of all the other objects by using a recursive positioning algorithm.

As for the improved accuracy of distance measurements, a system developed by a group at the University of Cambridge, also called Dolphin [9], proposes an innovative approach by using broadband ultrasonic communications. This improves the level of noise robustness and increases update rates of distance measurements by accepting signals from multiple transmitters simultaneously at a receiver.

In Constellation [10], a grid array of multiple ultrasonic transponders on a ceiling emits ultrasonic signals to three ultrasonic microphones attached to a head mounted display to identify its position and pose. Constellation also implements inertial sensors (gyros) to increase the level of accuracy, and is used for virtual reality applications. Recently, a motion capture system using ultrasonic TOF and inertial measurements has been devised and evaluated in everyday environments [11]. Because the device that mounts ultrasonic and inertial sensors is sufficiently small and lightweight, it is possible to attach the device to a person for motion recognition. The remarkable difference between these predecessors and the system proposed in this paper is that our system does not use three or more spatially distributed beacons for trilateration in a 3D space, but uses a single compact receiver unit. Therefore, users are required to install only one unit, which will reduce the deployment task, while our system maintains a sufficient level of accuracy.

3 Proposed 3D Positioning System

3.1 Distance Measurement Using Time-of-Arrival (TOA) and the Phase Accordance Method

The time difference of arrival between radio and ultrasonic signals after transmitting both of them at the same time can be used in distance measurements between a transmitter and a receiver. Because the radio propagation time is extremely short, its propagation delay can be ignored compared with the ultrasonic propagation time. The time difference of arrival between the signals is then converted into the distance between the transmitter and the receiver as shown in Figure 1.

To achieve accurate distance measurements, a method for accurately detecting the arrival time of an ultrasonic signal is required. Therefore, an innovative technique called the Phase Accordance Method (PAM) has been devised by our group [12]. In PAM, a special burst pulse that consists of two ultrasonic waves of different frequencies is transmitted, creating a beat wave called the “sync

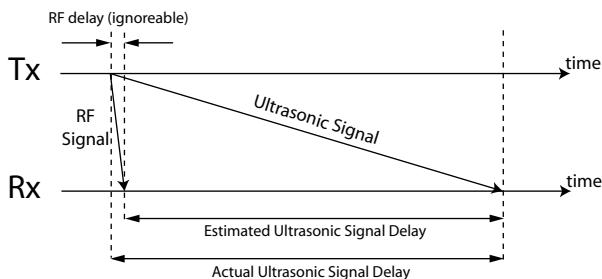


Fig. 1. Time difference of arrival between radio and ultrasonic signals

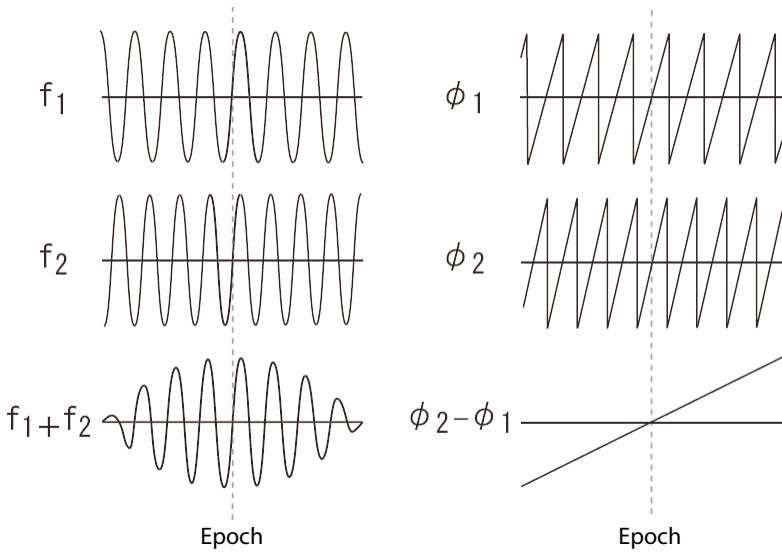


Fig. 2. Sync pattern and its epoch

pattern”, as shown in Figure 2. At the receiver side, the time when the two wave phases of the sync pattern become equal (i.e., $\phi_2 - \phi_1 = 0$) is used to indicate the arrival time of the transmitted signal. This time point is called the “epoch”. In the current implementation, the frequencies ($f_1 = 39.75$ kHz and $f_2 = 40.25$ kHz) are used to generate the sync pattern, which lasts 2 ms.

Experimental results in [12] show that PAM can estimate the distance between a transmitter and a receiver within 1 mm error in a 3 m distance measurement in a 2D space.

3.2 Finding the Position in 3D

In the proposed system, the position of an object in a 3D space can be obtained by using three ultrasonic microphones in the receiver unit, which figures out an intersecting point of three spheres. The radius of each sphere represents the distance measured between the transmitter and the microphone. Ideally, the spheres would intersect at exactly one point that defines the position of the object. However, in reality, there are always some errors in distance measurements. Therefore, some statistical approaches, for instance, the least squares method described below, are applied to obtain the true position of the object in a 3D space.

Suppose that a receiver unit is fixed at a known position; the coordinates of the transmitter should then satisfy the following equation:

$$(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2 = r_i^2,$$

where (x_0, y_0, z_0) are the coordinates of the object (ultrasonic transmitter), (x_i, y_i, z_i) are the coordinates of the i^{th} receiver microphone, r_i is the distance from the transmitter to the i^{th} microphone ($1 \leq i \leq k, 3 \leq k$, and in this study, $k = 3$). The linearization to remove the quadratic terms is conducted as follows.

$$2x_0(x_k - x_i) + 2y_0(y_k - y_i)^2 + 2z_0(z_k - z_i) = r_i^2 - r_k^2 - x_i^2 - y_i^2 - z_i^2 + x_k^2 + y_k^2 + z_k^2.$$

Then, $Ax = b$ is obtained, where:

$$A = \begin{pmatrix} 2(x_k - x_1) & 2(y_k - y_1) & 2(z_k - z_1) \\ 2(x_k - x_2) & 2(y_k - y_2) & 2(z_k - z_2) \\ \dots & \dots & \dots \\ 2(x_k - x_{k-1}) & 2(y_k - y_{k-1}) & 2(z_k - z_{k-1}) \end{pmatrix}, x = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix}$$

and

$$b = \begin{pmatrix} r_1^2 - r_k^2 - x_1^2 - y_1^2 - z_1^2 + x_k^2 + y_k^2 + z_k^2 \\ r_2^2 - r_k^2 - x_2^2 - y_2^2 - z_2^2 + x_k^2 + y_k^2 + z_k^2 \\ \dots \\ r_{k-1}^2 - r_k^2 - x_{k-1}^2 - y_{k-1}^2 - z_{k-1}^2 + x_k^2 + y_k^2 + z_k^2 \end{pmatrix}$$

The coordinates of the object are found by:

$$x = A^+ b, \quad (1)$$

where A^+ is the Moore–Penrose pseudo-inverse of Matrix A [13]. This is the most common algorithm to estimate a 3D position from distance measurements.

3.3 Beacon Geometry

Beacon placement strongly affects the accuracy of positioning results from equation (1). To achieve a sufficient level of positioning accuracy, existing systems rely on their good beacon geometry; beacons must be widely separated to create large angles between the object and the beacons, as shown in Figure 3. However,

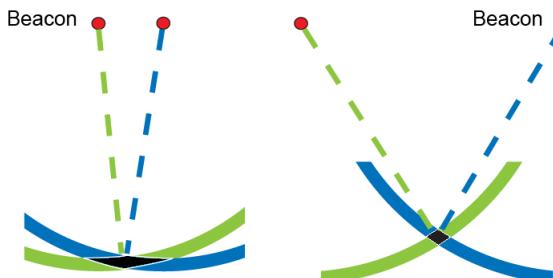


Fig. 3. Examples of a dense alignment of beacons (left) and a sparse alignment of beacons (right)

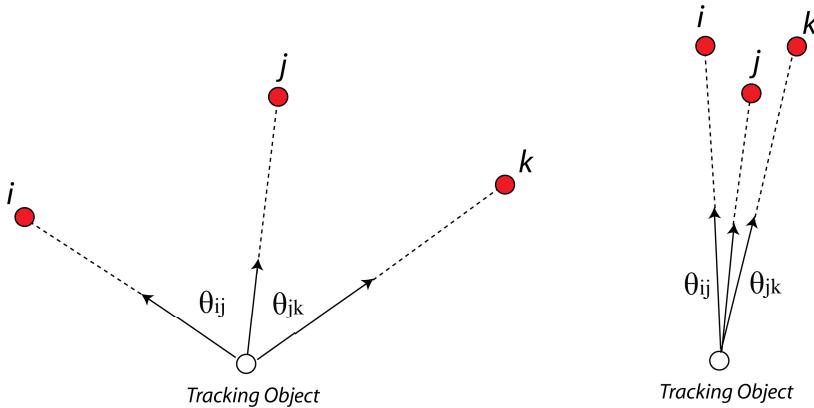


Fig. 4. Better beacon geometry (left) and worse geometry (right) for accurate positioning. Larger values of the angle θ give better beacon geometry, but the beacons must be more widely distributed.

this forces the user to perform a difficult task to make the system available, because many beacons must be calibrated and deployed to cover a space for object positioning and tracking (Figure 4).

The beacon geometry is evaluated by the Dilution of Precision (DOP) value, which is used to explain the quality of the GPS localization [14]. The smaller the DOP value, the better the position estimate. According to [15], the beacon geometry is evaluated as “ideal” (DOP value: 1), “excellent” (2–3), “good” (4–6), “moderate” (7–8), “fair” (9–20), or “poor” (21–50). In this study, three ultrasonic microphones are densely mounted in a single receiver unit. This, of course, means that the proposed system has a poor (large) DOP value. However, while using only one pair of a transmitter and a receiver unit, the proposed system still maintains a sufficient level of accuracy, comparable to previously proposed trilateration-based systems that have good (small) DOP values. The beacon geometry issue related to the proposed system is discussed in more detail in Section 4.

3.4 System Implementation

Figure 5 shows the current implementation of the proposed 3D position system. The receiver unit contains three ultrasonic microphones and has dimensions 100 mm \times 100 mm \times 80 mm. The three microphones form an isosceles triangle with sides of 80 mm, 80 mm, and 89.4 mm. Each microphone is connected to a signal processing board that is used to calculate the epoch in PAM. Both the receiver unit and a transmitter are individually connected to an RF module that is used for time synchronization and measurement triggering, as discussed in Section 3.1.

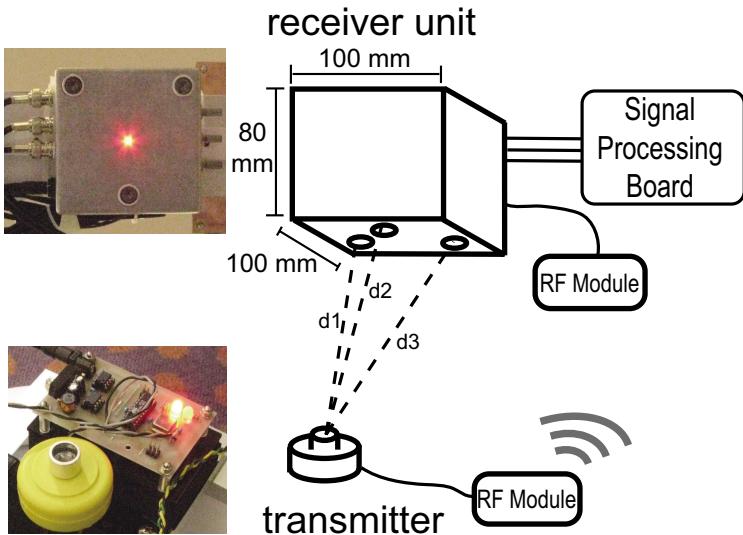


Fig. 5. System overview

4 Experiments and Evaluations

To investigate the performance of the proposed system, we conducted experiments in both static and dynamic settings by using a mobile robot tracking system. In this system, the mobile robot with an ultrasonic transmitter attached can move freely inside the tracking area. An ultrasonic receiver unit attached to the ceiling is used to receive ultrasonic signals from the transmitter on the robot and then calculates its position in a 3D space. To compare the performance of the proposed method with the ground truth, a vision-based positioning method that has a higher level of accuracy is used as a reference.

4.1 Experimental Setting

The experimental setting is shown in Figure 6. The tracking area was $1.8 \text{ m} \times 1.8 \text{ m}$. The ultrasonic receiver unit was mounted on the ceiling at a height of 2.5 meters. A high-resolution camera (Prosilica EC1600, 1600×1200 pixels) that was used for vision-based positioning was installed next to the ultrasonic receiver unit to obtain optical reference measurements. Calibration tasks using a checkered pattern was conducted to reduce position errors caused by the distortion of the camera lens. The resolution of the camera in the tracking area is 1.55 mm/pixel . A commercial mobile robot called iRobot was used as a tracking object, and an ultrasonic transmitter was mounted on it. An RF module (Crossbow Technology, MICAz), which was used for time synchronization and measurement triggering, was attached to the ultrasonic receiver unit, the

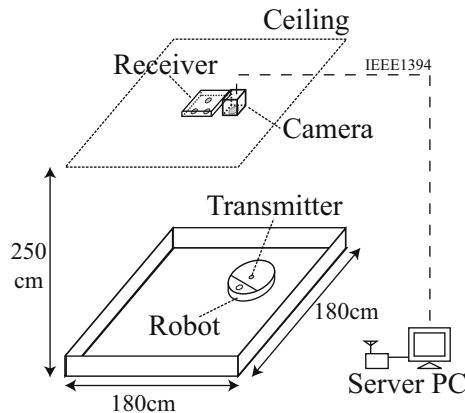


Fig. 6. Experimental setting

transmitter, and a server PC. When the receiver unit receives the sync pattern signal from the transmitter on the robot, the signal processing board connected to the unit (Figure 5) calculates the distances to the robot from the three microphones to identify its position. Because of the limited computational capability of the microprocessor unit (Renesas Technology Corp., SH-2, 28 MHz) on the current version of the board, the frequency of ultrasonic measurements was set to 10 per second. The position information of the robot was then forwarded to the PC server for visualization (moving trajectories of the robot) and further data analyses.

The tracking area in this experiment was limited to $1.8\text{ m} \times 1.8\text{ m}$ because of the signal reception angle of the ultrasonic microphones (Figure 7). The off-the-shelf cheap ultrasonic sensors (transmitter and microphone: Nihon Ceramic T4016A1 and R4016A, cost about six US dollars) used in this experiment have a beam angle of approximately 60 degrees, which allows strong directivity. As shown in Figure 7, signals received from larger angles are degraded and should not be used.

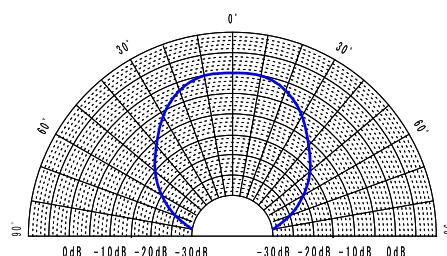


Fig. 7. Directivity of the ultrasonic microphone used in the experiment

4.2 Evaluation Method

By comparing the positioning results obtained by the proposed system with those from the vision-based positioning system, the performance of the proposed system was assessed. In the vision-based positioning system, a camera connected to the PC captures an image of the infrared (IR) LEDs mounted on the robot. The captured image is then processed by the PC to determine the actual coordinates of the robot from the position of the LEDs in the image. To make the identification of the LEDs easier, the camera was fitted with an IR filter in front of its lens so that it captured only the IR LEDs of the robot.

The time synchronization between the vision-based and ultrasonic positioning systems was established by using trigger packets. When the RF module attached to the PC server receives these packets from the transmitter, it orders the PC to capture a frame immediately from the camera. As discussed in Section 4.4, this synchronization does not always guarantee that the position estimates of the robot are simultaneously carried out by the vision-based and ultrasonic positioning systems.

The accuracy of the proposed system was assessed using Euclidean distance errors using the following formula: $\text{error} = \sqrt{(x_m - x_0)^2 + (y_m - y_0)^2 + (z_m - z_0)^2}$, where (x_0, y_0, z_0) is the robot's reference coordinates given by the vision-based positioning system and (x_m, y_m, z_m) is its measured coordinates given by the proposed ultrasonic positioning system.

4.3 Experiment: Static Setting

For the static setting, the robot was placed at the center and each corner of the tracking area as shown in Figure 8. The real positions of the robot were measured physically using a measuring tape. The robot was set to the stationary mode. Two thousand location samples were taken for each position.

The results of this experiment are shown in Table 1. It can be seen that the result taken at P0 shows better accuracy than those taken at other points. The reason seems to be that at the corners of the tracking area, the angle from the transmitter to the receiver unit is larger, and the distance between them is also greater. As shown in Figure 7, the sensitivity of the received signal degrades if the incident angle to the transmitter is larger.

The average position error ranges from 4.14 mm (P0) to 23.65 mm (P4). A strength of our system, confirmed through the experiment, is that the standard deviation (SD) at all points is less than 10 mm. This means that the proposed system can perform stable position recognition, and more accurate positioning may be possible through calibrations.

4.4 Experiment: Dynamic Setting

The experimental setting is the same as the static setting except that the robot moved freely in the tracking area. The speed of the robot was set to 140 mm/s.

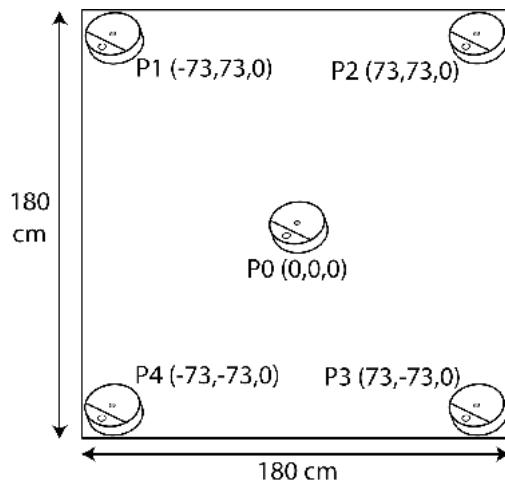


Fig. 8. Robot positioning experiments in static settings

Table 1. Experimental data for static settings

Point	Real position (mm)	Average measured position (mm)	Average error (mm)	Standard deviation (mm)
P0	(0.0, 0.0, 0.0)	(3.08, 2.76 , 0.23)	4.14	3.53
P1	(-730.0, 730.0, 0.0)	(-737.74, 751.00, -3.62)	22.67	8.21
P2	(730.0, 730.0, 0.0)	(743.50, 746.38, -2.54)	21.53	9.25
P3	(730.0, -730.0, 0.0)	(748.04, -732.12, -1.48)	18.22	8.49
P4	(-730.0, -730.0, 0.0)	(-752.53, 736.38, -3.32)	23.65	7.66

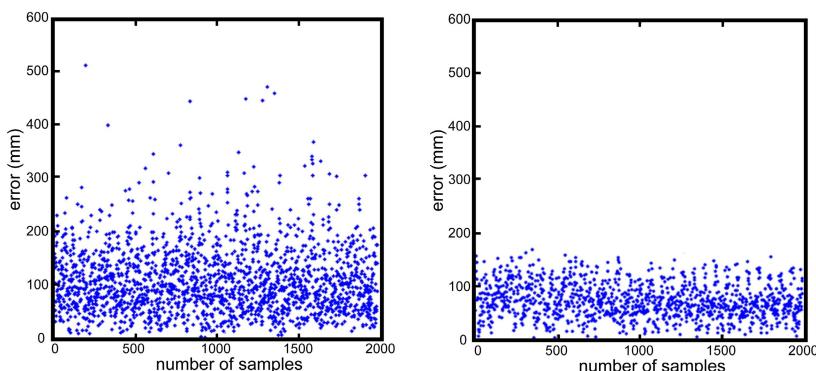


Fig. 9. Dynamic robot tracking without (left) and with (right) an averaging filter

Table 2. Comparison of the results with and without averaging filter

Experiment	Average position error (mm)	Position error SD (mm)	Max position error (mm)
Without filter	102.22	65.84	515.65
With filter	52.77	30.96	176.88

While the robot was moving, location samples were taken for about 30 minutes. The location data gained were evaluated with and without an averaging filter that reduces the effect of noises. The averaging filter uses 10 distance samples from each microphone in 1 second (1 Hz), and then simply calculates their average.

The resulting graphs obtained through the dynamic tracking experiment with and without the averaging filter are shown in Figure 9. A summary of the results is given in Table 2. From the table, it can been seen that the proposed system itself can identify positions of the moving robot with an error level of approximately 100 mm, which seems useful for many location-aware computing applications, although the maximum error is too large. However, applying the averaging filter to the proposed system can significantly improve its performance: The average position error was reduced to 51.6 percent of its original value, the position error to 47.0 percent, and the maximum position error to 34.3 percent. The averaging filter is therefore useful for the proposed system in the dynamic setting.

Further analysis of the results shown in Figure 10 is summarized in Table 3, where SDs of position errors in the three axes are shown. The X and Y axes have higher SDs than the Z axis. Figure 3 in Section 2 may explain why such results are obtained: when the microphones are densely arranged on the X-Y plane, position errors of the transmission become larger in the X and Y axes than in the Z axis.

However, despite the close alignment of the ultrasonic microphones in the receiver unit, the proposed system achieves a sufficient level of accuracy: the average position error is 52.77 mm and the maximum error is less than 180 mm by averaging 10 sample points from each microphone. This is considered useful for many applications such as automated robot navigation.

One of the error sources that might affect the result of the experiment is the temperature variation during the experiment period. The speed of ultrasound varies when the temperature changes ($v = 331.5 + 0.61T$, v : sound velocity (m/s), T : centigrade temperature). Another possible error source is the latency in the vision-based positioning system. When the RF module accepts a trigger packet

Table 3. Standard deviations of position estimates in each axis

Axis	X (mm)	Y (mm)	Z (mm)
SD of position error	38.97	36.68	12.78

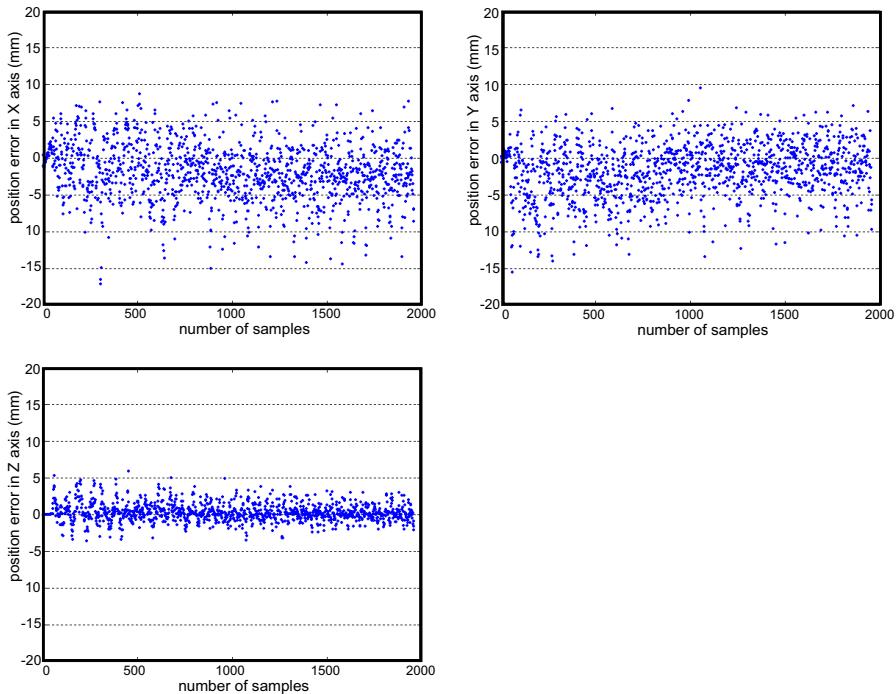


Fig. 10. Position errors in the X, Y, and Z axes

from the transmitter, it immediately notifies the PC through an RS232C interface. The PC then issues an image capture command to the camera. However, the latency in the operating system of the PC that manages the RS232C communication and the camera driver may be as great as 0.1 s. As the speed of the robot is 140 mm/s, it moves 14 mm during the latency. Therefore, this latency reduces the reliability of the vision-based positioning system as the ground truth. Further possible error sources are the Doppler effect caused by the moving robot and the phase characteristics of the microphone. The phase characteristics, as well as the sensitivity level (amplitude) of the microphone, as shown in Figure 7, depend on the frequency and direction of the transmitted signals and may degrade the performance of distance measurements by PAM.

Finally, we present comparisons between the proposed system and existing systems based on beacon geometry. The metrics Position Dilution of Precision (PDOP), Horizontal Dilution of Precision (HDOP), and Vertical Dilution of Precision (VDOP) are used. In GPS, coordinates of four satellites and receiver clock bias are usually used for calculating DOP values. In this study, those of three microphones are used and the clock bias is not considered for the calculation. However, it retains the geometric meaning of PDOP, VDOP, and HDOP values as shown in [14]. According to the paper on Active Bat [3], beacons are arranged

Table 4. DOP values of the proposed system

point	PDOP	HDOP	VDOP
P0	58.5	58.5	0.71
P1	65.7	63.1	18.2
P2	65.7	63.1	18.2
P3	66.1	63.4	18.8
P4	66.1	63.4	18.8

at 1.2 meter intervals. If Active Bat was installed in the experimental setting of the proposed system, its PDOP values would range from 4.23 to 4.73 at the five points in Figure 6, HDOP values from 4.18 to 4.58, and VDOP values from 0.65 to 1.18, all of which would be much smaller than the values of the proposed system, as shown in Table 4. The remarkable differences in these values illustrate how poor the beacon geometry of the proposed system is, but nevertheless the system retains an acceptable level of positioning accuracy. Table 4 also explains why the position errors of the proposed system in the Z axis are smaller than those in the X and Y axes, as the HDOP values related to the X and Y axes are much larger than the VDOP value related to the Z axis.

5 Conclusions and Future Work

We have described an ultrasonic 3D positioning system. The main advantage of the proposed system is that it can identify positions of static and moving objects in a three-dimensional space by using only one compact receiver unit mounting three ultrasonic microphones, which will reduce users' deployment tasks. Experimental results prove that the system shows a sufficient accuracy level and stable performance of position measurements in static and dynamic situations, even though the beacon geometry of the system is poor.

One task for the future is the application of different filtering methods. The averaging filter is proved to be useful for reducing the effect of noise. However, it introduces a small delay in reporting position information because it must wait for a number of samples to be collected before calculating their average. For a fast-moving object, this delay may cause large errors in a real-time application. When an object moves at a faster speed, the Doppler effect must be taken into account. Therefore, Doppler effect compensation should be included in the proposed system, to achieve an accurate fast-moving object tracking system.

In this paper, we used the least squares method for 3D position estimates. It seemed to perform well in the experiment. However, in a realistic environment where noisier and multipathed signals are received, the method does not always work [16]. Therefore, we plan to evaluate different estimation methods that are suitable for our system by deploying more receiver units inside a building to cover larger tracking areas.

References

1. Moran, T., Dourish, P.: Introduction to This Special Issue on Context-Aware Computing. *Journal of Human-Computer Interaction* 16(2–4), 87–95 (2001)
2. Hazas, M., Scott, J., Krumm, J.: Location-aware Computing Comes of Age. *IEEE Computer* 37(2), 95–97 (2004)
3. Ward, A., Jones, A., Hopper, A.: A New Location Technique for the Active Office. *IEEE Personal Communications* 4(5), 43–47 (1997)
4. Smith, A., Balakrishnan, H., Goraczko, M., Priyantha, N.: Tracking Moving Devices with the Cricket Location System. In: Proceedings of MOBISYS 2004, Boston, MA, pp. 190–202 (2004)
5. Muller, H., McCarthy, M., Randell, C.: Particle Filters for Position Sensing with Asynchronous Ultrasonic Beacons. In: Hazas, M., Krumm, J., Strang, T. (eds.) LoCA 2006. LNCS, vol. 3987, pp. 1–13. Springer, Heidelberg (2006)
6. McCarthy, M., Muller, H.: RF Free Ultrasonic Positioning. In: Proceedings of IEEE ISWC 2003, White Plains, NY, pp. 79–85 (2003)
7. Duff, P., Muller, H.: A New Method for Auto-Calibrated Object Tracking. In: Beigl, M., Intille, S.S., Rekimoto, J., Tokuda, H. (eds.) UbiComp 2005. LNCS, vol. 3660, pp. 123–140. Springer, Heidelberg (2005)
8. Minami, M., Fukuiji, Y., Hirasawa, K., Yokoyama, S., Mizumachi, M., Morikawa, H., Aoyama, T.: DOLPHIN: A Practical Approach for Implementing a Fully Distributed Indoor Ultrasonic Positioning System. In: Davies, N., Mynatt, E.D., Sio, I. (eds.) UbiComp 2004. LNCS, vol. 3205, pp. 347–365. Springer, Heidelberg (2004)
9. Hazas, M., Hopper, A.: Broadband Ultrasonic Location Systems for Improved Indoor Positioning. *IEEE Transactions on Mobile Computing* 5(5), 536–547 (2006)
10. Foxlin, E., Harrington, M., Pfeifer, G.: ConstellationTM: A Wide-Range Wireless Motion-Tracking System for Augmented Reality and Virtual Set Applications. In: Proceedings of ACM SIGGRAPH 1998, Orlando, FL, pp. 371–378 (1998)
11. Vlasic, V., Adelsberger, R., Vannucci, G., Barnwell, J., Gross, M., Matusik, W., Popovic, J.: Practical Motion Capture in Everyday Surroundings. *ACM Transactions on Computer Graphics* 26(3) (2007)
12. Hashizume, H., Kaneko, A., Sugano, Y., Yatani, K., Sugimoto, M.: Fast and Accurate Positioning Technique Using Ultrasonic Phase Accordance Method. In: Proceedings of IEEE TENCON 2005, Melbourne, Australia, pp. 826–831 (2005)
13. Golub, G., Van Loan, F.: Matrix Computations, 3rd edn. Johns Hopkins University Press, Baltimore (1996)
14. Misra, P., Burke, B.P., Pratt, M.M.: GPS performance in Navigation. *Proceedings of IEEE* 87(1), 65–85 (1999)
15. Lee, Y., Suh, Y., Shibasaki, R.: Ajax GIS Application for GNSS Availability Simulation. *KSCE Journal of Civil Engineering* 11(6), 303–310 (2007)
16. Rice, A., Harle, R.: Evaluating Lateration-Based Positioning Algorithms for Fine-Grained Tracking. In: Proceedings of DIALM-POMC 2005, Cologne, Germany, pp. 54–61 (2005)

LOCK: A Highly Accurate, Easy-to-Use Location-Based Access Control System

Yongcai Wang, Junhui Zhao, and Toshikazu Fukushima

NEC Labs. Beijing, China, 100084

{wangyongcai,zhaojunhui,fukushima}@research.nec.com.cn

Abstract. With proliferation of ubiquitous computing, digital access is facing an increasing risk since unauthorized client located at any place may intrude a local server. Location Based Access Control (LBAC) is a promising solution that tries to protect the client's access within some user-defined secure zones. Although a lot of prior work has focused on LBAC, most of them suffer from coarse judgment resolution problem or considerable manual setting-up efforts. This paper proposes LOCK, a highly accurate, easy-to-use LBAC system, which uses autonomous ultrasound positioning devices and an access control engine to precisely characterize the secure zones and accurately judge the online access authority. Particularly, the ultrasound positioning device provides relative 3D coordinate of the mobile clients. Measurement-Free Calibration (MFC) is proposed to easily calibrate these positioning devices to transform their relative positioning results into an absolute coordinate system. In this coordinate system, secure zones are characterized by a Coherent Secure Zone Fitting (CSZF) method to compensate the disparity between manually measured secure zone and the secure zone seen by the positioning devices. Furthermore, a Round-Trip Judgment (RTJ) algorithm is designed to fast online determine the geographical relationship between the client's position and such secure zones. A prototype of LOCK system was implemented by defining a meeting table as secure zone to control the client's access to a FTP server. Experiment results show that the system can be easily set up and can control the client's access with centimeter level judgment resolution.

Keywords: Location-Based Access Control, Ultrasound Positioning Device, Measurement-free, Highly accurate, Easy-to-use.

1 Introduction

With the increasing growth of networks, sensors and mobile devices, we are moving towards an era of ubiquitous computing, where data can be shared and accessed conveniently through ubiquitous digital links. However, digital access is vulnerable since unauthorized client hiding at anywhere of the world may launch attack to a server in office. To address this problem, Location Based Access Control (LBAC) has brought out a significant amount of research efforts, which allows taking user's physical location into account when determining the

user's access privileges. There are a wide variety of applications related to LBAC. For example, many companies desire location-based security policies, such as employees are required to access the confidential information only at their group's working area, and visitors or interns may not be allowed to access in certain areas. A typical application scenario can be depicted as: a server containing top secret information can be placed in a secure zone, outside of which, any access will be denied. In such scenarios, we wish to use the location of the client as a "physical key". It can work together with other authentication mechanisms to enhance the security level of data access. To realize such a "physical key", some fundamental requirements are posed from applications.

The first necessity is to guarantee the correctness of the LBAC, which indeed depends on two parties: 1).The accuracy of the positioning results. 2).The precise definition of the secure zone . Highly accurate positioning is always pursued by LBAC studies by densely deployment of sensors, or positioning algorithm improvements[2]. Secure zones are commonly defined by off-line training in signal space[3] or manually characterization. The second necessity of LBAC system is that it should be easy-to-use. Because the positioning system calibration, the secure zone characterization in the 3-D space may involve many manually measurements efforts. Although many prior studies have focused on LBAC, to our best knowledge, most of them cannot provide fine-grained judgment resolution or need considerable manual setting-up efforts.

Inspired by above requirements, we proposed LOCK, a highly accurate, easy-to-use LBAC system, which is designed in following ways to provide fine-grained judging resolution and to be easy for use. In off-line phase, we deploy autonomous ultrasound positioning devices to locate mobile clients. Every positioning device can be easily installed. It monitors the accurate relative 3D coordinates of the mobile clients with respect to itself. A Measurement-Free Calibration method (MFC) is proposed to automatically calibrate the positioning devices to transform their relative positioning results into an absolute coordinate system. So that, the LOCK system can be easily deployed, easily calibrated, and the positions of the mobile clients are accurately monitored. Then, secure zones are defined to characterize where the information access can be allowed. We observed that if the secure zones are manually measured, there is disparity between the user desired secure zone and the secure zone seen by the positioning devices, which is an important cause of false access judgement. We propose a Coherent Secure Zone Fitting (CSFZ) method to precisely characterize the secure zones by continuously positioning on the boundary points of the secure zone and polygon fitting. CSFZ can offset many error factors because the secure zone characterization process is subjected to the same noises as the online client locating process.

In online phase, when a client wants to access the location sensitive information, the client's real-time position is checked against the off-line defined secure zones to judge its authority. We proposed a Round-Trip Judgment (RTJ) algorithm to fast online determine the geographical relationship between the client's real-time position and the secure zones. The basic idea is that if a client is within a secure zone, when another client walks along the boundary of the secure zone,

it should make a round-trip around the first client. RTJ is in $O(n)$ complexity, making the online judgement very fast. Furthermore, a prototype of LOCK system was implemented by defining a meeting table as secure zone to control the client's access to a FTP server. Experiment results show that: 1) the proposed LOCK system can be setup without manually measurement efforts; 2) the system can well judge the physical access authorities of the mobile clients in a resolution of centimeter.

The rest of this paper is organized as: we introduce related works in Section 2. The system architecture of LOCK is presented in Section 3. The design methodologies in off-line phase are presented in Section 4. The RTJ algorithm in online phase is presented in Section 5. Prototype of LOCK system and the experimental results are presented in Section 6 and 7, respectively. Finally, Section 8 concludes the paper.

2 Related Work

2.1 Location Based Access Control

For LBAC, a general method is location attribute verification. In [5], Capkun et al. presented a position verification method, called Verifiable Multilateration. It used some reference points, to calculate the position of user by multilateration and used the location as access evidence. In [7], Marcela D. et.al. presented a hospital information access control system. It used WLAN as the locating system and used neural network for location calculation. In [8], YounSun Cho, et. al. presented a Location-Aware Access Control protocol. It coarsely defined the location area by the overlapping areas of multiple access points. A drawback of such approaches is that the locating accuracy is not good, degrading the reliability of LBAC.

Some other methods simplified the secure zones to improve the reliability of LBAC. *Secure boundary methods* were presented. In [4], Apu Kapadia, et. al. proposed Virtual Wall to protect user's digital finger print. It used Solar system as the sensing infrastructure. In [3], Pan. J.J. et. al. presented Digital Wall to train a wall in off-line phase using learning method. In online phase, user's access was allowed within the boundary and denied outside. In [9], Sastry et. al proposed an Echo protocol to prove the location claim of client by checking its round trip time of a RF signal and its echo using ultrasound. *Proximity method* used limited-range radio to restrict access within the device's neighborhood. In [10], location-limited channels were restricted to a short range and used to guarantee the authenticity of both communication endpoints. In [11], Balfanz et al. proposed using location-limited channels to bootstrap a wide range key-exchange protocols. In [12], limited-range radio broadcasts are used to verify proximity. A drawback of these approaches is that the secure zones definition is not flexible and not precise.

Comparing with above solutions, our method uses highly accurate positioning system; flexibly and precisely define the "secure zone seen by positioning system" by automatic method, which is more accurate and easier-to-use.

2.2 Location Systems and Methods

Locating methods have brought out a lot of research. Here, we focus on the indoor locating methods, and categorize these methods by signal they used.

WLAN and RF based positioning systems infer object's location by processing the radio signals, which can be summarized into two categories: (1) Radio-Propagation Models [13];(2) Empirical learning[6]. In Radio-Propagation method, the locations of reference nodes are needed and the received signal strength (RSS) is transformed into distance. Then least square or trilateration method are used to estimate the most probable coordinate [13]. Another class of location estimation systems is based on Learning-based Models, which employ machine learning techniques. Signal patterns can be captured when sufficient empirical data are manually collected at different locations [6][14]. The locations of reference points are not needed. The RADAR system [6] uses reference tags to construct a radio map. A similar technique is used in MOTETRACK[14] system to train a radio map in off-line phase, and infer target's location by matching algorithms in signal space.

Infrared based locating system use infrared signal and proximity method to infer the position of object. Active Badge [16] provides a means of locating by determining the location of their Active Badge which transmits infrared-red signal. Smart Locator [19] is a system that deployed infrared transmitters on the ceiling to transmit location ID, and the mobile terminals equipped with ID receivers received those location ID to determine their locations. A drawback of RF-based or infrared locating systems is the coarse accuracy. The positioning resolution is only 3-5 meters.

Ultrasound based positioning system provide much better accuracy. They commonly used TOA-based positioning method, which measures the propagation time of ultrasound to infer the distance between transmitter and the receiver and use multilateration or trilateration to infer object's position. Active Bat system [17] deployed a array of ultrasound receivers on the ceiling to track the position of the ultrasound transmitter (Tag). The Cricket system [18] deployed beacons which are the ultrasound receivers in the environment to navigate the mobile ultrasound transmitters. Although high accuracy can be achieved, drawback of such systems are the complex installation and calibration of the network of sensors. [20] studied an autocalibration algorithm to compute the position of ultrasonic transmitters using sets of distance readings, but their algorithm is limited the layout of ultrasound transmitters.

3 System Architecture

We proposed the architecture of the LOCK system as shown in Fig.1. The space where the LBAC will be carried out is denoted as the real space. Autonomous ultrasound positioning devices are installed to monitor the real-time positions of the mobile clients in the space. These mobile clients may move randomly and access the information at their wills. Each positioning device only calculates the relative position of the client and sends the result to a location server. An

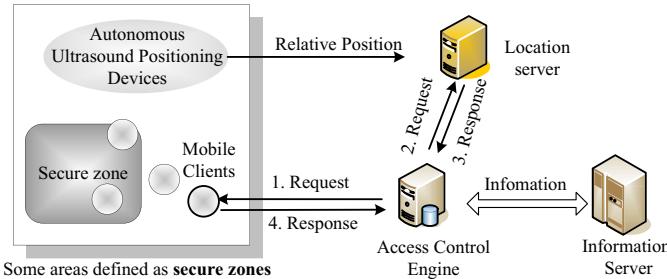


Fig. 1. System architecture of LOCK system

Access Control Engine (ACE) carries out the access control tasks. In off-line phase, the ACE calibrates the positioning devices to transform their relative positioning results into an absolute coordinate system and characterizes some secure zones. In online phase, the ACE determines the geographical relationship between the position of the client and the secure zones to make the access control decision.

The runtime work-flow of the LOCK system is as following. When a mobile client tries to access the information server, it firstly requests the ACE; ACE then requests the location server for the real-time position of the client. The location server answers the ACE with relative location data. The ACE transforms the relative position to absolute position and authenticates the client by judging the geographical relationship between its position and the predefined secure zones. If this location-based access is approved, the client will proceed its access to the information server by conventional security schemes (as password, etc.)

3.1 Access Control Engine

An overview of the working process of ACE is shown in Fig.2, containing off-line and online two phases. In off-line phase, we developed two methods to make the LOCK system accurate and easy-to-use. 1). Because the ultrasound positioning devices only provide the relative position of the mobile client, MFC(Measurement-Free Calibration) is proposed to calibrate the positioning devices for transformation of their relative positioning results to an absolute coordinate system 2). In the absolute coordinate system, CSZF (Coherent Secure Zone Fitting) is proposed to characterize the secure zones by boundary point positioning to compensate the disparity between the user desired secure zone and the secure zone seen by the positioning devices.

The second phase is the online phase, where the client's location in the absolute coordinate system is obtained and judged against the secure zones to make access control decision. The key method developed in this phase is a Round-Trip Judgement (RTJ) algorithm, which uses quadrant relation mining methods to fast determine the geographical relationship between the mobile client and the secure zones.

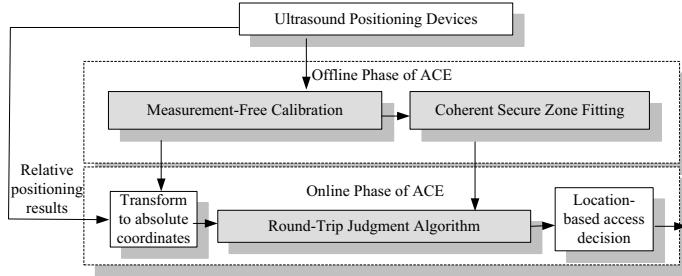


Fig. 2. Working process of ACE

3.2 Autonomous Ultrasound Positioning Device

The autonomous ultrasound positioning device in above architecture can localize mobile targets with accuracy of around 10 centimeters. The device is named as Positioning on One Device (POD), which is mounted on the ceiling of the room. POD integrates single RF and multiple ultrasound receivers into one device so that it can be easily deployed into application environment. Structural topology is designed for POD to reduce the installation and calibration efforts. A Tag actively transmits the ultrasound signal, which is commonly carried by the mobile object to be located.

Fig.3 shows the POD installation process. An initial form of POD is just like a compact Frisbee that contains single RF and multiple ultrasound receivers. While in use, the surrounding receivers can be extended to the outside so that POD has an umbrella-like appearance. Rather than deploy one by one networked sensors in the building as presented in prior art[17][18], POD can be easily installed only once. Therefore the installation efforts will be reduced tremendously.

After installation, the POD can autonomously locate the mobile Tags. It is noticeable that the POD only calculates the object's relative position in its relative coordinate system[1]. This coordinate system sets the coordinate of the POD center as the origin and sets the edge formed by the central ultrasound receiver and the first surrounding ultrasound receiver as the X-axis. For more

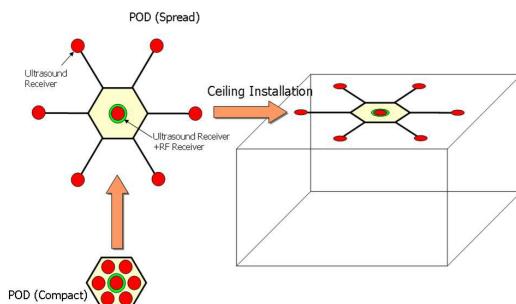


Fig. 3. Easy installation of the positioning device

information about the ultrasound positioning device, please refer to our early work[1].

4 Off-Line Setting Phase of ACE

This section presents design methodologies of the off-line phase, including MFC and CSZF.

4.1 Measurement-Free Calibration

Because different PODs may have different relative coordinate systems, it is necessary to calibrate the PODs into a global absolute coordinate system before the secure zone characterization and online locating. Calibration is required to carry out two tasks: 1). To determine the coordinates of the central ultrasound receiver, i.e., the coordinates of the relative coordinate system's origin in the absolute coordinate system. 2). To determine the angle from the the relative coordinate system's X-axis to the absolute coordinate system's X-axis, which is denoted as θ .

Traditional methods[17][18] commonly adopt manually measurements to calibrate the positioning devices and the size of the room, making the system inconvenient to use. We propose a measurement-free method to do automatic calibration by feature point positioning and least square estimation. The feature points are selected on the X-axis or Y-axis of the absolute coordinate system, which are commonly easily determined in the real space and showing features of the absolute coordinate system. In the example shown in Fig.4, an absolute coordinate system is defined, using the bottom left corner of the room as the origin and using the long edge of the room as the X-axis. The room corners are selected as the feature points. Tags are placed at these feature points to broadcast signals. The uncalibrated POD receives the signals to calculate the relative positions of these points in its own coordinate system. We proposed MFC to automatically calibrate the PODs, based on two facts: 1) Linear mapping from

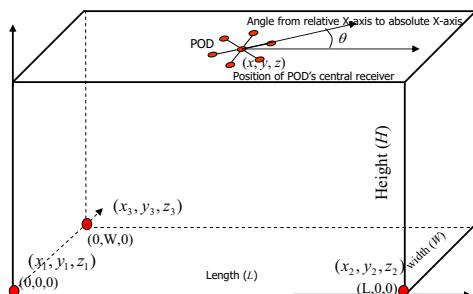


Fig. 4. Measurement-Free Calibration

the relative coordinates to the absolute coordinates. 2). Distance between two feature points is constant in the two coordinate systems.

The idea is shown in Fig.4, where a POD is installed on the ceiling. The coordinates of its central ultrasound receiver, denoted as (x, y, z) and the angle θ are to be calibrated. Three room corners are selected as feature points. (x_1, y_1, z_1) , (x_2, y_2, z_2) , (x_3, y_3, z_3) are their relative coordinates and $(0, 0, 0)$, $(L, 0, 0)$, $(0, W, 0)$ are the absolute coordinates. Equation group is built by the coordinate mapping and the distance equalities.

$$\left\{ \begin{array}{l} L = \sqrt{(x_1 - x_3)^2 + (y_1 - y_3)^2} \\ W = \sqrt{(x_2 - x_3)^2 + (y_2 - y_3)^2} \\ x_1 \cos(\theta) - y_1 \sin(\theta) + x = 0 \\ x_1 \sin(\theta) + y_1 \cos(\theta) + y = 0 \\ x_2 \cos(\theta) - y_2 \sin(\theta) + x = L \\ x_2 \sin(\theta) + y_2 \cos(\theta) + y = W \\ x_3 \cos(\theta) - y_3 \sin(\theta) + x = L \\ x_3 \sin(\theta) - y_3 \cos(\theta) + y = 0 \end{array} \right. \quad (1)$$

Solve this equation group with least square method. The unknown parameters of the room (L, W, H) , central coordinate of POD (x, y, z) and θ are derived. Basically three feature points are enough for measurement-free calibration. The calibration accuracy will be increased by selecting more feature points. For multiple POD case, uncalibrated POD can be calibrated successively by the calibrated POD. When a Tag is placed in the overlapped area of an uncalibrated POD and a calibrated POD, the uncalibrated POD gives positions in its own coordinate system, while the calibrated POD give the absolute coordinates. If two feature points in the overlapped area to develop equations by coordinate mapping, the coordinates of the uncalibrated POD's center and the angle from its first edge to the global X-axis can be derived. The calibration accuracy will be increased if more feature points in the overlapped area are selected.

Hence, after calibration of the PODs by MFC, the relative positioning results given by the PODs can be linearly mapped to the absolute coordinate system.

4.2 Coherent Secure Zone Fitting

After positioning device calibration, secure zones can be characterized in the absolute coordinate system for location-based access control.

Conventional methods commonly manually measured the boundary of the secure zones and stored them for online use. But 1) manual measurement is not user-friendly; 2) we observed that there is disparity between the manually characterized secure zone and the secure zone seen by the positioning devices. An explanation is shown in Fig.5. Suppose the rectangle is the user desired secure zone. User can carefully measure the coordinates of the vertexes of the rectangle in the absolute coordinate system to characterize it. However, because the environmental noises and system error, etc., in online locating phase, when a Tag moves along the boundary of the rectangle, the track may be the dashed

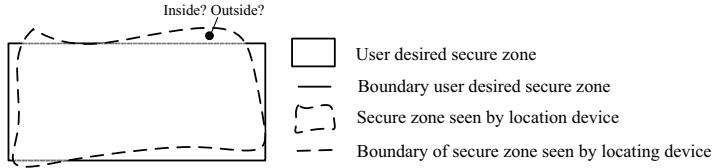


Fig. 5. The disparity of manually measured secure zone and secure zone seen by positioning devices

line shown in the Fig.5. This is the “secure zone seen by the positioning device”. This disparity will cause false judgement in the LBAC. Paper [21] proposed a method to construct 3D inferences by making a few sounds at the desired location in an audio locating system, but their 3D inferences are very rough.

To precisely characterize the secure zones, we propose a Coherent Secure Zone Fitting (CSZF) method to compensate the disparity. The basic idea is to characterize the secure zones by continuously positioning at the boundary points of the secure zones and fit the secure zone by a polygon. CSZF can offset many error factors because the secure zone calibration process is subjected to the same noises as the online client locating process. Therefore, the secure zones are more precise characterized than the manual measurement method.

To carry out CSZF, firstly, a user moves a Tag along the boundary of the desired secure zone in clockwise direction. The Tag is continuously tracked along the boundary. After a round-trip, a sequence of positioning results of the boundary points is collected. This sequence may contain hundreds or thousands of positioning results. If these points in the sequence are directly stored as vertexes to define a secure zone, the efficiency of online geographical relationship judgement will be low. We propose a polygon fitting method to precisely fit the secure zone.

In order to use polygon to fit the secure zone, the collected position sequence will be divided into subsequences, with expectation that each subsequence contains exactly the points on one edge of the polygon. A slope-based subsequence division algorithm is proposed to conduct 2D secure zone characterization. Fig.4.2 illustrated the idea. Every two adjacent points in the sequence

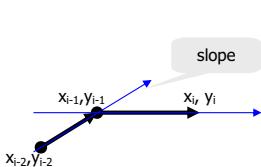


Fig. 6. Slope comparison

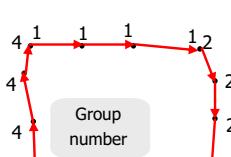


Fig. 7. Subsequence division

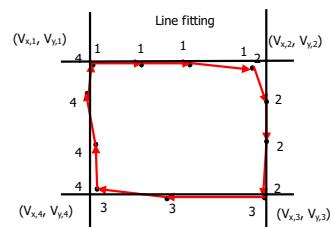


Fig. 8. Line-fitting-based vertex determination

determines a line with slope $\frac{y_{i+1}-y_i}{x_{i+1}-x_i}$. Start from the first two points (x_1, y_1) and (x_2, y_2) in the sequence. They determine the first edge with slope $s_1 = \frac{y_2-y_1}{x_2-x_1}$ and generate the first subsequence. Then (x_3, y_3) is checked by $\|\frac{y_3-y_2}{x_3-x_2} - s_1\| < H$, where H is a small threshold. If the inequality is true, (x_3, y_3) is added to the same subsequence of (x_1, y_1) and (x_2, y_2) . Otherwise a new subsequence indicating a new edge is generated by (x_3, y_3) . The slope of this new subsequence is $s_2 = \frac{y_3-y_2}{x_3-x_2}$. Repeat the process to (x_4, y_4) until the last point. N subsequence will be divided indicating N edges of a polygon. An illustration of the result is shown in Fig.7.

After N subsequences are obtained, line fitting will be applied on each subsequence to determine the vertexes of the polygon. The line fitted by the i th subsequence is denoted as $y = k_i x + z_i$. Then the coordinates of the vertexes of the polygon can be evaluated as following.

If $i==1$

$$\begin{cases} V_{x,1} = \frac{z_N-z_1}{k_1-k_N} \\ V_{y,1} = \frac{k_1z_N-k_Nz_1}{k_1-k_N} \end{cases} \quad (2)$$

Else

$$\begin{cases} V_{x,i} = \frac{z_{i+1}-z_i}{k_i-k_{i+1}} \\ V_{y,i} = \frac{k_iz_{i+1}-k_{i+1}z_i}{k_i-k_{i+1}} \end{cases} \quad (3)$$

Thus the secure zone is easily and precisely determined as a polygon which is characterized by these vertexes. Fig.8 illustrates an example of line fitting based vertex determination.

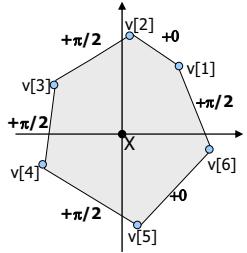
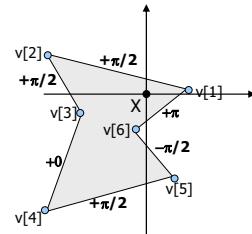
5 Online Judgement by RTJ

After above off-line phase, the positioning devices are calibrated and the secure zones are characterized in the absolute coordinate system. The secure zones are represented by vector of polygon vertexes. For example, a polygon P can be represented by $P = \{v[1], v[2], v[3], v[4]\dots\}$.

In online phase, suppose the real-time position of the client is X . The problem to determine the geographical relationship between X and the secure zone, i.e., to determine whether X is within polygon P or not. We propose a Round-Trip Judgement (RTJ) algorithm to do fast judgement. The basic idea is that if the point X is inside P , when another client walks along the boundary of P , it should make a round-trip around X .

The algorithm is carried out by firstly setting a coordinate system using X as the origin. The truth behind RTJ algorithm is that if X is inside P , the trip starting from a vertex and ending at the same vertex should travel the four quadrants of the coordinate system for exactly once. Therefore, denote $v[i]$ as the current position of the trip and $v[i+1]$ the next vertex on the trip. To formulate the algorithm, four quadrant relations are defined:

- Same: if $v[i+1]$ and $v[i]$ are in the same quadrant.
- Next: if $v[i+1]$ is in the anticlockwise adjacent quadrant of $v[i]$.

**Fig. 9.** RTJ applied on convex polygon**Fig. 10.** RTJ applied on concave polygon

- Previous: if $v[i+1]$ is in the clockwise adjacent quadrant of $v[i]$.
- Diagonal: if $v[i+1]$ is in the diagonal quadrant of $v[i]$.

A round-trip judger: $Angle$ is initialized as zero. It changes along the round-trip according to the quadrant relations:

- $Angle + 0$, if $v[i]$ and $v[i+1]$ are in the same quadrant;
- $Angle + \pi/2$, if $v[i+1]$ in the next quadrant of $v[i]$;
- $Angle - \pi/2$, if $v[i+1]$ in the previous quadrant of $v[i]$;
- $Angle + \pi$, if $v[i+1]$ and $v[i]$ are diagonal, and $(y_{i+1} * x_i - x_{i+1} * y_i) > 0$;
- $Angle - \pi$, if $v[i+1]$ and $v[i]$ are diagonal, and $(y_{i+1} * x_i - x_{i+1} * y_i) < 0$;
- $Angle + 0$, if $v[i+1]$ and $v[i]$ are diagonal, and $(y_{i+1} * x_i - x_{i+1} * y_i) = 0$;

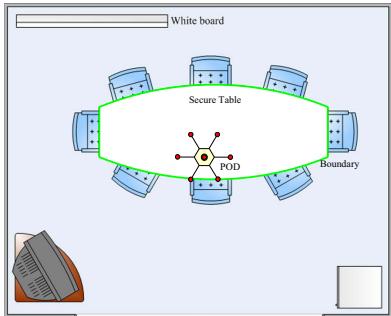
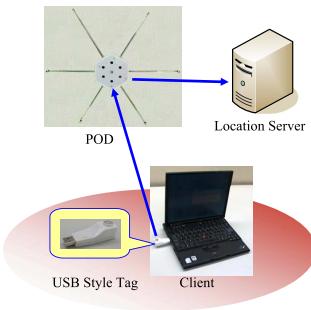
where $x_i, y_i, x_{i+1}, y_{i+1}$ are the coordinates of $v[i]$ and $v[i+1]$ in the coordinate system originated at X . So the trip starts from $v[1]$ and heads to $v[2]$ in anticlockwise direction. A round-trip will be formed after coming back to $v[1]$. Judgement is done by verifying $Angle$ after finishing the round-trip:

- X is in P , if $Angle = 2\pi$;
- X is not in P , if $Angle = 0$;
- X is on the edge of P , if $Angle = \pi$;

Applications of RTJ to convex and concave polygon shape secure zones have been shown in Fig.9 and Fig.10. In both the two examples, $Angle = 2\pi$ after the round-trip and the results match the real conditions. The RTJ algorithm need only once loop processing of the vertexes, with complexity $O(n)$, which is very fast.

6 Secure Table: Prototype of the LOCK System

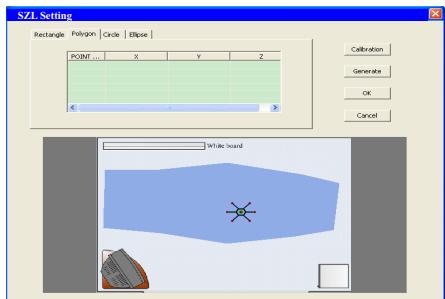
In the previous sections, we have introduced the setting-up and run time judgement methods of the LOCK system. Based on them, We built the prototype of the LOCK system in a meeting room of our lab. The size of the room is 580cm×480cm. A table in the meeting room was our desired secure zone, which

**Fig. 11.** Secure table scenario**Fig. 12.** Ultrasound positioning device

was named as “secure table”. Only when the mobile clients were within the table, they could access the confidential files. The confidential files were stored in a FTP server and the mobile client was attached with a USB-style Tag. It moved in the room and tried to access the FTP server. An ultrasound positioning system with a POD, and a location server was deployed in the room to provide positioning service. The “secure table” and the positioning device are shown in Fig.11 and Fig.12.

6.1 MFC and CSZF Implementation

MFC and CSZF were implemented in the access control engine, which was implemented as a middle-ware by Visual C++ 7.0. We developed a Graphical User Interface(GUI) to carry out the interactive process in MFC and CSZF characterization. Fig.13 shows the snapshot of MFC implementation. Its left part shows the map of the room, which was loaded into the ACE to work as background. To calibrate the parameters of the POD and the size of the room, three feature points were selected at the room corners. The working process of MFC is implemented as following.

**Fig. 13.** Calibrate the POD and the Room with MFC method**Fig. 14.** Characterize secure zone by CSZF

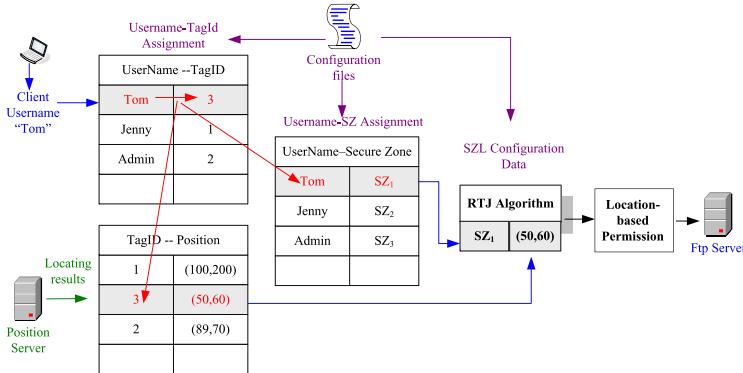


Fig. 15. Online working process of ACE to control location-based FTP access

1. Click on the map of the GUI will generate a feature point, i.e. the green point.
2. Place a Tag at the real place of a feature point, the relative position of the Tag will be shown in the right window, i.e., the red points in Fig.13.
3. After three such feature points are located, MFC method automatically calibrates the POD and the size of the room.

So that the POD is calibrated without manual measurements.

Fig.14 shows the automatic CSZF process. To define the “secure table”, a Tag is moved along the boundary of the table to collect a sequence of positioning results. Then the slope-based subsequence division and the line-fitting based vertex determination methods were applied to automatically determine the “Secure Table”. The polygon shape area shown in the Fig.14 is the “Secure Table” seen by the positioning devices.

6.2 FTP Access Control Implementation

In addition to the above settings, ACE sets up two tabulations for implementation of FTP access control. The first table is the Username-TagID assignment. Because every client is attached with a Tag, this table matches username of each client to a TagID. The second table is the Username-secure zone assignment, assigning each username to a secure zone.

The online working process of ACE for FTP control is shown in Fig.15.

1. It receives the client's request attempt to the FTP server.
2. Picks up the Username, and matches to a TagID according to the Username-TagID assignment.
3. Retrieves the Tag's real-time location from the location server using the TagID.
4. Gets the secure zone of the user from the Username-secure zone assignment.

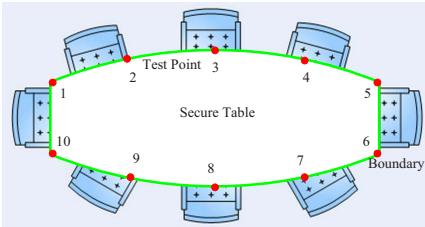


Fig. 16. Selection of Test Points

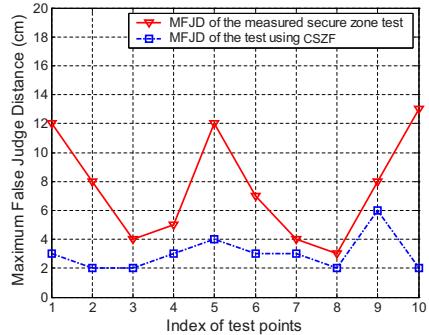


Fig. 17. MFJD Evaluation

5. Judges the location-based permission by RTJ based on the real-time position and the client's assigned secure zone.
6. Allows the Ftp access, if the location-based permission is TRUE. Denies the access if the location-based permission is FALSE.

7 Experiment Evaluation

Based on the above prototype, we carried out a series of experiments to evaluate the performances of the LOCK system, mainly on access judgement resolution and the calibration effort.

7.1 Access Judgement Resolution

We firstly verified the access judgment resolution of the LOCK system. A criterion regarding the access judgment resolution is defined as the “Maximum False Judge Distance(MFJD)”. It is defined with following heuristics. There are two kinds of false judgements. 1). The Tag is indeed inside the boundary, but is judged outside; 2). The Tag is outside the boundary, but is judged inside. When the judgement is false, the distance from the Tag to a closest point on the boundary indeed indicates the severity of this false. There is a critical distance d_{max} , at which the judgment will change from false to correct. Therefore, the maximum distance from the Tag to the boundary within which the judgement keeps false is defined as the MFJD criterion.

Ten points on the boundary of the “secure table” were selected to test the access judgement resolution with above criterion. These points were indexed from 1 to 10 as shown in Fig.16. At each test point, a Tag was moved repeatedly across the boundary of the table to test whether the location-based access judgement was correct. When the judgement was false, the critical distance from the Tag to the table boundary was recorded. Thirty times of tests were carried out at each point, and the mean value of MFJD was recorded. The results of mean MFJD

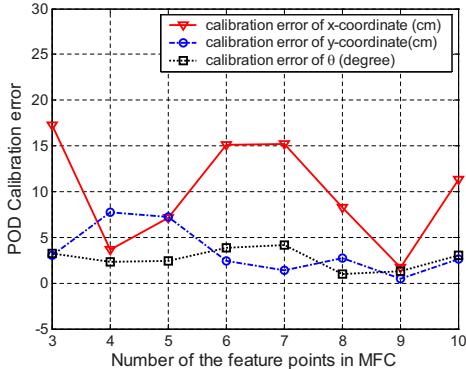


Fig. 18. POD calibration error VS. the number of feature points used in MFC

at each test point were plotted in Fig.17 in dashed line. We can see the MFJD is less than 6cm at all the test points.

To work as a comparison, another test using the manually measured secure table was carried out. In this test, the secure table was manually measured in the global coordinate system and stored for online use. The same tests were done at the same ten test points. The mean MFJD at each test point was plotted in Fig.17 in solid line. The overall MFJD is less than 14cm. We can see the MFJDs in the test using manually measured secure table are much larger than in the test using CSZF. This supports our analysis that CSZF will compensate the disparity between the user desired secure zone and the “secure zone seen by the positioning device”. The access judgement resolution is therefore much improved.

7.2 Calibration Efforts

Then we evaluated the calibration efforts, which used to be generally difficult to be quantized. To evaluate this performance of the LOCK system, we proposed the number of feature points used in the MFC process as a criterion. The calibration accuracy was evaluated as a function of the number of feature points. To calculate the calibration error, the ground truth of the x,y coordinates of the POD center and the angle of the POD’s X-axis was manually measured. Their calibration results were got from MFC process. The differences from calibration results to the ground truth were used as the calibration error.

The experiments were carried out in the same environment of the “secure table”. Eight tests were done, in which, the number of feature points was increased from three to ten. The calibration errors as a function of the number of feature points are shown in Fig.18. The solid line is the calibration error of the x-coordinate of the POD center; the dashed line is the calibration error of the y-coordinate. The dotted line is the calibration error of the angle from POD’s X-axis to the absolute X-axis. We can see that the overall calibration errors of

x-coordinate are less than 20cm; the overall errors of y-coordinate are less than 10cm; and the overall errors of θ are less than 5 degrees . The errors decrease slightly with increasing of the number of the feature points. This gives us hints that the POD can be easily and accurately calibrated with small number of feature points, which verifies that the LOCK system can be easily set up without manual measurement efforts.

8 Conclusion

In this paper, we designed, implemented and evaluated LOCK, a highly accurate and easy-to-use LBAC system. It uses autonomous ultrasound positioning devices and an access control engine to precisely characterize the secure zones and accurately judge the online access authority. The LOCK meets the application desired properties from following aspects. 1). A Measurement-Free Calibration method has been designed to automatically calibrate the autonomous positioning devices to transform their relative positioning results to an absolute coordinate system. These positioning devices are self ease-to-install and can provide highly accurate relative positioning. So that the LOCK system can be easily deployed and provides accurate access control. 2). Secure zones are precisely defined by a Coherent Secure Zone Fitting method. It on one hand removes the manually secure zone characterization efforts and on the other hand compensates the disparity between the user-desired secure zone and the secure zone seen by the locating devices to improve the accuracy of LBAC. 3). In online phase a round-trip judgement algorithm has been presented to determine the geographical relationship between client's position and the secure zones, making the access control decision very fast. To our knowledge, LOCK is the first LBAC system which can be set up without manual measurements and can provides LBAC with centimeter resolution. A prototype of the LOCK system has been developed and evaluated by defining a meeting table as secure zone to control the FTP access. The results verified above desired properties. In future work, simultaneously multiple POD calibration can be pursued to further improve the easy-to-use property of the LOCK. Hierarchical relationship among the secure zones can be further exploited to automatically generate the location-based secure policies.

References

1. Junhui, Z., Yongcai, W.: Autonomous Ultrasonic Indoor Tracking System. In: Proceedings of the 2008 International Conference on Intelligent Pervasive Computing (IPC 2008), Sydney, Australia (2008)
2. Mao, G., Fidan, B., Anderson, B.D.: Wireless sensor network localization techniques. Comput. Netw. 51(10), 2529–2553 (2007)
3. Pan, J.J., Pan, S.J., Zheng, V.W., Yang, Q.: Digital Wall: A Power-efficient Solution for Location-based Data Sharing. In: PERCOM 2008, pp. 645–650. IEEE Computer Society, Washington (2008)

4. Kapadia, A., Henderson, T., Fielding, J., Kotz, D.: Virtual walls: Protecting digital privacy in pervasive environments. In: LaMarca, A., Langheinrich, M., Truong, K.N. (eds.) *Pervasive 2007*. LNCS, vol. 4480, pp. 162–179. Springer, Heidelberg (2007)
5. Capkun, Hubaux, J.P.: Secure positioning of wireless devices with application to sensor networks. In: Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 3, pp. 1917–1928 (2005)
6. Bahl, P., Padmanabhan, V.: RADAR: An in-building RF-based user location and tracking system. In: Proceedings of the Conference on Computer Communications, Tel Aviv, Israel, vol. 2, pp. 775–784 (2000)
7. Rodriguez, M.D., et al.: Location-Aware Access to Hospital Information and Services. *IEEE Transactions on Information Technology in Biomedicine* 8(4), 448–455 (2004)
8. Cho, Y., Bao, L., et al.: LAAC: A Location-Aware Access Control Protocol. In: Proceedings of Third Annual International Conference on Mobile and Ubiquitous Systems, San Jose, CA, pp. 1–7 (2006)
9. Sastry, N., Shankar, U., Wagner, D.: Secure verification of location claims. In: Proceedings of the 2nd ACM workshop on Wireless security, New York, NY, USA, pp. 1–10 (2003)
10. Stajano, F., Anderson, R.J.: The resurrecting duckling: Security issues for ad-hoc wireless networks. In: Proceedings of the 7th International Workshop on Security Protocols, London, UK, pp. 172–194 (2000)
11. Balfanz, D., et al.: Talking to strangers: Authentication in adhoc wireless networks (Feburary 2002)
12. Rasmussen, K.B., Capkun, S., Cagalj, M.: Secnav: secure broadcast localization and time synchronization in wireless networks. In: *MobiCom 2007*, New York, NY, USA, pp. 310–313 (2007)
13. Goldsmith, A.: *Wireless Communications*. Cambridge University Press, Cambridge (2005)
14. Lorincz, K., Welsh, M.: MoteTrack: A Robust, Decentralized Approach to RF-Based Location Tracking. In: Strang, T., Linnhoff-Popien, C. (eds.) *LoCA 2005*. LNCS, vol. 3479, pp. 63–82. Springer, Heidelberg (2005)
15. Savvides, A., Han, C., Strivastava, M.B.: Dynamic fine-grained localization in ad-hoc networks of sensors. In: *MobiCom 2001*, Rome, Italy, pp. 166–179 (2001)
16. Want, R., et al.: The Active Badge Location System. *ACM Transactions on Information Systems (TOIS)* 10(1), 91–102 (1992)
17. Ward, A., Jones, A., Hopper, A.: A New Location Technique for the Active Office. *IEEE Personal Communications* 4(5) (October 1997)
18. Priyantha, N.B., Chakraborty, A., Balakrishnan, H.: The Cricket Location-Support System. In: *ACM MobiCom 2000*, Boston, Massachusetts, USA (August 2000)
19. Naohisa, H., et al.: Assets Location Management Solution Based on the Combination of SmartLocator and RFID. *NEC Technical Journal* 1(2) (2006)
20. Duff, P., Muller, H.: Autocalibration Algorithm for Ultrasonic Location Systems. In: Proceedings of Seventh IEEE International Symposium on Wearable Computers (ISWC 2003) (2003)
21. Scott, J., Dragovic, B.: Audio Location: Accurate Low-Cost Location Sensing. In: Gellersen, H.-W., Want, R., Schmidt, A. (eds.) *PERVASIVE 2005*. LNCS, vol. 3468, pp. 1–18. Springer, Heidelberg (2005)

Characterizing the Space by Presenting Non-visual Information

Tomohiro Akagawa¹, Takuji Narumi², Young Ah Seong²,
and Takashi Kiriyma¹

¹ Tokyo University of the Arts

2-5-1 Shinko, Naka-ku, Yokohama, Japan

{toakmoak, kiriyma}@gsfnm.jp

² The University of Tokyo

7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan

narumi@cyber.t.u-tokyo.ac.jp, yabird@hc.ic.i.u-tokyo.ac.jp

Abstract. We are aiming at adding characteristics to the existing space by providing people with non-visual, location-dependent information. By manipulating non-visual information presented to people, we can change implicit partitioning of the space without physically reconstructing it. “Thermotaxis” is a system that gives sensations of cool and warm to users by controlling thermoelectric devices wirelessly. In this system, the space is characterized as being cool or warm. Users experience the difference in temperatures while they walk in the space. Preliminary analysis shows that people stay close in the area of a comfortable temperature.

Keywords: ambient controlling, reconfigurable spatial structure, thermal sensation.

1 Introduction

To change the spatial structures of our living spaces, we usually take an architectural approach. Spatial structures are constructed by building walls to partition open spaces. But once such structures have been constructed, re-configuring them needs substantial costs. Reserving flexibility for changing spatial design is essential for effective use of spaces.

Our research aims at making spatial design flexible. We assume that presenting location-based information to the occupants will provide the designer with additional flexibility in the use of spaces. Combining information technology and architecture have been argued [1,2]. The Austrian architect Hans Hollein has explored several possibilities of new electronic media with architecture in 1960's. In his claim, a space is as sculpture, or as a “determined activated region in indefinite three dimensionality”. It suggests that an area of space can be dynamically characterized. Once new characteristics are added to an existing space, the relationship between the space and the occupants will change.

In spatial information design, there are many studies on presenting information that dynamically changes, including pervasive, ubiquitous and ambient

computing[3]. Most of them focus on visually presenting additional information about the place [4,5]. But visual information presentation may be obstructive in some situations. The kind of information we are interested in is not visual explanations about the location but part of a spatial structure that implicitly affects occupants' activity in the spaces. We call it ambient controlling of human behavior by presenting non-visual information.

We have studied the recognition of a spatial structure without presenting visual information[6]. In this study, we investigated how the space was recognized if people walked blind while being guided by a rail. The result indicates that people have more accuracy in recognizing topology than distance. Since the blind condition disables to see a surrounding situation, it is difficult to correspond to an imaginary spatial information recognized by people and a real space. And routing guide reduces the flexibility of the space.

In this paper we propose a method of characterizing the space by presenting thermal information. Although thermal sensing is slow in response and low in resolution than visual and audio sensing, it is suitable for informing users gradually without giving a clear border of characteristics. In the rest of this paper, we discuss an experiment of presenting thermal information to characterize the space. We also describe analysis of people's behavior in the space.

2 Presenting Non-visual Information

As discussed in the previous section, we introduce a thermal characteristic to a space. A thermal characteristic has a larger ambiguity than that of visual and audio information. And the response time is longer than visual and auditory sensation. However, among all types of sensations, thermal sensing is effective in ambient controlling for presenting information non-intrusively.

There is an advantage of using non-visual information presentation. Screens and displays force the eyes of users on them. For example, ambient display[4] was studied as a way of implicit information presentation. Without regular computer displays, users become free from the desktop to receive information. There is a drawback in this approach, however, that users are required to pay attention to where the information is presented, which constrains their activities. Another example is the concept of ubiquitous computing and pervasive computing, which try to transform spaces that are filled with computers into intelligent spaces with communication capabilities [7,8,9,10,11]. In many of these studies, users have to walk around with a mobile terminal or a hand-held communication device to get annotated information. But using devices with visual monitors are not applicable to implicit information presentation. We want to encourage changing user's behavior regardless of whether the user expects it.

There are several ways to present non-visual information such as sound and touch. Some applications that construct spatial structures using non-visual information presentation have been developed.

Yeppe Hein put a fixed number of infrared emitters in an empty room in his artwork called "Invisible Labyrinth"[12]. In this room, he defined some different

programmed labyrinths using infrared emitters mounted on the ceiling in a grid. Visitors wear a headphone on the head. They can freely walk around in the space. But if they cross a virtual wall, an infrared sensor attached to the headset reacts to trigger vibration. The open room presents a spatial structure without physical walls. By using information technology, the space is augmented with a new flexible structure.

Another example of presenting non-visual information is “Monolith”, which is an LED sculpture by United Visual Artists[13]. This artwork is a symbolic responsive LED sculpture that makes a huge noise if the visitor comes close to it. By this behavior of the system, the visitor is forced to move to a moderately comfortable place. In this case, the sculpture creates a spatial structure by presenting audio information that affects users around it.

In these studies non-visual information is used to characterize spaces. However, information representation using auditory properties is susceptible to surrounding sound environment, besides the sound can be intrusive noise for unconcerned people. One solution is providing a wearable headphone for each user, but presenting information by a headphone disturbs their talking activities. In addition, information representation using touch sensation such as a tactile display and vibration does not have sustainability in daily life.

To make spatial design flexible, we have to consider not to disturb the occupants’ activities. These activities include walking around freely, speaking with others and hear some voice. Presenting thermal information is suitable for unintrusively informing users without giving a clear border of characteristics.

3 Thermotaxis

3.1 System Overview

In this section, we propose a system of controlling thermal characteristics and discuss how thermal information affects people’s behavior.

Thermotaxis is a system that creates thermal sensory spots in an open space. It makes spatial partitions without physical walls. Instead, it displays temperatures in several grades. In this system, we use an earmuffs-like wearable devices that provide thermal senses depending on the location of people. By feeling the temperature, people can distinguish different thermal areas, though there is no visual separation between them.

In Thermotaxis, the space is divided into several thermal fields. Visitors are expected to walk around to find their comfortable position based on their thermal senses. Due to the variations of desired conditions and surrounding environment such as the air temperature, different positions are found to be comfortable by different visitors. For example, on a cold day in winter, a comfortable place would be warm unlike a summer day. People who have a similar preference would gather together. (Fig. 1)

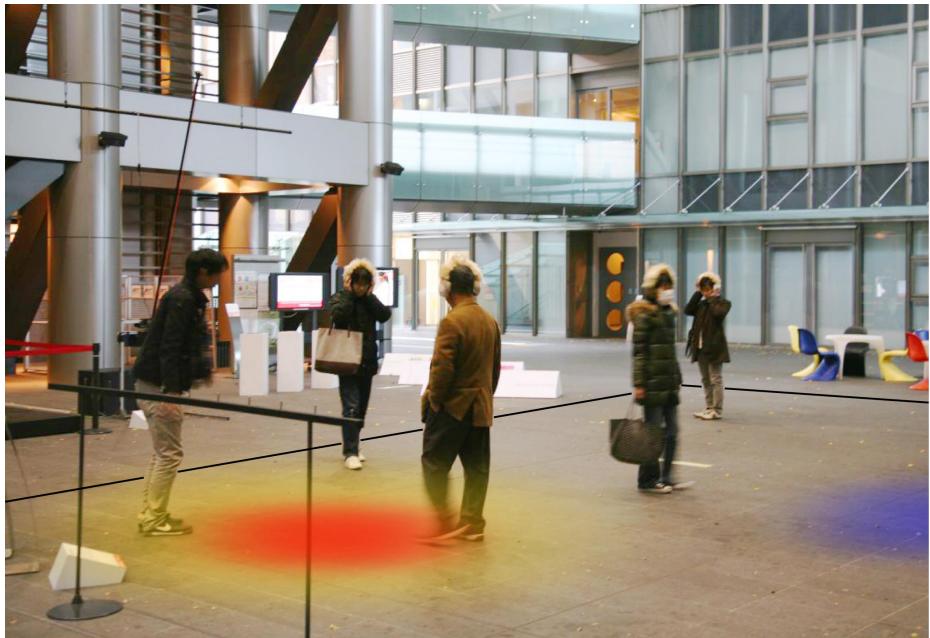


Fig. 1. Thermotaxis — People are walking around to find thermal spot

3.2 Hardware Configuration

The experimental device is designed to be controlled by a computer as an operating unit via wireless communication. All electronic modules that control temperature are installed in the earmuffs.

This system consists of several earmuffs-like wearable devices and a control unit that controls the wearable devices and recognizes their locations. Figure 3 shows the configuration of devices.

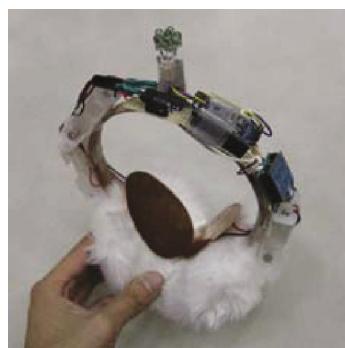


Fig. 2. Earmuffs Device

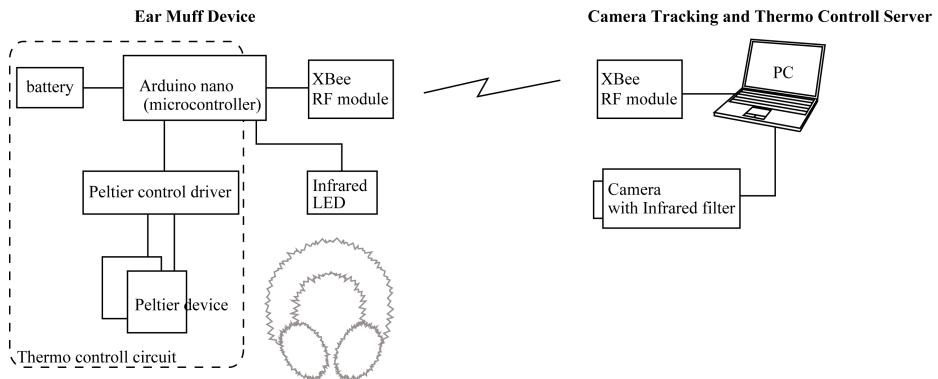


Fig. 3. Configuration of devices and control unit

An IEEE 1394 camera with an infrared filter for earmuffs tracking was attached to the ceiling about eight meters above the floor. Infrared LEDs are attached to the top of the earmuffs device for camera tracking. Sensing with a camera is easy to install if the camera has a clear line of sight to all wearable devices. Figure 4 depicts the system layout. When the control unit requests locations of the wearable devices, the camera detects their positions by capturing blinking infrared LEDs mounted on their tops.

To build a wearable device that displays warm and cool temperatures (Fig. 2), we use an Arduino Nano microcontroller. It controls two peltier devices in each side of the earmuffs. It also controls infrared LEDs. There are five heating levels on the peltier device control. Two of them are heating, two of them are cooling and one in the middle is without heating or cooling level. The

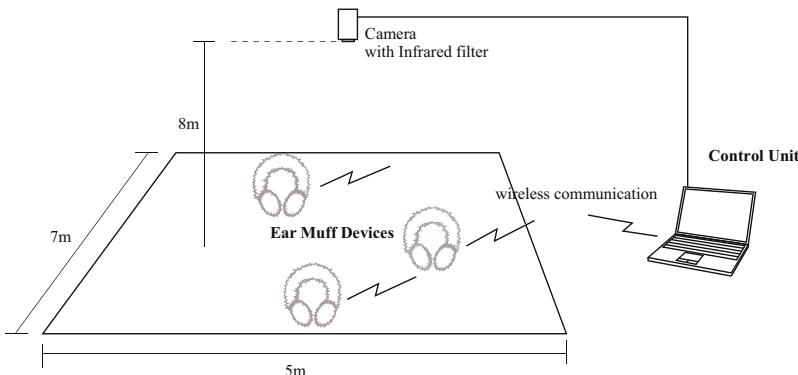


Fig. 4. System layout

difference in temperature created by these peltier devices is about twenty degrees to forty degrees.

The control unit and the earmuffs device communicate via a zigbee network. If the position was detected, the unit determines the heating level according to the thermal field send the temperature level to the earmuffs device. Because the thermal fields are defined by software, we can change the map of fields dynamically.

4 Experiment

4.1 Experimental Settings

An experiment was performed to examine how the thermal characteristic of the space affects people's behavior in the space. There were six earmuffs, which allowed six people to experience the system at the same time. This experiment was conducted in December 2008 at the University of Tokyo as an art exhibition. The air temperature during the experiment ranged from a low of 8 degrees to a high of 12 degrees in Celsius, averaging 10 degrees.

Approximately a total of 400 people ranging from teens to 60s experienced this system. All of them were told to put on the earmuffs device and walk around in the open space to find various thermal areas. The dimensions of the open space was about 5mx7m. In order to examine the trajectories of people in that space, the system was programmed to record logs of positions detected by the control unit. The characterizing map was designed to have five thermal grades. (see the map of Figure 5).

4.2 Post-Processing

Before performing analysis, the recorded data were preprocessed to remove outliers and detecting errors. These errors include (i) wrong locations due to the sunlight, (ii) inaccurate data due to people moving faster than the desired speed for the system, and (iii) missing data caused by wrong direction of infrared LEDs on the earmuffs.

Since the Thermotaxis system uses infrared light to detect the positions of earmuffs devices, it is disturbed by the sunlight reflection. Errors in data were removed by hand because we could easily detect them.

This system works well if the infrared LEDs are located in the center of the space and aim at the camera correctly. Leaps in trajectories occur when people move too fast to track or the LEDs go out of sight of the camera. To analyze the movement of people among different thermal areas, inadequate points should be corrected.

5 Analysis and Discussion

In this section, we present analysis of recorded data described earlier and discuss people's behavior in the space. This analysis consists of the following three points;

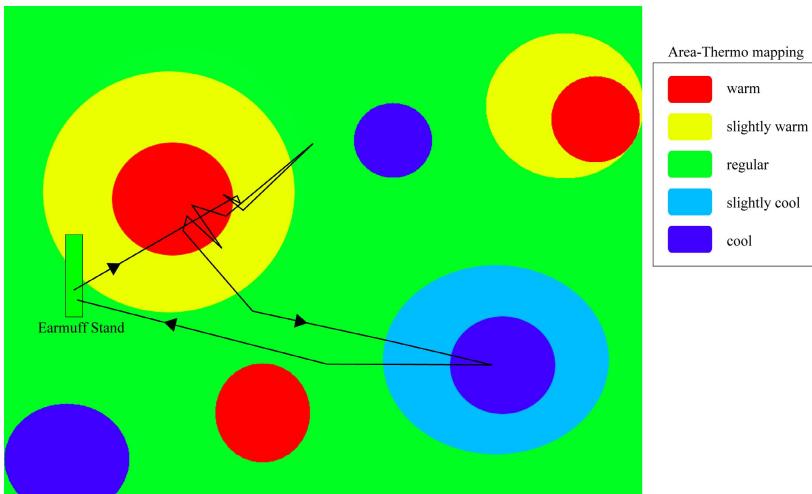


Fig. 5. A trajectory of a certain subject

- the relation between area ratio and sojourn time in each area,
- the transition behavior of subject on the boundary between areas, and
- the influence for other people on subject being in certain area.

The purpose of the first analysis is to determine which areas people prefer. The second analysis is performed to investigate transitions among areas over time. The last analysis is to measure influences of one's location to other people.

5.1 Area Ratio and Sojourn Time

Figure 5 illustrates an example of user's trajectory. In this map, red areas are warm, yellow areas are slightly warm, cyan areas are slightly cool, blue areas are cool and green areas are at a regular temperature area. Table 1 compares areas of the thermal fields indicated in Figure 5. The regular area is the largest, followed by the slightly warm area. The cool, warm and slightly cool areas are approximately of the same size.

Table 2 shows the total sojourn time of all people. The total time spent in warm is the longest although the regular area is the largest. This result means that people preferred to stay longer in the warm area.

Figure 6 depicts a Venn diagram of arrival rate of each area. About 83 percent of people reached the warm area, while only 22.4 percent reached the cool area. This result can be attributed to the topology of the map. As shown in Figure 5, the warm and slightly warm areas are closer to the earmuffs stand which is the initial position of the trajectory. In other words, 80 percent of people searched near the starting point and about 60 percent were satisfied to discover only the warm area.

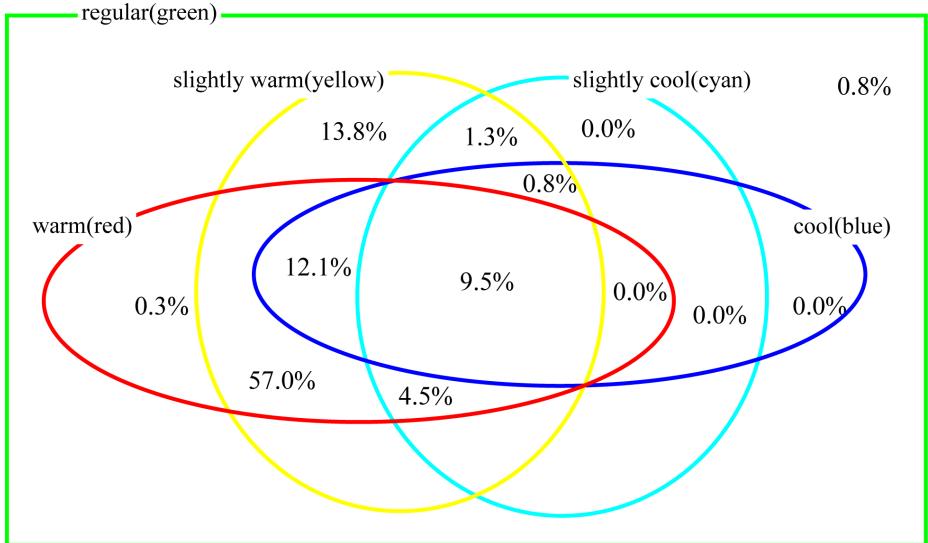
The air temperature may be a factor of this result. It is likely that people preferred to the warm area because of the low air temperature.

Table 1. Thermal Area Ratio

cool(blue)	slightly cool(cyan)	regular(green)	slightly warm(yellow)	warm(red)
6.9%	7.9%	62.6%	15.0%	7.5%

Table 2. Sojourn time summated all subjects

cool(blue)	slightly cool(cyan)	regular(green)	slightly warm(yellow)	warm(red)
1846sec	1043sec	9515sec	9313sec	11526sec

**Fig. 6.** Venn diagram of arrival factor

5.2 Area Transitions

Area transitions in one's trajectory can be represented as in Table 3. It shows examples of transitions. Trajectory A is shown in Figure 5 as an example.

Figure 7 illustrates, all possible transitions among areas. Numbers on links indicate the times these transitions occurred. The numbers of transitions between 3(regular) and 4(slightly warm) and between 4 and 5(warm) are significantly larger than others.

Almost all people experienced the transition shown in Figure 8-(a). But in the transitions between 3-4-5 there were come-and-go transitions as shown in Figure8-(b). Such transitions occur when people search for the boundary between areas. People walked on the edge of areas to compare the difference of them.

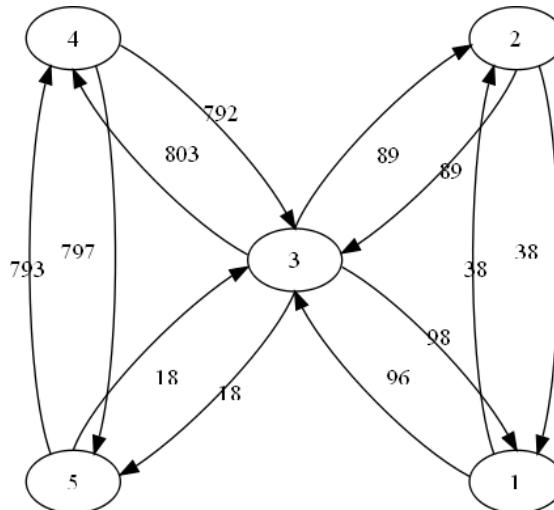


Fig. 7. Transitions between thermal areas

To support this claim, we examine trajectories of people who experienced 3, 4 and 5 areas to find such come-and-go transition. To count come-and-go transitions that appear more than two times in a single trajectory, we use following calculation:

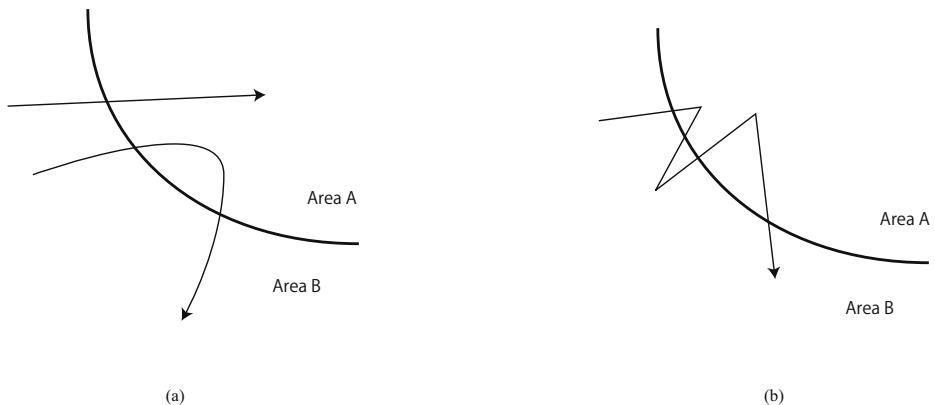
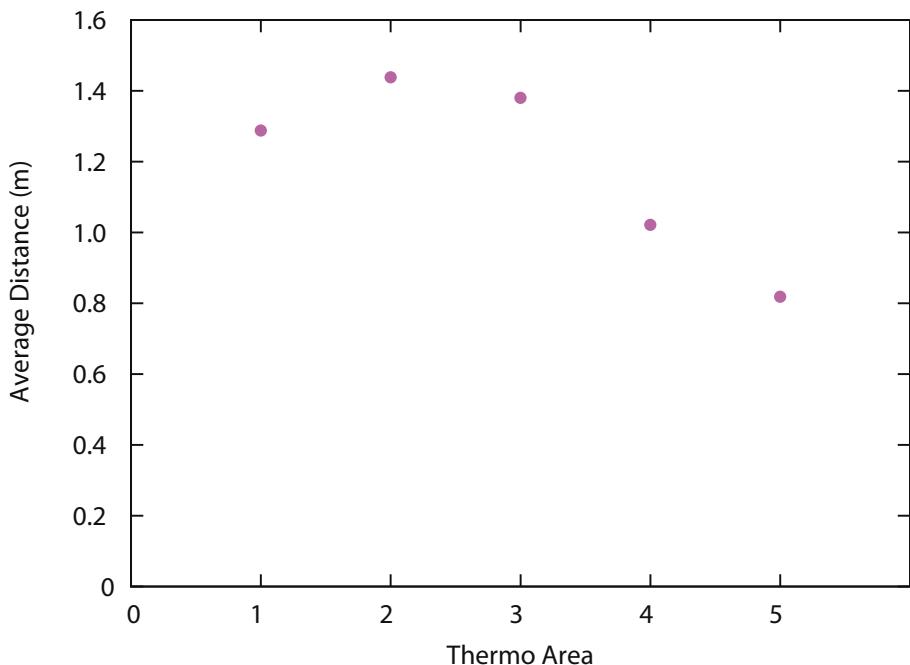
$$count_{H_1, H_2} = \sum_{n=i}^N \sum_{t=0}^T C(n, t) \quad (H_1, H_2 = 3, 4, 5) \quad (1)$$

$$C(n, t) = \begin{cases} 1 & \text{if } (H_1(n, t) = H_1(n, t + 2) \text{ and } H_2(n, t + 1) = H_2(n, t + 3)) \\ 0 & \text{otherwise} \end{cases},$$

where H is a function that returns the area, from given subject n and time t . As a result, 48.6% out of all the people searched for boundary between areas. This result indicates at least half of the people performed searching on the border to find the difference between thermal senses.

Table 3. Examples of user's transition sequence (The number means each thermal area : 1=cool 2=slightly cool 3=regular 4=slightly warm 5=warm)

Trajectory	Area Transitions
A	3 4 5 4 5 4 3 4 5 4 5 4 5 4 3 2 1 2 3
B	3 4 5 4 3 4 5 4 3
C	3 4 5 4 3 4 5 4 5 4 5 4 5 4 5 4 3 4 3
D	3 4 5 4 3 4 5 4 5 4 3 2 1 2 3
E	3 4 5 4 3 4 3 4 5 4 5 4 3 4 5 4 3 4 3 4 3
F	3 4 5 4 3
G	3 4 5 4 3 1 3 4 3 4 3 4 5 4 5 4 3

**Fig. 8.** Action of searching difference**Fig. 9.** Average distance between people sorted by locations

5.3 Influences on Other Subjects

In an open space, people can see each other. If one stays in a particular place, other people may wonder why he or she is there. People influence to each other by their positions.

Figure 9 shows average distance between people when one of them is in area $N(N = 1\ldots 5)$. The average distance in the warm($N = 5$) and slightly warm($N = 4$) areas are small. There is little difference in the regular($N = 3$), slightly cool($N = 2$) and cool($N = 1$) area.

This result means that if there is at least one person in the warm or slightly warm area, people tend to be close each other. In other word, people stay close in the warm and slightly warm areas.

6 Conclusions and Future Work

This paper discussed the idea of characterizing the space by presenting non-visual information. Unlike physical walls or partitions, creating non-intrusive partitions using information mapping in an existing space enables us to distinguish areas in space flexibly. In Thermotaxis, we characterize an open space with thermal information. Wearable thermal devices have been created to present an imaginary spatial structure.

We conclude that affecting people's behavior is possible by reconfiguring an open space with non-visual spatial structure presented as thermal information. The result of our experiment shows that presenting thermal information affects people's behavior in making them get together in a warm area under certain conditions.

We believe that flexible areas and landmarks created by this system can stimulate communication among occupants. Creating added value to a space by non-visual information makes the space more attractive.

As for implementation, there is obviously a need for further improvement of the system and experiment. In Thermotaxis, using a camera required a large space and high ceiling. Tracking and location detection methods should be improved. Further investment is needed about how people's behavior changes if the thermal map dynamically changes.

We are also interested in using other media to perform ambient controlling. Although the use of visual or audio senses is reliable to deliver information accurately, there is an advantage of presenting information passively and unconsciously. We believe that it is worth investigating application of ambient controlling in new media.

Acknowledgement

We would like to thank Taro Suzuki, Takeshi Naemura, Hiroshi Harashima and Michitaka Hirose for their comments on this work.

This research is supported by JST(Japan Science and Technology Agency) CREST (Core Research for Evolutional Science and Technology) and by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for JSPS(Japan Society for the Promotion of Science) Fellows.

References

1. Ujigawa, M., Hanazato, T.: A study on the influence of the information technologies to buildings. *AIJ Journal of Technology and Design* (22), 573–576 (2005)
2. Lefavire, L.: Everything is architecture — multiple hans hollein and the art of crossing over. *Harvard Design Magazine* (18) (2003)
3. Weiser, M.: The computer for the 21st century. *Scientific American* 265(3), 94–104 (1991)
4. Wisneski, C., Ishii, H., Dahley, A., Gorbet, M., Brave, S., Ullmer, B., Yarin, P.: Ambient displays: Turning architectural space into an interface between people and digital information. LNCS, pp. 22–32. Springer, Heidelberg (1998)
5. Redström, J., Skog, T., Hallnäs, L.: Informative art: using amplified artworks as information displays. In: DARE 2000: Proceedings of DARE 2000 on Designing augmented reality environments, pp. 103–114. ACM, New York (2000)
6. Narumi, T., Akagawa, T., Seong, Y.A., Hirose, M.: Absolute field: Proposal for a re-configurable spatial structure. In: ACM International Conference on Advance in Computer Entertainment Technology (ACE 2008) (2008)
7. Shklovski, I., Chang, M.: Guest editors' introduction: Urban computing—navigating space and context. *Computer* 39(9), 36–37 (2006)
8. Kindberg, T., Chalmers, M., Paulos, E.: Guest editors' introduction: Urban computing. *Pervasive Computing*, IEEE 6(3), 18–20 (2007)
9. Wilson, J., Walker, B., Lindsay, J., Cambias, C., Dellaert, F.: Swan: System for wearable audio navigation. In: 2007 11th IEEE International Symposium on Swan: System for wearable audio navigation. *Wearable Computers*, pp. 91–98 (2007)
10. Nishimura, T., Itoh, H., Nakamura, Y., Yamamoto, Y., Nakashima, H.: A Compact Battery-Less Information Terminal for Real World Interaction. In: Ferscha, A., Mattern, F. (eds.) *PERVASIVE 2004*. LNCS, vol. 3001, pp. 124–139. Springer, Heidelberg (2004)
11. Maeda, E., Minami, Y.: Steps towards ambient intelligence. *NTT Technical Review* 4(1), 50–55 (2006)
12. Hein, J.: Invisible labyrinth (2005)
13. United Visual Artists: Monolith, <http://www.uva.co.uk/archives/31>

Author Index

- Akagawa, Tomohiro 271
Barceló-Arroyo, Francisco 207
Blanke, Ulf 192
Bolliger, Philipp 37
Burke, Jeff 52

Choujaa, Driss 19
Chu, Maurice 37
Ciurana, Marc 207

Dulay, Naranker 19
Estrin, Deborah 52
Fukushima, Toshikazu 254

Hansen, Mark 52
Hansen, Rene 138
Harle, Robert 120
Hashizume, Hiromichi 240
Hay, Simon 120
Hazas, Mike 222

Ito, Toshio 240
Kalnis, Panos 70
Kawauchi, Kensaku 103
King, Thomas 138
Kiriyama, Takashi 271
Kjærgaard, Mikkel Baun 138
Langheinrich, Marc 37
Lemelson, Hendrik 138
López, David 207

Miyaki, Takashi 103
Murray-Smith, Roderick 88
Muthukrishnan, Kavitha 222

Narumi, Takuji 271

Partridge, Kurt 37
Pung, Hung Keng 70

Reddy, Sasank 52
Rekimoto, Jun 103

Sato, Tetsuya 240
Schiele, Bernt 1, 156, 192
Seong, Young Ah 271
Shafer, Steven 174
Shilton, Katie 52
Srivastava, Mani 52
Stikic, Maja 156
Strachan, Steven 88
Sugimoto, Masanori 240

Tulathimutte, Kan 240

Wang, Yongcai 254
Wojek, Christian 1

Xue, Mingqiang 70

Zhao, Junhui 254
Zinnen, Andreas 1