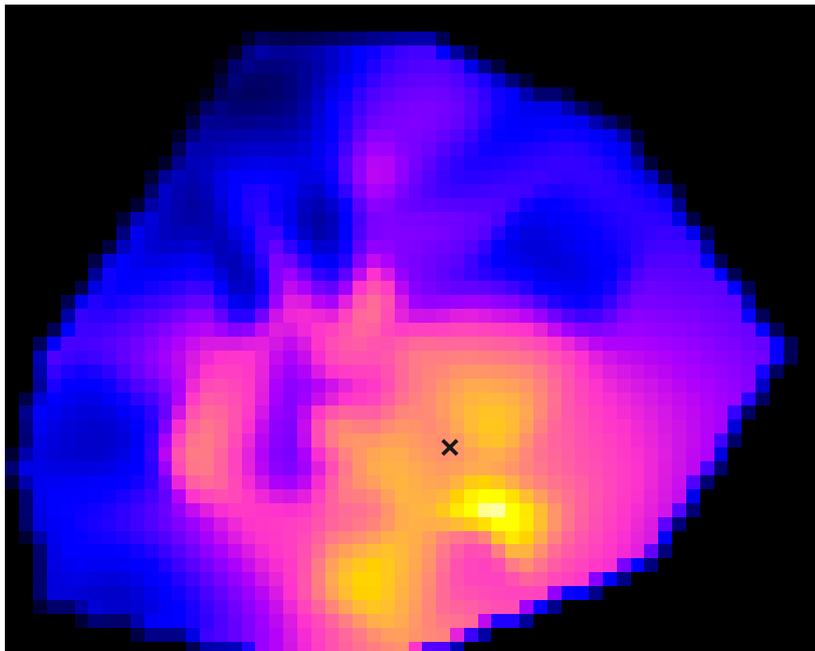


Indoor Localization for Wireless Sensor Networks

Applying multiple frequencies to mitigate multipath effects



Martin Bor

Indoor Localization for Wireless Sensor Networks

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

by

Martin Bor

born in Nieuw-Beijerland, the Netherlands



Embedded Software Group
Department of Software Technology
Faculty EEMCS, Delft University of Technology
Delft, the Netherlands
<http://www.es.ewi.tudelft.nl>

© 2009 by Martin Bor. Some rights reserved.

This work is licensed under the Creative Commons Attribution-Noncommercial-Share Alike 3.0 Netherlands License. To view a copy of this license, visit <http://www.creativecommons.org/licenses/by-nc-sa/3.0/nl/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

Cover picture: Radio map of anchor 7, using multiple frequencies. The color indicates the RSSI, ranging from black for low RSSI (< -90 dBm) to blue, violet, yellow and white for high RSSI (> -45 dBm). The cross indicates the position of the anchor.

Indoor Localization for Wireless Sensor Networks

Author: Martin Bor
Student ID: 1149679
E-mail: m.c.bor@student.tudelft.nl

Abstract

Localization has an important role in wireless sensor networks, as it adds context to gathered data. The goal of this thesis is to create an application that determines the location of a mobile sensor node in an indoor environment. A secondary goal is to get practical experience with indoor localization. Previously, in a literature review, we selected two localization algorithms for this purpose. RIPS was selected since it was tested in practice, uses no additional hardware and has a very good accuracy. CAB was selected because it is a range-free localization algorithm and has low computational complexity. RIPS turned out to be too hardware specific and was not portable to the target hardware. CAB failed because of multipath effects. However, by transmitting beacons on multiple frequencies and averaging over the received signal strength, these multipath effects can be mitigated. In this thesis, we tested the effect of multiple frequencies on a simple proximity algorithm. Although the accuracy did not improve as we expected, it did improve the robustness against temporal disturbances. We also tested the effect of multiple frequencies on two different fingerprinting systems, with the fingerprint database constructed in two ways: a small database, where samples are averaged over all channels and a large database, where samples are averaged per channel. We show that the small database provides the best accuracy with a mean localization error of 2.19 meters.

Thesis Committee:

Chair: prof. dr. K.G. Langendoen, Faculty EEMCS, TU Delft
Committee Members: dr. S.O. Dulman, Faculty EEMCS, TU Delft
dr. ir. A.C.C. Lo, Faculty EEMCS, TU Delft

Acknowledgments

Years of studying, attending lectures, performing lab exercises and taking examinations all led up to a master thesis: the grand finale of the Computer Science Master program. The master thesis is supposed to be a one-man project, but it cannot be done without the help from others. Therefore I want to express my gratitude to a number of people who all made this possible.

First and foremost I would like to thank my parents, for without them I would not be here. Their continuing support and encouragements made this all possible. Equally important are my supervisors Koen Langendoen and Aline Baggio, who inspired and encouraged me. Furthermore I would like to thank Stefan Dulman, Gertjan Halkes and Otto Visser for their support and suggestions. Your help was invaluable.

Lastly I would like to thank my friends and colleagues in the aquarium (or terrarium, depending on the outside temperature and CPU load) on the 9th floor for their talks, discussions, proofreading, lunch breaks and coffee breaks. In no particular order: Niels, Venkat, Maarten, Lian Ien, Frank, Matthias, Vincent, Gijs, Joost, Bart, Stephan, Boudewijn and all the others I forgot to mention.

Thank you.

Martin Bor
Delft, the Netherlands
August 17, 2009

Contents

Contents	i
1 Introduction	1
1.1 Problem Statement	2
1.2 Approach	2
1.3 Structure	2
2 Localization Application	3
2.1 Overview	3
2.2 Deployment Environment	4
2.3 Hardware	4
2.4 Requirements	5
3 Related Work	7
3.1 Definition and Terminology	7
3.2 Range-based Localization	8
3.3 Range-free Localization	8
3.4 Indoor Localization	9
4 RIPS and CAB	11
4.1 RIPS	11
4.2 CAB	13
5 Multiple Frequencies	17
5.1 Multiple Frequencies	17
5.2 Experimental Setup	17
5.3 Results	18
5.4 Proximity	18
6 Fingerprinting	23
6.1 Fingerprinting	23
6.2 Experimental Setup	24
6.3 Results	28
6.4 Discussion	30
7 Conclusions and Future Work	35
7.1 Conclusions	35
7.2 Future Work	36
Bibliography	39

Introduction

Large-scale networks of wireless sensors have become a popular research topic. Advances in hardware technology have led to dramatic reductions in size, energy consumption and cost of digital circuitry, radio chips and sensors. This has led to the concept of “Smart Dust” [11], a hypothetical network of tiny Micro ElectroMechanical Systems (MEMS) equipped with wireless communication that can detect anything from light and temperature, to vibrations and radiation. These networks of small, cheap, energy-efficient and robust sensor nodes can be used for a wide range of applications, such as the long-term monitoring of seabirds living in an adverse environment [20]. Other possible applications are water quality monitoring [7] and the detection of forest fires [36]. With a live demo of a sensor network of 800 nodes at the Intel Developers Forum in 2001, the Smart Dust project was a huge success and has sparked a whole new research field now commonly known as Wireless Sensor Networks (WSNs).

Localization is the process of determining the physical position of a sensor node. It has an important role in WSNs. In many applications it is necessary to locate the nodes and add location information to the gathered data. For example, a forest-fire detection network that only reports that it detected a fire is not very useful. For the gathered data to be meaningful it is essential to know *where* the fire has been detected. Location information adds context to the measured values. In a deployment where the nodes are mobile, localization information is key. For instance when monitoring the migration and interaction of zebras [37]. For indoor environments there are many location-aware applications such as navigational tools for humans, asset tracking, resource discovery and security.

A possible solution to the localization problem is the Global Positioning System (GPS), a well known and freely usable system. There are, however, a number of reasons why GPS is often not a viable option for WSNs. Firstly, adding a GPS chip significantly increases the price of a sensor node. For example, the low-power SiRFstarIII GSC3f/LP [27] costs around \$60, while a typical sensor node cost around \$50. Secondly, a GPS chip has a relatively high power consumption. The previously mentioned low-power SiRFstarIII GSC3f/LP draws a current of around 60 mA in tracking mode, while a typical sensor node has a peak current consumption of around 20 mA. High power consumption greatly reduces the lifetime of a sensor node. Thirdly, GPS is often not usable indoors. There are more sensitive chips such as the u-blox NEO-5Q [31] that can position indoors, but they cost around \$100. Finally, a GPS chip often requires a separate antenna increasing the form factor of the node consid-

erably. GPS is for these reasons not a practical option in WSNs for many applications and thus a new approach to indoor localization is required.

1.1 Problem Statement

The goal of this master thesis is to *create an application that determines the location of a mobile node in indoor environments*. Previous work on this subject either used specialized and expensive hardware or achieved at best room-level accuracy. Our aim is to achieve sub-room level accuracy with no additional hardware. As a secondary goal, we would like to *gain practical experience with localization in an indoor environment*. Localization algorithms are often designed with certain assumptions on the behavior of radio messages in an indoor environment. With practical experience we can verify certain assumptions and acquire new assumptions, which helps in the development of new localization algorithms.

1.2 Approach

To create a localization application a suitable algorithm has to be implemented. Previously, in a literature review of localization algorithms we selected two localization algorithms suitable for our localization application [5]. We selected RIPS, because it was tested in practice, uses no additional hardware and has a very good accuracy. The second algorithm we selected was CAB, because it is a range-free localization algorithm and has low computational complexity. Since the algorithms have not been tested in an indoor environment the first step in reaching our goal is to perform an experiment to see whether they are suitable. From the result of these experiments we decided that neither of these algorithms were suitable for our target application. RIPS was too hardware specific and could not be ported to our target platform. CAB did not work due too multipath effects. However, by transmitting beacons on multiple frequencies and averaging over the received signal strength, these multipath effects can be mitigated [16]. We tested this approach a simple proximity localization. The use of multiple frequencies did increase the robustness against temporal disturbances, but the accuracy was not sufficient for our target application. We use multiple frequencies in a fingerprinting system to achieve sub-room level accuracy.

1.3 Structure

In Chapter 2 the requirements and limitations of the localization application are described. An overview of related work is given in Chapter 3. RIPS and CAB are described in Chapter 4, after which Chapter 5 looks at the use of multiple frequencies. Chapter 6 presents fingerprinting with the application of multiple frequencies. Conclusions and directions for future work are given in Chapter 7.

Localization Application

This chapter describes our target application. Section 2.1 describes the application itself. Sections 2.2 and 2.3 presents the deployment environment and the hardware we used. The application limitations are described in Section 2.4.

2.1 Overview

The purpose of our application is to localize mobile nodes in an indoor environment using a wireless sensor network. An overview of the application is shown in Figure 2.1.

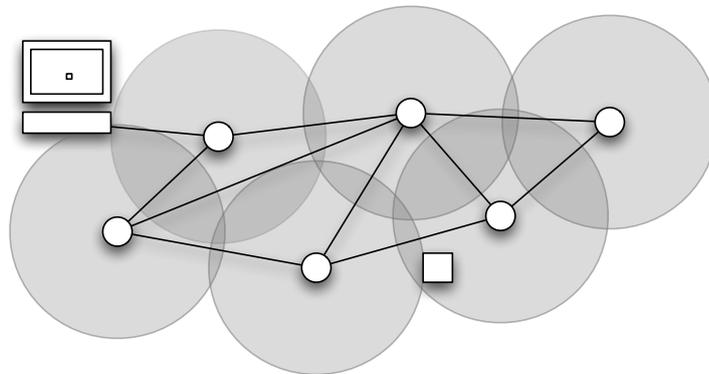


Figure 2.1: Application overview. The square indicates the mobile node, the dots indicate the static sensors with the communication range indicated by a gray circle.

There are three components in the application: the sensor network, one or more trackable mobile nodes and a base station. The sensor network is a collection of static, autonomous nodes, which communicate wirelessly with each other. The task of the sensor network is to detect and localize the mobile node. The sensor nodes send their information to the base station.

The target is an object that roams freely in the environment. It is not necessarily in range of the sensor network. In our target application, the mobile object is a sensor node, which actively participates in the localization process.

The base station is the end point in the sensor network and receives the localization information of the nodes. The task of the base station is to visualize the location information.

2.2 Deployment Environment

The application is going to be deployed in an indoor environment on the 9th floor of the EEMCS building at Delft University of Technology. The floor layout is shown in Figure 2.2. The floor includes 25 rooms.

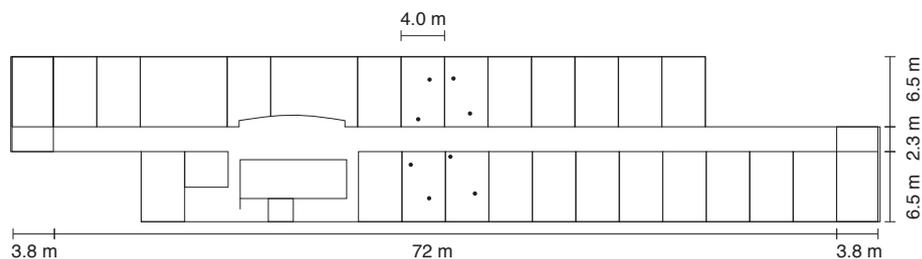


Figure 2.2: Layout of the 9th floor of the EEMCS building at Delft University of Technology. The black dots indicate the position of the nodes in the testbed.

2.3 Hardware

The application uses three types of components: a static sensor network, several mobile nodes and a base station.

Sensor Network The sensor network is a testbed of 8 Moteiv Tmote Sky nodes with known positions, located on the 9th floor of the EEMCS building. The Tmote Sky (Figure 2.3a) has an 8 MHz MSP430 CPU with 10 KB RAM and 48 KB flash, 1024 KB of flash storage and an IEEE 802.15.4 compliant CC2420 radio. The nodes are mounted in the ceiling space, where there is no line of sight with the room below. Localization using infrared or ultrasound, like Active Badge [33] is therefore not possible in this setup. The nodes are connected via USB to Linksys NSLU2's running an embedded Linux platform, which enables the nodes to send serial data over an Ethernet backbone to the central base station.

Mobile Node The user carries a Tmote Invent node (Figure 2.3b), which actively participates in the localization. The Tmote Invent is a Tmote Sky in a user-friendly package, with some additional components, including a 2-axis accelerometer, microphone, a light sensor, an internal temperature and an internal voltage sensor. For user feedback and control there is a speaker, three LEDs, a programmable user button and a reset button.

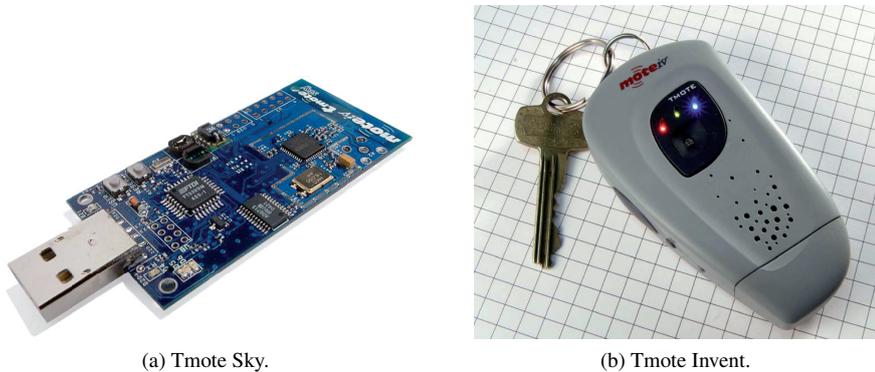


Figure 2.3: Moteiv hardware.

Base Station The base station is a desktop-class PC, which is intended to visualize the localization it receives. The base station can participate in the localization computation. The visualization of the information is, for example, done using a floor plan with the current location of the mobile node and its track.

2.4 Requirements

Given the application requirements and available hardware, there are certain limitations in size, accuracy, latency and privacy.

Size The size of the sensor network is limited to a total of 8 anchor nodes, which span an area of approximately 122 square meters.

Accuracy For practical deployment, at least room level accuracy should be achieved. We aim for sub-room level accuracy, which means an accuracy in the order of 3 meters.

Latency Latency is the time it takes between the start of the localization process and when the position of the mobile node is calculated. Since we intend to use this system at a later stage for tracking, the latency should be no more than 3 seconds. This should be enough to accurately localize a walking person.

Privacy Privacy is the condition that the (estimated) position of the mobile node is only known by the mobile node itself. Because of the setup of the application we do not take privacy in consideration.

Related Work

This chapter describes the related work. Section 3.1 gives a definition of localization and introduces terminology often used. Sections 3.2 and 3.3 present range-based and range-free localization. Several indoor localization deployments are described in Section 3.4.

3.1 Definition and Terminology

Localization is the process of determining the physical position of a node. This position can either be expressed descriptive, i.e. room 12, or spatial such as (3, 5). A spatial position can either be expressed in a global coordinate system or in a local coordinate system. In a global system, the coordinates are expressed in externally meaningful geographic coordinate system, such as latitude/longitude. In a local system, the location of a node is based purely on the topology of the network nodes. Most of the time, the local coordinate system is set up in such a way that there exists a rigid transformation (rotation, reflection, translation) to a global coordinate system.

To simplify the task of assigning coordinates to the nodes, a small number of nodes, so called *anchor nodes* (also often referred to as beacon nodes) know their location beforehand. The location knowledge can be pre-programmed, or acquired through additional hardware such as a GPS receiver.

Many localization algorithms follow a three-phase approach to calculate the node positions, as described by Langendoen and Reijers [17]. The three phases consist of (i) determining the distances between nodes and anchors, (ii) calculating the position of the nodes and (iii) refining the positions using information from neighboring nodes.

There are several methods to calculate the position of a node from the distances between nodes and anchors. Trilateration is an often applied method using the intersection of three circles to calculate the position of a node. Another method is multilateration in which an overdetermined system of non-linear equations is solved. With this approach, the distances from more than three anchors can be used. Multidimensional scaling (MDS) is a data analysis technique from mathematical psychometrics and psychophysics, which determines the placement of points given only the pairwise distances between the points.

Over the years, many localization schemes have been proposed. From the different proposed taxonomies of localization techniques [1, 2, 4, 8, 9, 29, 35], there is a clear division between range-based and range-free localization.

3.2 Range-based Localization

Range-based localization tries to directly determine the distance between two nodes to localize nodes. This process often requires additional hardware. There are a number of techniques to measure the distances between nodes.

RSSI Received Signal Strength Indication (RSSI) is one of the few ranging techniques that does not require additional hardware. Many radio chips can measure the RSSI of a message. By applying a radio signal propagation model, one can calculate the distance from the received signal strength.

RSSI however is not very suitable for indoor localization due to the relative fine-grained requirements. In addition it is difficult to construct a robust propagation model due to multipath, fading and shadowing effects typical for indoor environments [26].

Time of Flight Time of Flight (ToF) measures how long it takes to transfer a radio message from one node to another. Since radio waves travel at the speed of light, one can calculate the distance between two nodes from the duration. Unfortunately, the timers on microcontrollers often used in sensor nodes are not precise enough to be useful. This problem can partially be solved by using ultrasound, which travels at much lower speeds. Ultrasound however requires additional hardware and has a limited range. In addition, ToF requires that the clocks on the nodes are synchronized.

Time Difference of Arrival Time Difference of Arrival (TDoA) exploits the fact that signals with significantly different frequencies travel at different speeds. By measuring the difference in arrival time, one can determine fairly precise the distance between two nodes. This method is often applied with ultrasound and radio messages. TDoA circumvents the time synchronization problem of ToF.

3.3 Range-free Localization

Since additional hardware is often not desired, many range-free localization techniques have been proposed. Range-free localization tries to determine the distance between nodes from radio connectivity.

Proximity Proximity assigns the position of the strongest anchor to the unlocalized node. While this is a very crude method, there are proximity algorithms that are a bit more sophisticated. Centroid [6], for example, defines that the location of the node is at the center of gravity (centroid) of all the anchors it can hear. Proximity-based localization methods are useful in situations where nodes have very limited computational power. The accuracy of this approach however, is not very high. Particularly when the anchor density is low, unevenly distributed or for nodes that are on the edges of the network.

Hop-counting Hop-counting is based on the observation that if two nodes can communicate with each other, the distance between them can be at most the maximum range of their radios. By keeping track of the paths to other nodes in the network, a node can estimate the distance to another node by counting the number of hops on the shortest path. In many hop-counting algorithms, such as the Gradient algorithm by Nagpal et al. [22] or the Ad Hoc Positioning System by Niculescu and Nath [23],

an anchor node initiates a flood of the network by broadcasting its position and a counter, which gets increased at every hop. When a node has estimated the distance to at least three anchors, it can use, for instance, multilateration to infer its own location.

Fingerprinting Fingerprinting is a technique that uses measurements of signal characteristics (*fingerprints*) to estimate the position of a node. By recording a database of fingerprints at known positions, a node can estimate its position by comparing its measurements with the database. Many fingerprint systems have been proposed that differ in scale, the signal characteristic that is recorded, how the fingerprints are collected, how the fingerprints are used and how the node estimates its position [12]. Fingerprinting is discussed in more detail in Chapter 6.

3.4 Indoor Localization

Of particular interest to us are indoor localization systems deployed into practice. One of the earliest indoor localization systems is the *Active Badge Location System* by Want et al. [33]. In this system people carry an Active Badge (see Figure 3.1) that transmits a unique code via infrared every 15 seconds. These periodic signals are received by sensors, which are placed at fixed positions within a building and relayed to the location manager software. The location manager software is able to provide information about the person's location to the requesting services and applications. Active Badge was able to localize people at room level.



Figure 3.1: Active Badge.

Due to the requirement for a fine-grained three-dimensional location system, which Active Badge could not supply, the *Bat Ultrasonic Location System* was developed [34]. The Bat System works similar as Active Badge, but instead of using infrared pulses, the Bat (see Figure 3.2) periodically transmits an ultrasound pulse and a radio message at the same time. Receivers located in the ceiling receive these messages and use TDoA to calculate the distance. The three-dimensional position of the Bat is calculated by a centralized system using trilateration. The Bat System was able to achieve an accuracy of around 14 centimeters.

The Cricket Indoor Location System by Priyantha [24] improves on the Bat System by introducing a decentralized approach. Cricket inverses the operation of the Bat



Figure 3.2: Ultrasonic Bat.

System, by letting the anchors transmit the ultrasound pulse and radio message. The mobile node then calculates its position using a least-squares minimization. With this approach, Cricket achieved an accuracy of 10 centimeters.

RADAR by Bahl and Padmanabhan [3] is a fingerprint-based system for locating and tracking users inside buildings. It operates in two phases: an offline data-collection phase and a real-time phase. In the offline data-collection phase, static nodes record the coordinates of a mobile node, along with the received signal-strength information (received signal-strength and signal-to-noise ratio) of packets (*beacons*) the mobile node periodically broadcasts. The combination of signal-strength information and coordinates of the mobile node (the *signature*) are stored in a central database.

In the real-time phase the central database is used to localize a mobile node. Just as in the offline data-collection phase, the static nodes measure the signal-strength information of the beacons the mobile node periodically broadcasts. The location of the mobile node is determined by averaging over the physical coordinates of the k -nearest neighbors in signal space of the observed signals.

On an experimental testbed, RADAR achieved an accuracy of 2 to 3 meters with ranges up to 40 meter. Since a significant effort is needed to construct the data set, a second approach was devised that uses a propagation model to reduce the dependency on empirical data. While this makes deployment much easier, the accuracy went down to 5 meters.

While RADAR seems like an ideal candidate for our target application, there are a few drawbacks. For each environment a significant effort is needed to construct the data set. In addition, this process needs to be repeated whenever the environment changes, for example, when anchor nodes or physical objects such as closets are moved. This approach is therefore not very robust.

RIPS and CAB

This chapter describes the two algorithms we selected from a review of the localization literature [5] and tested to see whether they are suitable for our target application. RIPS is discussed in Section 4.1, CAB in Section 4.2

4.1 RIPS

Radio interferometry is a novel approach in sensor node localization, first applied in the Radio Interferometric Positing System (RIPS) by Maróti et al. [21]. Traditional radio interferometry is based on two directional antennas measuring the radio signal from a single source and performing a cross correlation. This requires tunable directional antennas, very high sampling rates and high-precision time synchronization. Since these are generally not available in a WSN, the method is not directly applicable.

The idea behind RIPS is to have two transmitters that create an interference signal directly (see Figure 4.1). If the frequency of the signals are almost the same, the frequency of the interference signal is low enough to be measured by the hardware of a sensor node. Using the measured signal to calculate the node location directly requires tight time synchronization. Since this requires special hardware, which is generally not available, the relative phase offset of two receivers is used. By making multiple measurements in a network of at least 6 nodes, it is possible to calculate the relative position of the nodes in 2D.

In a field experiment RIPS achieved an average localization accuracy of 3 centimeters with a maximum error of 6 centimeters over ranges of up to 160 meters. Compared with other methods that use standard hardware this is quite impressive.

RIPS, however, does come at a price. Although the signal measurements takes place on the nodes themselves, the actual calculation of the positions takes place on a central computer. RIPS uses a genetic algorithm to solve the set of nonlinear equations that is too computational intensive for sensor nodes. In a 16-node network it took 80 minutes to localize all the nodes. Improved versions of RIPS such as inTrack [13], mTrack [15] and dTrack [14] reduce the computational complexity of the method and speed up the localization process significantly to around two seconds per node. The downside of reducing the computational complexity is that the accuracy went down from the original 3 – 6 centimeters in the best and worst case to 1.3 – 2.2 meters in the best and worst case.

The first versions of RIPS are not suitable for an indoor environment, because

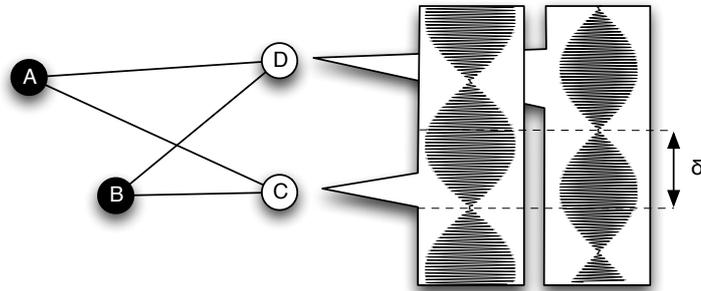


Figure 4.1: Radio interferometric ranging technique. Nodes A and B emit the interfering radio wave, nodes C and D measure the interfering frequency. The relative phase offset, denoted with δ , is used to calculate the in-between distances of sender and receiver.

multipath effects, which are typical for this environment, change the phase of a signal. Measuring the relative phase offset in an indoor environment is therefore not possible. A solution to this problem is to measure Doppler shifts, as introduced with dTrack. Multipath effects do not change the frequency of a signal. This approach, however, has only been tested in an open outdoor environment.

4.1.1 Feasibility Study

RIPS is an interesting technique because it can provide a high accuracy using standard hardware, a feature that many indoor localization algorithms do not have. In order to use RIPS in our target application, there are two questions that need to be answered. The first question is whether it is possible to use RIPS indoors. The earlier versions are not suitable, as explained in Section 4.1. The Doppler shift variant should not have this problem, but this has not been tested in practice.

The second question is whether it is possible to run RIPS on our target platform. The standard hardware that RIPS used were the MICA2 and the eXtreme Scale Mote (a ruggedized MICA2 clone). The CC1000 radio chip used on the MICA2 is quite different from the CC2420 used on our target platform (see Section 2.3). For RIPS to work, the radio has to meet two requirements. Firstly, the radio has to be able to send an unmodulated sine wave. In normal operation, the CC2420 sends messages using OQPSK modulation. It is possible to set the CC2420 in a transmitter test mode, in which it can transmit an unmodulated sine wave. So the CC2420 meets the first requirement.

The second requirement is that the radio needs a relative fine-grained frequency control. The fine-grained frequency control is necessary to be able to transmit two signals that have a low enough interfering frequency that can be measured. The CC1000 has a frequency resolution of 65 Hz. The CC2420 however, has a frequency resolution of 1 MHz, which produces an interfering frequency that is too high to be measured. RIPS is therefore unsuitable for our target platform.

Since we still wanted to test RIPS, we decided to compile and run the publicly available mTrack on TNodes. TNodes (see Figure 4.2) are coin-size MICA2 clones, developed by TNO. The only difference between the MICA2's that were used with

RIPS and the TNodes we are using is the operating frequency of the radio. The MICA2 operates in the 433 MHz ISM band, while the TNode uses the 868 MHz ISM band. It is not possible for the TNodes to use the 433 MHz band, because of regulations and the required changes in hardware (capacitors, resistors). Although this is a seemingly minor difference, this difference broke the protocol. RIPS uses a custom radio driver, which has a very precise timing for every phase of calibrating, tuning, transmitting and measuring. Changing the frequency of the radio changes these timings. Because of the convoluted source code of mTrack and the unavailability of the equipment to measure these timings, we could not port RIPS to the TNodes. RIPS is therefore not feasible for our target application.

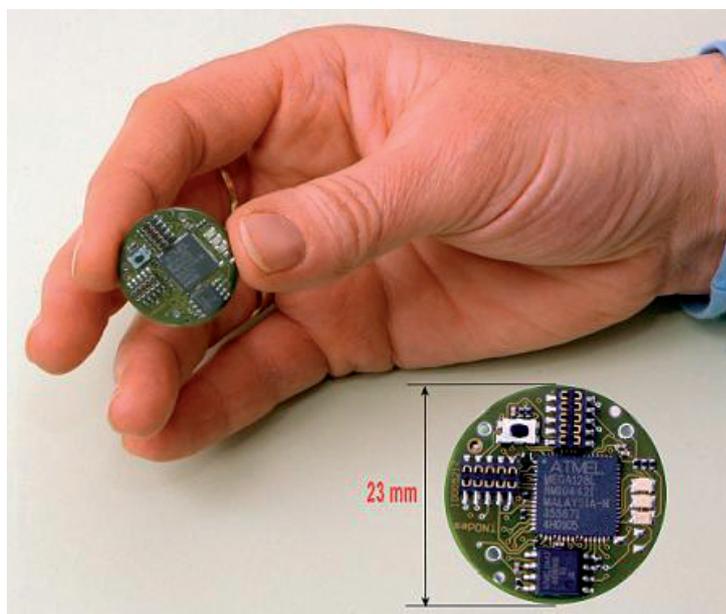


Figure 4.2: TNode.

4.2 CAB

Concentric Anchor Beacon (CAB) by Vivekanandan and Wong [32] is a range-free localization scheme, where anchors emit beacons at different power levels. Each beacon includes the anchor's id, location, the transmitted power level and the estimated maximum transmission distance. An unlocalized node then determines in which circular ring it is located with each anchor (see Figure 4.3). By taking the center of the intersections of the circular rings, a node can estimate its position. The advantages of CAB is that it requires only a small number of anchors, has a low computational complexity and does not require any communication between neighboring nodes.

The disadvantage of CAB is that it assumes a circular radio model. Although CAB performed well in simulations with an irregular radio propagation pattern [9], it has not been tested by an experiment.

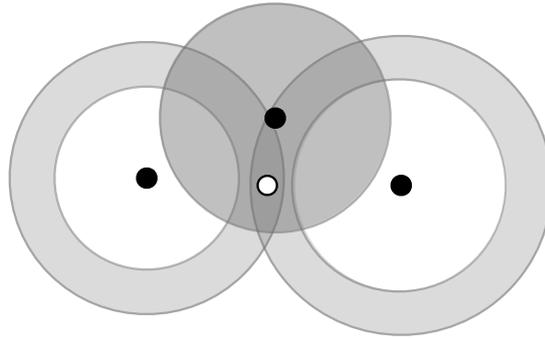


Figure 4.3: Example of localization using CAB. The black dots indicate the anchors, the grey circular rings indicate the range to the beacons, the white dot is the estimated location of the node.

4.2.1 Experimental Setup

To test CAB in practice, we decided to perform an experiment to validate one of the fundamental assumptions of CAB. CAB assumes that a beacon transmitted at a particular power level covers a certain area. To validate this assumption, we placed 8 Tmote Inventis on the floor in a hallway, 2 meters apart, covering a range of 14 meters (see Figure 4.4). One node acted as an anchor node, transmitting beacons at a regular interval, with increasing power level. The power level of the CC2420 can be configured using a 5-bit value in the transmit control register, where 31 equals 0 dBm.

The other nodes recorded the RSSI of the received beacons. When 1024 beacons were transmitted, the 7 receiving nodes were moved to cover the range of 16 until 28 meters and the experiment was repeated. In total a range of 50 meters was covered, which is the specified maximum distance for the Tmote Invent in an indoor environment. The experiment was repeated several times, during different parts of the day.

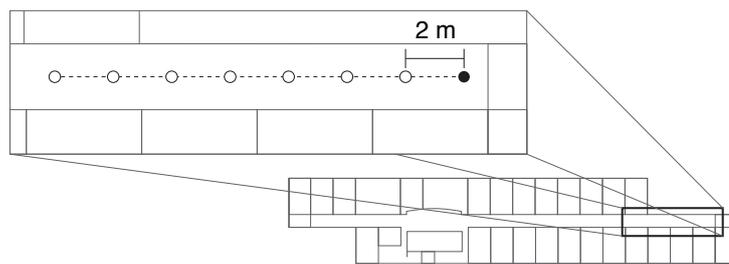


Figure 4.4: CAB experimental setup. The black dot is the anchor node, the white dots are the receiving nodes. The nodes lie 2 meters apart.

4.2.2 Results

The result of the experiment is shown in Figure 4.5. The figure shows the mean RSSI versus distance. While the 5-bit power level register should give us 32 different power

levels, we noticed that only 8 of them gave a different RSSI at the same distance. The data sheet of the CC2420 has a table with 8 register values and their corresponding output power in dBm and power consumption [30, pg. 51]. The 8 power levels specified in the data sheet corresponded with the 8 power levels that produced a different result in the experiments. Figure 4.5 therefore only contains these 8 power levels, even though all measurements were made on all 32 levels.

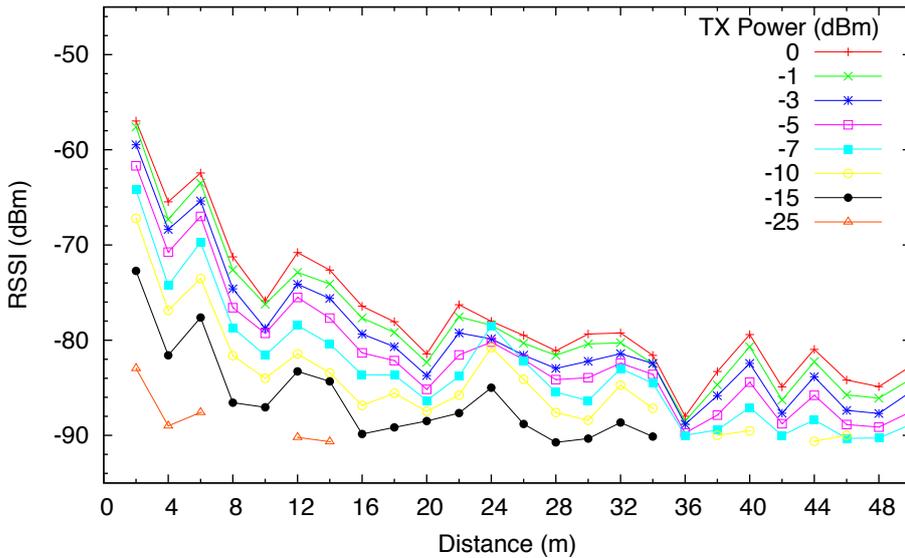


Figure 4.5: Mean RSSI versus distance with different power levels.

While the figure shows that the signal strength decreases exponentially with the distance, as is stated by the theory, the curve decreases very erratic, with a sudden drop at 36 meters. The erratic behavior is most likely due to multipath effects caused by the environment. We repeated the experiment several times, on different parts of the day with different nodes. The results were similar with respect to the drops.

With this experiment we wanted to validate the assumption in CAB that beacons transmitted at a certain power level cover a certain area. If we look, for instance, at beacons transmitted with a transmit power of -10 dBm this is clearly not the case. Beacons transmitted at this power level can be received up to 34 meters, between 38 and 40 meters, between 44 and 46 meters and at 50 meters. The area it covers is not continuous. We could solve this by using a threshold, but the same problem would arise with the other power levels.

Another problem is that most of the power levels are very close to each other. The idea of using different power levels was to have a finer granularity than when using centroid. From the 8 power levels, 6 power levels can reach up to 50 meters. Even the lowest power of -25 dBm has a range of at least 14 meters. This gives us an accuracy that is not much better than centroid. CAB is therefore not a suitable localization scheme for our target application.

Multiple Frequencies

This chapter describes the use of multiple frequencies to reduce multipath effects. Section 5.1 describes the use of multiple frequencies. Experimental setup and results are discussed in Section 5.2 and 5.3. Section 5.4 describes the proximity localization scheme, which is used as a test case for multiple frequencies.

5.1 Multiple Frequencies

As we have seen from the experiments with CAB (see Section 4.2.2), the environment has a major influence on the propagation of radio signals. Especially multipath effects that are typical for an indoor environment influence the radio signals. Since we only have the radio as means to measure the ‘distances’ between nodes and we do not want to add any additional hardware, we will have to find a way to reduce these multipath effects.

Ladha et al. [16] describe a technique of using multiple frequencies to mitigate multipath effects. The idea is that radio signals with different frequencies are influenced differently by the environment. Some of the radio signals at a certain frequency get dampened, while signals at another frequency get amplified. By simply averaging over the RSSI of signals, transmitted on multiple frequencies, multipath effect can be dramatically reduced.

The effects of using multiple frequencies has been tested by the authors in an indoor environment. These experiments however, were taken in an open office environment, where the nodes have a line of sight. In our target application the mobile nodes do not have a line of sight with the anchors.

We first performed a set of experiments to verify the effect of multiple frequencies. Since the results were promising, we tested the use of multiple frequencies on a simple proximity localization protocol.

5.2 Experimental Setup

The setup for the experiment is similar to the experiments performed with CAB (see Section 4.4). We placed 8 Tmote Invents on the floor in a hallway, 2 meters apart, covering a range of 14 meters. One node acted as an anchor node, transmitting beacons at a regular interval, with increasing frequency at maximum transmission power.

The frequency of the CC2420 can be configured using a 10-bit value in the frequency control register, with 1 MHz resolution. We did not use this fine-grained

frequency control for two reasons. Firstly, while the frequency control register has a range of 2048 – 3072 MHz, the hardware of the Tmote Invent is designed to operate in the range of 2400 – 2483.5 MHz. Secondly, the CC2420 has a channel bandwidth of 4 MHz. Using steps of 1 MHz results in overlapping channels that could interfere with each other. We therefore used the IEEE 802.15.4-2006 channel model [10], which specifies 16 channels within the 2.4 GHz band, in 5 MHz steps, numbered 11 through 26¹. This gives us a frequency range of 2405 – 2485 MHz. In this way the channels do not overlap and thus cannot interfere with each other.

The other nodes recorded the RSSI of the received beacons. While in the CAB experiment the nodes could just passively listen for beacons, the nodes now had to switch to the next channel after receiving a beacon. They did this until they had received 32 beacons per channel. In total a range of 32 meters was covered. The experiment was repeated several times, during different parts of the day.

5.3 Results

Figure 5.1a shows the mean RSSI versus distance for all 16 channels separately. The curve for every channel shows the same erratic behavior as we have seen in Figure 4.5. Some channels have an excellent RSSI at a certain range, while other channels have a very bad RSSI. If we look, for example, at 6 meters, beacons transmitted on channel 16 have an average RSSI of -67 dBm, while beacons transmitted on channel 21 have an average RSSI of -79 dBm.

Figure 5.1b shows the mean RSSI versus distance averaged over all channels. While the curve is not a smooth exponential decrease, it approaches the theory much better than using one channel. Using multiple frequencies does mitigate multipath effects. There are still a few irregularities, for example, at 12 meters where the hallway changes from glass windows on one side and office rooms on the other side to office rooms on both sides. Also from 26 meters to 30 meters there is an increase instead of a decrease in RSSI.

While this result suggests that the application of a propagation model would be feasible, constructing a robust propagation model for an indoor environment is still quite difficult. The main reason is that we would need a propagation model for every situation. A propagation model based on the data from the experiments in the hallway does not work for the nodes that are located in the ceiling space. The effect of obstacles like walls and cabinets should be included in the propagation model. In addition, we noticed small differences in the order of 1~2 dBm in RSSI between nodes at the same distance. Because of the inverse exponential relation between RSSI and distance, this small difference in RSSI could translate in an error in the order of tens of meters.

In conclusion, using multiple frequencies does mitigate multipath effects, but there are still a lot of factors that should be considered in constructing a robust propagation model for RSSI-based localization in an indoor environment.

5.4 Proximity

Since the use of multiple frequencies showed some promising results, we wanted to test its effect in a more real-world deployment. The previous experiments were taken

¹Channel 0 is in the 868 MHz band, channel 1 – 10 are in the 915 MHz band.

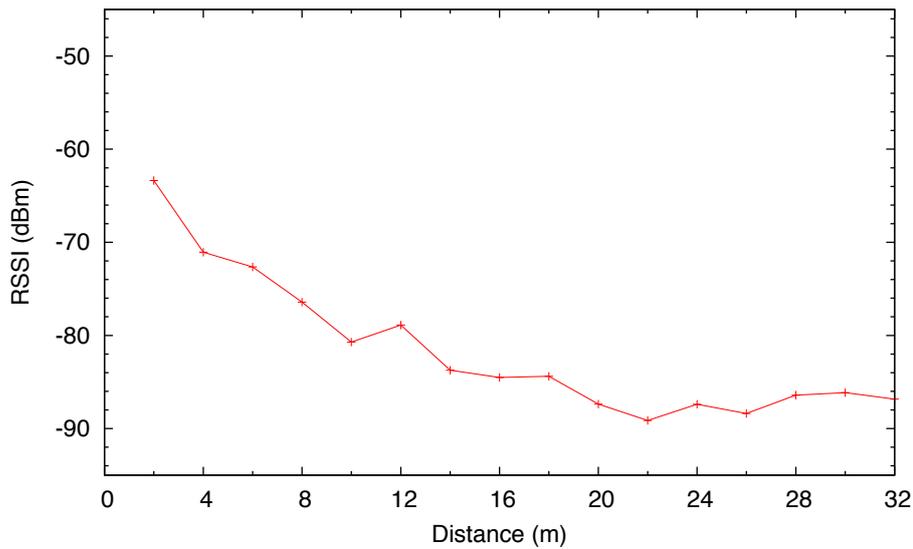
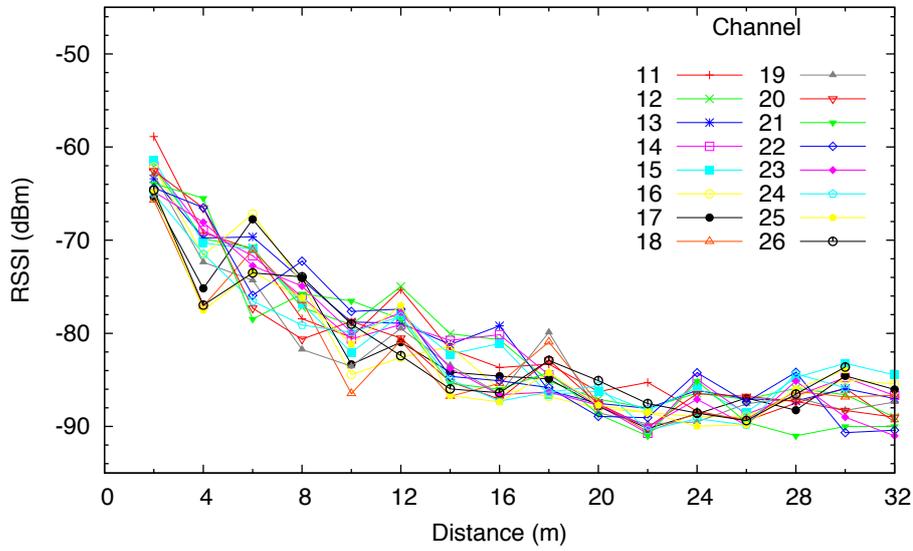


Figure 5.1: Multiple frequencies experimental results.

in a one-dimensional setup, where all nodes lie on one line. As a test case we use proximity localization. The idea of proximity localization is quite simple: the position of the node is the position of the strongest anchor.

5.4.1 Experimental Setup

For this experiment we used the 8 Tmote Sky nodes of the testbed as anchors. There is one mobile node that transmitted beacons with maximum power at a regular interval on increasing frequency. The mobile node is placed, preferably on a desk, in 6

locations in the 4 rooms that the anchors cover. At each location, the mobile node transmitted 4800 beacons distributed over 16 channels, which results in 300 beacons per channel. The beacons are transmitted at an interval of 31.25 milliseconds. Figure 5.2 shows the location of the anchors and the positions of the mobile node.

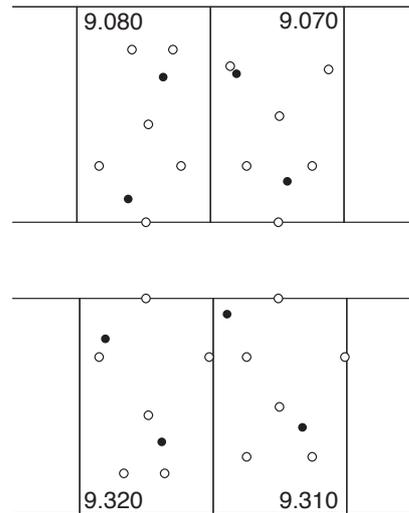


Figure 5.2: Proximity experimental setup. Black circles are the anchor nodes, white circles the positions of the mobile node.

The anchors send the RSSI of each received beacon to a central computer. While usually the anchors transmit the beacons and the mobile nodes listens, we chose to invert this scheme to simplify the scheduling problem. If the anchors were active transmitters, a schedule is needed to determine who transmits when at which channel. This schedule could either be predetermined or generated ad hoc. By letting the mobile node transmit at a regular interval, the anchors only have to switch to the next channel when they receive a beacon. Furthermore, we can now reach 8 anchors at the same time, while otherwise we need 8 beacons to get the RSSI for the 8 anchors.

The anchors initially listened on channel 11 for a beacon. As soon as they received a beacon, they switch to the next channel. To be resilient to single packet loss, the anchor switched to the second next channel if it did not receive a beacon within 1.5 times the beacon interval. After this the anchor did not switch to another channel anymore, unless it received a beacon.

5.4.2 Results

Figure 5.3 shows the positions of the mobile node, with a line from the position to the closest anchor. The closest anchor is defined as being the one that receives the most beacons with the strongest RSSI. In the ideal case this would mean that the closest anchor receives 4800 of these beacons. In most cases the closest anchor receives between 80 – 100% of the beacons, although there is a case where the closest anchor receives at most 48% of the strongest beacons.

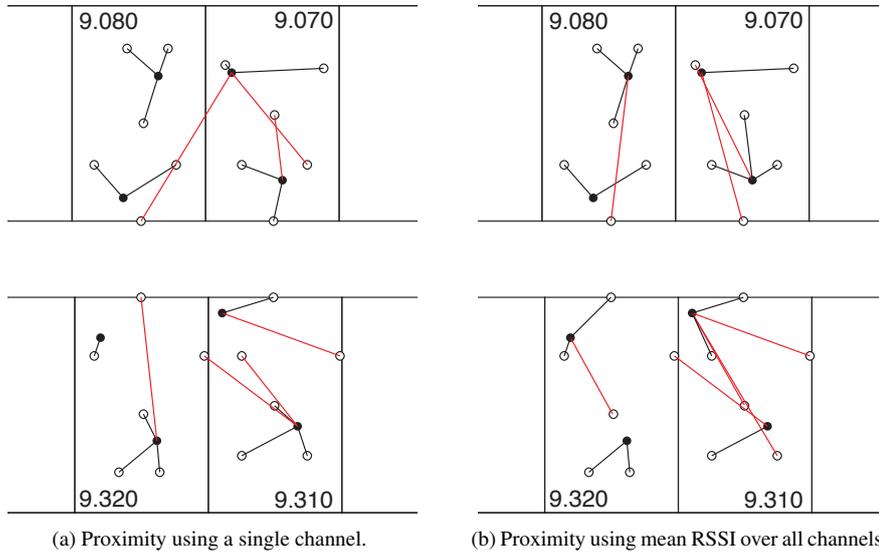


Figure 5.3: Proximity experimental results. Black circles are the anchors, white circles the positions of the mobile node. The lines indicate which anchor is the ‘closest’ for that particular position. A red line indicates an incorrect association.

Figure 5.3a shows the result of using only a single channel for proximity localization. What we can see from the figure is that the anchor receiving the strongest beacons is not necessarily the closest anchor. In all of the rooms, there is at least one position that is associated with an anchor at the other end of the room. Even in two cases, the position is associated with an anchor in another room. In total there are 7 positions associated with the wrong anchor.

Figure 5.3b shows the result of using the mean RSSI over all channels for proximity localization. Compared to using a single channel for proximity, the results are actually a bit worse. In total there are now 8 incorrect associations, one more than in the previous experiments. What is noticeable is that most of these incorrect associations are different from the single channel case. Some positions that were incorrectly associated in the previous case are now associated with correct anchors and vice versa.

So using multiple frequencies does not improve the accuracy of proximity localization. Actually, the closest anchor is not necessarily the one that receives the strongest beacons. Using multiple frequencies for proximity localization however, does provide one improvement. The average number of beacons that the strongest anchor receives over time improved from 88% to 94%. This means that while the position can be associated with the wrong anchor, it is more resilient to temporal perturbations.

While the use of multiple frequencies slightly improves the robustness of proximity localization, the acquired accuracy is not good enough for our target application. With this approach a node can only be localized in a room where there is at least an anchor node. In addition the best accuracy that can be achieved with this approach is room level, while our goal is sub-room level localization.

Fingerprinting

As we have seen in the previous chapters, using RSSI to measure the inter-node distance directly via a propagation model or indirectly by using CAB, or to determine proximity does not provide the required accuracy for the target application. In addition, these approaches often require a uniform deployment of anchors to cover a certain area. In order to provide an accurate localization method using RSSI we need a different approach. We previously dismissed fingerprinting as being too labor intensive and not resilient to environmental changes [5]. Since the use of multiple frequencies can improve the robustness of a localization method, we take a second look at fingerprinting to see how it performs in practice.

Section 6.1 describes fingerprinting itself. Measurement setup and algorithms are discussed in Section 6.2. The results of the experiments are presented and discussed in Section 6.3 and Section 6.4.

6.1 Fingerprinting

The core idea of fingerprinting is that the position of the mobile node correlates with one or more signal characteristics. Often RSSI and anchor ID are used, but characteristics like reception rate, link quality indication and signal-to-noise-ratio have also been used. By measuring these signal characteristics at known positions and storing them in a database, the position of a mobile node can be approximated by measuring the same signal characteristics and comparing it to the database. Fingerprinting does not depend on theoretical signal propagation properties but actually exploits the erratic behavior of signal propagation.

6.1.1 Setups

There are several setups proposed for fingerprinting. One of the most common setups is an infrastructure-based approach. Several fixed base stations transmit or receive beacons, and one or more mobile nodes receive or transmit beacons. Often a central computer is used to store the database. Less common is an infrastructure-less approach, which only uses mobile nodes and distributes the database across the nodes. Given the requirements of the target application, we used an infrastructure-based setup, with the nodes of the testbed as fixed base stations.

Storing the database on either a central computer, the base stations, or on the mobile node itself has its advantages and disadvantages in terms of scalability, privacy,

resource-requirements etc. Storing the database on the mobile node is attractive because of the inherent support for privacy. On the other hand, mobile nodes are often resource-poor and can therefore only store a part of the database. Storing the database on a central computer on the other hand, can be a problem in terms of privacy, since the mobile node has to communicate with the server to determine its positions. We chose to store the database on the central computer, because we have a central computer in our setup and privacy is not a requirement of the target application.

6.1.2 Estimation Methods

A central part in a fingerprinting system is the estimation method used for predicting the position of the mobile node from the measured signal characteristics and the available database. Estimation methods can be globally divided into two categories: deterministic and probabilistic. A deterministic estimation method uses the entries in the database only by their values, while a probabilistic approach considers these values as part of a random process. For the experiments we used the deterministic k -nearest neighbor search and the probabilistic probability maps as estimation methods. A description of other estimation methods can be found in [12].

k -Nearest Neighbor Search The k -nearest neighbor search (k -NNS) is one of the most often used deterministic estimation methods. k -NNS calculates distances in signal space between the measured sample and the samples in the database. The k database samples with the shortest distance are then used for the position estimate, which is often the weighted centroid of the positions. For measuring the distance the Euclidean distance metric is often used ($\sqrt{(p_1 - q_1)^2 + \dots + (p_n - q_n)^2}$).

Probability Maps Probability maps (PM) is a probabilistic estimation method in which a measured sample is represented by a normal distribution. For each anchor, a *radio map* is calculated by fitting the database samples to a uniform grid covering the deployment area. Then for each anchor a *probability map* is calculated by applying the probability density function (*pdf*) to the radio map. By combining these probability maps using multiplication, the position of the node can be approximated as the coordinates with the highest probability.

6.2 Experimental Setup

This section describes the experimental setup. Section 6.2.1 describes how the measurements were done. How the fingerprint databases are constructed and an outline of the algorithms is given in Section 6.2.2 and 6.2.3.

6.2.1 Measurement Setup

The measurement setup is similar to the proximity experiment as described in Section 5.4.1. We used the same 24 positions and the same measurement approach. In addition, we took measurements in rooms where there are no anchors and in the hallway. In total we added 53 new positions, for a total of 77 positions. Figure 6.1 shows the location of the anchors and the positions of the mobile node.

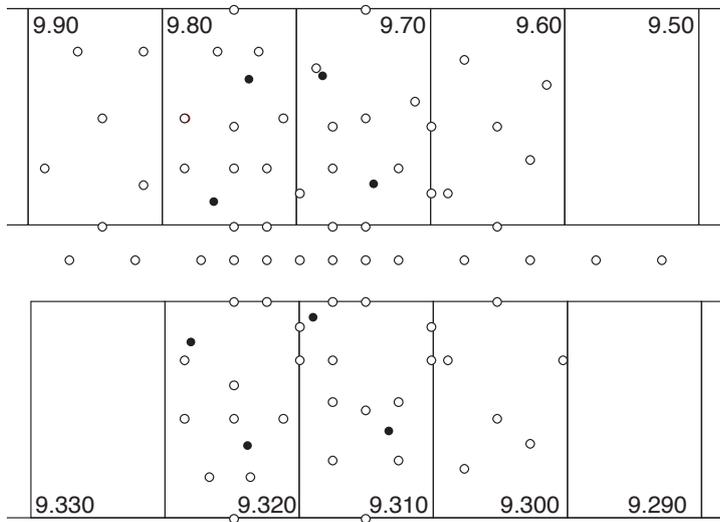


Figure 6.1: Fingerprinting measurement setup. Black circles are the anchor nodes, white circles the positions of the mobile node.

6.2.2 Database Setup

From the measured data, we used for every position 70% of the data to construct the database and 30% as input for evaluation. We construct three different databases. The first is *baseline*, which uses the data from only one channel. The purpose of this database is to evaluate whether using multiple frequencies provides any improvement. The second database is *combined*, where per position the samples are averaged over *all* channels. This results in a database with 77 (x, y) vectors and corresponding 8-dimensional RSSI vectors, since there are 8 anchors in the setup. The last database is *separated*, where per position the samples are averaged *per* channel. This gives a database with 1232 $(x, y, channel)$ vectors with corresponding 8-dimensional RSSI vector. The general consensus in fingerprinting systems is the more different signal characteristics, the better. With this approach we can also use the channel information in the calculations.

6.2.3 Algorithms Outline

The input for all the algorithms is an array of s samples. A sample is defined as being an 8-dimensional vector with the RSSI of one beacon, received by at least one anchor and the channel on which the beacon was transmitted. In the ideal case, all 8 anchors receive the beacon. If an anchor has not received the beacon, the value is ‘undefined’ and ignored when calculating sums and means. For all the algorithms, the vector mean of the s samples is calculated, resulting in an 8-dimensional RSSI vector, with the mean RSSI for each anchor.

k -NNS An outline of the k -NNS algorithm for *combined* is shown in Figure 6.2. From the averaged sample the Euclidean distance is calculated for each of the 77

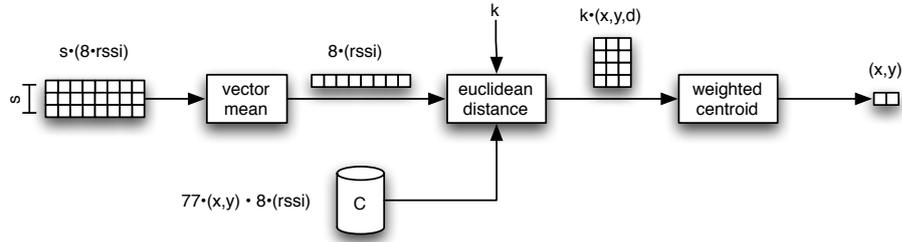


Figure 6.2: k -Nearest Neighbor Search *combined* algorithm outline.

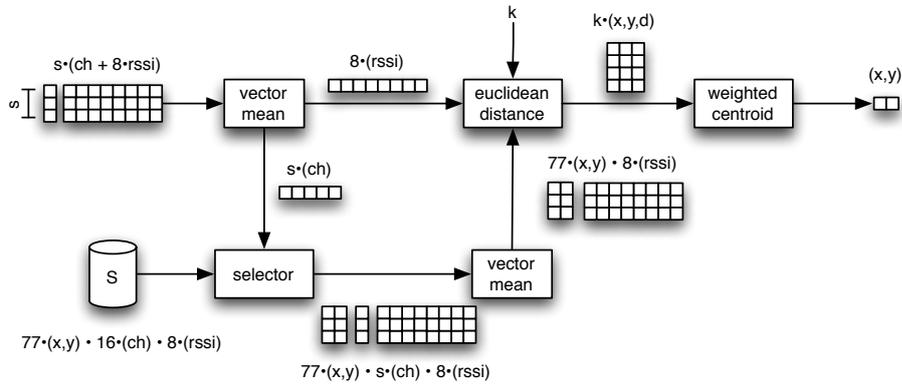


Figure 6.3: k -Nearest Neighbor Search *separated* algorithm outline.

RSSI vectors in the database. The output is an array of 77 (x, y) coordinates and the distance d from the averaged sample. The k samples with lowest distance d are used in a weighted centroid with weights $\frac{1}{d}$ to calculate the position estimate. If the distance d is 0, the (x, y) coordinate is used as the position estimate.

An outline of the k -NNS for *separated* is shown in Figure 6.3. The algorithm for k -NNS *separated* is similar to that of k -NNS *combined*. The main difference is what data from the database is used to calculate the Euclidean distances. In the case of k -NNS all the available data is used. In the case of k -NNS *separated*, a subset of the database is used based on which channels the beacons were received. This subset is averaged per position, resulting in an array of 77 (x, y) coordinates and the corresponding 8-dimensional averaged RSSI vector. The remainder of the algorithm is the same as k -NNS *combined*.

The algorithm for k -NNS *baseline* is the same as the algorithm for k -NNS *combined*.

Probability Maps An outline of the PM algorithm for *combined* is shown in Figure 6.4. For each anchor a radio map is generated by fitting the database samples from that anchor to a uniform $u \times v$ grid using cubic interpolation. In addition for each radio map, a correction factor γ is calculated, which is defined as $\gamma = \frac{1}{(\sigma_{rm})^\alpha}$, where σ_{rm} is the standard deviation of the radio map. For every anchor, a probability map

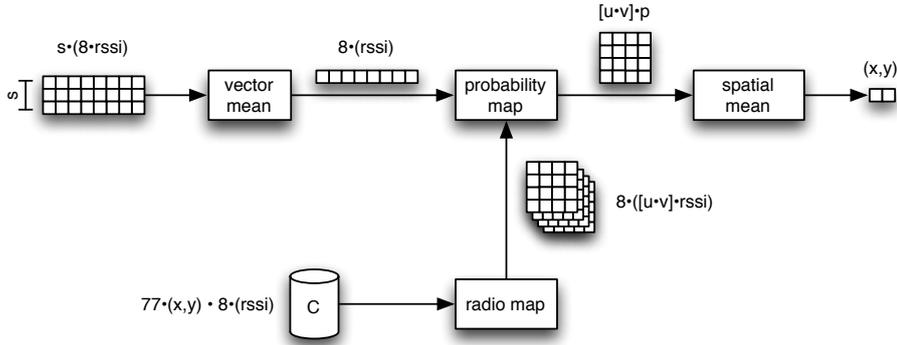


Figure 6.4: Probability Maps *combined* algorithm outline.

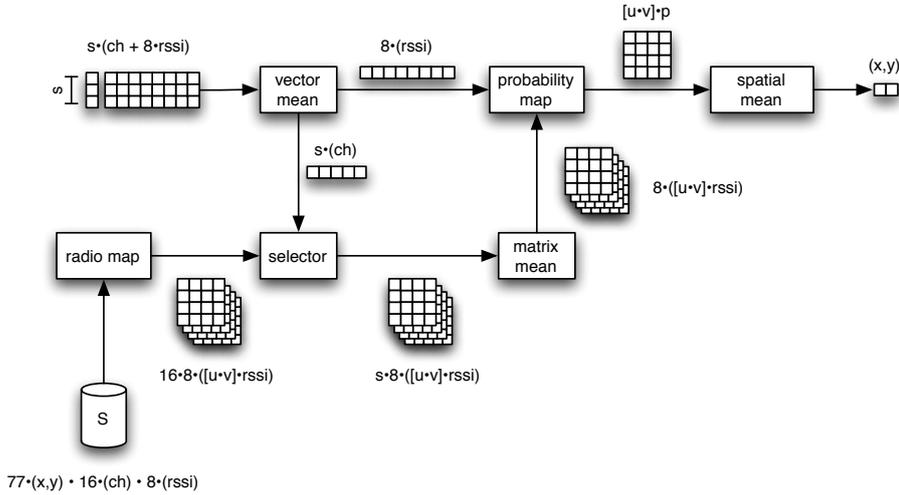


Figure 6.5: Probability Maps *separated* algorithm outline.

is calculated, using a normal distribution with $\mu = rssi_n$ and $\sigma = \beta \|\gamma_n\|$, where $rssi_n$ is the sample's mean RSSI of anchor n and $\|\gamma_n\|$ the normalized standard deviation of the radio map for anchor n . For our situation, we determined empirically that $\alpha = 3$ and $\beta = 3$ are good values. All the probability maps are then multiplied and from this single map the spatial mean is calculated. While initially we used the spatial maximum as a position estimate, using the spatial mean produced a much lower error.

An outline of the PM algorithm for *separated* is shown in Figure 6.5. The algorithm is similar to *k*-NNS *separated*. We also use a selector to only select the radio maps with the same channel. Then for every anchor the radio maps are matrix averaged. The remainder of the algorithm is the same as with PM *combined*.

The algorithm for PM *baseline* is the same as the algorithm for PM *combined*.

6.3 Results

We gathered almost 3 million raw samples. From these samples we constructed three different databases as described in Section 6.2.2. We used 30% of the measured data as input for evaluation. For every test vector, we excluded the vector from the database to emulate the real-time localization process. The results are therefore somewhat conservative. The error is defined as the Euclidean distance between the real position and the estimated position. For evaluation we use the mean error. The results for k -NNS and PM are discussed in Section 6.3.1 and 6.3.2.

6.3.1 k -Nearest Neighbor Search

Figure 6.6 shows the mean error when applying k -NNS on a single channel for $k = 4$. Other values of k produced similar shapes in terms of curve and channel order. There are two important observations that we can make from this figure. Firstly using more than 30 samples does not improve the mean error that much. If we look at channel 26, the difference in error between using 45 samples and 90 samples is 1 centimeter.

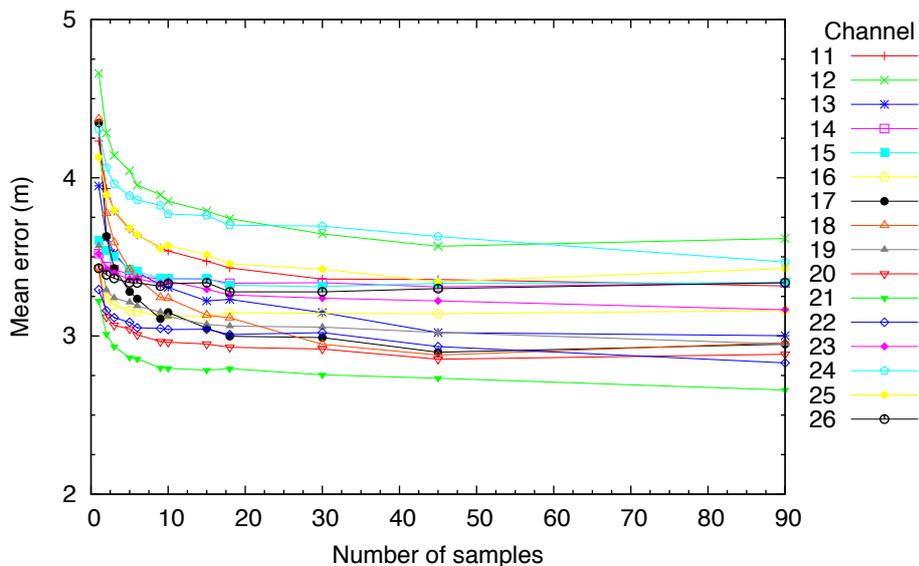


Figure 6.6: Mean error vs. the number of samples for k -NNS *baseline*, $k = 4$. Please note that the y-axis starts at 2.

The second observation we can make is that there is a significant difference between individual channels. Between the best channel (21) and the worst channel (12) there is a difference in mean error of at least 1 meter. We suspect that interference of, for instance, Wi-Fi and Bluetooth is the cause of this effect. For the remainder of the evaluation, we use channel 21 for *baseline*.

In order to evaluate the performance of k -NNS, we first need to determine the optimal number of neighbors. Figure 6.7 shows the number of neighbors vs. the mean error with $s = 30$. What is noticeable is that at around 4 neighbors the mean error increases. This effect is caused by the fact that signal space is not equal to the physical

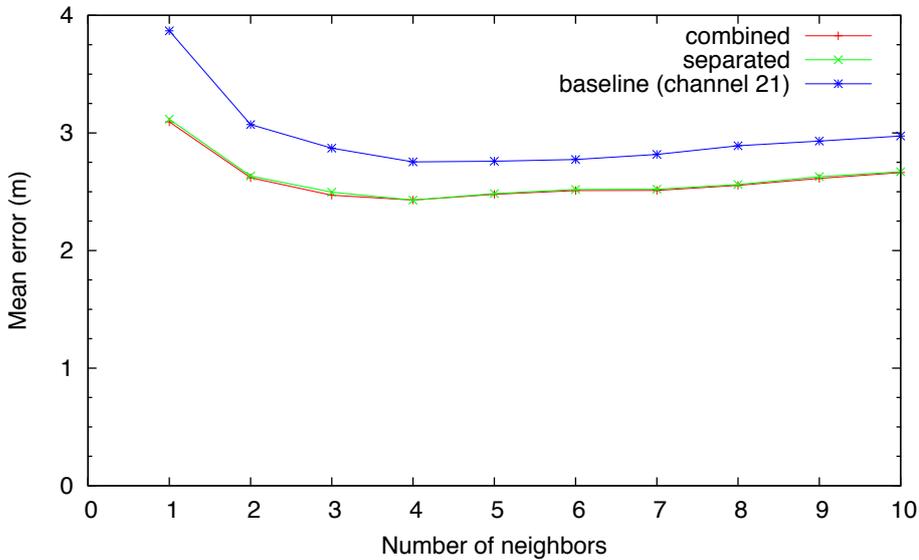


Figure 6.7: Mean error vs. the number of neighbors for k -NNS, $s = 30$.

space. Positions in physical space which are close to each other, do not necessarily have to be close to each other in signal space. This effect is also described by Bahl and Padmanabhan [3]. Because 4 neighbors produced the best result we use this value for the remainder of the evaluation.

Figure 6.7 also shows that for $k = 4$, *combined* and *separated* reduce the mean error by 0.3 meter compared to the best *baseline*. On the other hand, using *separated* does not provide an improvement over *combined*.

Figure 6.8 shows mean error vs. the number of samples. As we have seen in Figure 6.6, using more than 30 samples for *combined* and *separated* also does not improve the mean error that much. What is noticeable is that *separated* actually performs worse than *baseline* up until 5 samples. The reason for this is that *separated* uses all the channels, while *baseline* only uses the best channel. Averaging reduces the negative impact of the noisy channels on the mean error. So for *separated* to perform better than *baseline* we need at least 6 channels.

6.3.2 Probability Maps

By plotting a contour plot of the radio maps which are generated as part of the PM algorithm we get a nice view of the distribution of RSSI over the deployment area. Figure 6.9a and Figure 6.9b show the radio map of anchors 3 and 2. While the shape of the RSSI levels around anchor 3 are somehow circular, they get more irregular further away from the anchor. The radio map of anchor 2 is even worse, showing two peaks of which one lies near the anchor. This is exactly the kind of irregularity we can exploit with fingerprinting to identify the position of a node.

Figure 6.10a and Figure 6.10b show the radio map of channels 11 and 22 of anchor 8. Once again we can see that there is a significant difference between channels.

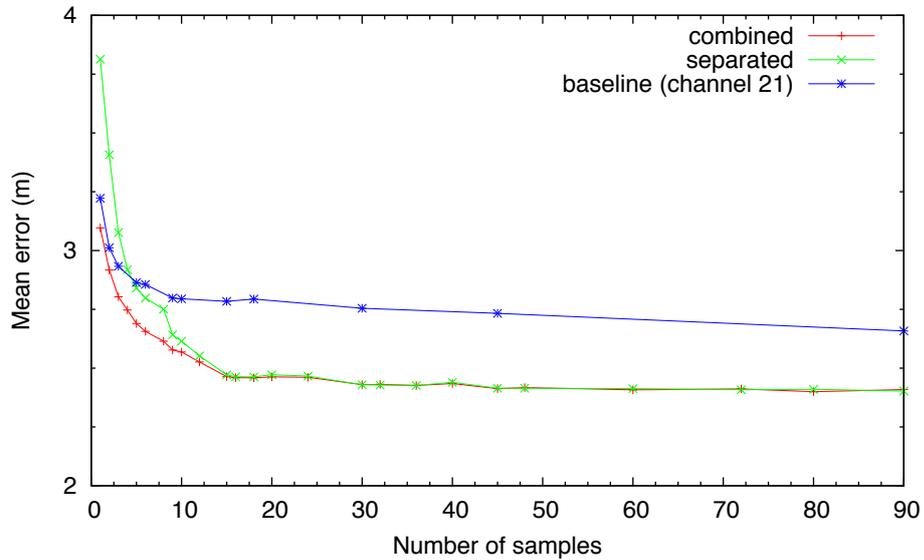


Figure 6.8: Mean error vs. the number of samples for k -NNS, $k = 4$. Please note that the y-axis starts at 2.

Figure 6.11 shows the mean error when applying PM on a single channel. As with k -NNS *baseline*, there is a significant difference between channels. The mean error between the best channel (21) and the worst channel (26) is 0.75 meter. Also, using more than 30 samples does not reduce the mean error that much.

Figure 6.12 shows the mean error vs. the number of samples for PM *combined*, *separated* and *baseline* on channel 21. As with k -NNS we see that *separated* initially performs worse than *baseline*. The main difference is that when using 2 or more samples it performs better, while with k -NNS we needed at least 6 samples. The cause of this effect is the averaging that was already done when generating the radio maps.

6.4 Discussion

Table 6.1 provides an overview of the accuracy of k -NNS and PM with *baseline*, *combined* and *separated*. The table contains the mean error as well as the 90th percentile error, which gives an indication on the number of outliers. If we look at the accuracy of *baseline*, it does not matter which estimation method we use. Both k -NNS and PM have a mean error of 2.75 meter. Although the 90th percentile error of PM is 5 centimeters more, this difference is negligible. Fingerprinting without the use of multiple frequencies therefore meets the requirement of an accuracy of 3 meters as defined in Section 2.4.

Using multiple frequencies, however, does offer a much better accuracy with little effort. For k -NNS the accuracy increases by 12% and for PM by 20%. In addition, we do not have to choose which channel we want to use. While channel 21 produced

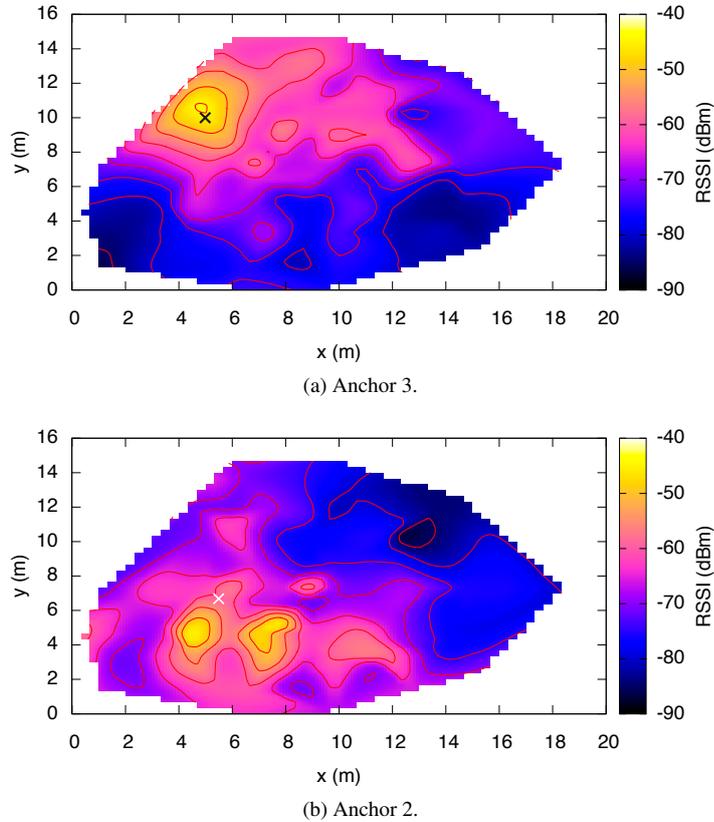


Figure 6.9: Radio map of two anchors. The cross indicates the position of the anchor.

the best results in our experiments, in other deployments it is possible that a different channel performs better. In terms of storage, *baseline* and *combined* require the same amount of space. Only *separated* requires 16 times more storage space, since channels are stored separately.

Using *separated*, however, does not provide the better accuracy over *combined* as we had expected. The mean error is either the same as with *k*-NNS or slightly higher as with PM. The only improvement *separated* has over *combined* is a slightly lower 90th percentile error. It is the question whether this improvement is worth the additional storage required.

In terms of latency, the whole localization process should take no more than 3 seconds as defined in Section 2.4. With the current implementation, recording one sample takes around 31.25 milliseconds. Recording 30 samples therefore takes 937.5 milliseconds. Table 6.2 provides an overview of the computation time needed per averaged sample for *k*-NNS and PM with *baseline*, *combined* and *separated*. The algorithms were run on a computer with an Intel Core 2 Duo 2.4 GHz processor and 4 GB RAM. The localization algorithms are single threaded. Independent of what estimation method or database setup we use, a position can be estimated in under 1 second. Consequently we amply meet the latency requirement.

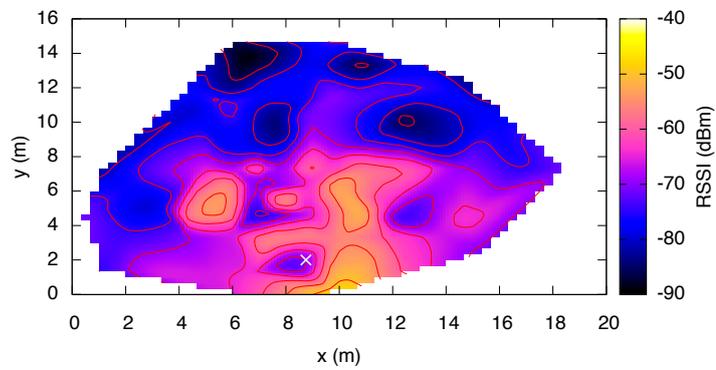
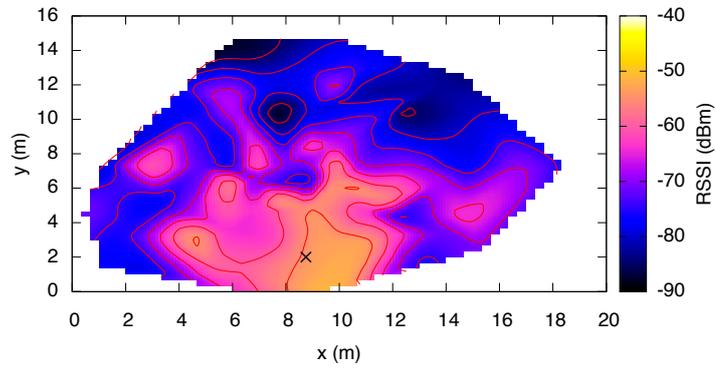


Figure 6.10: Radio map of anchor 8 on two different channels. The cross indicates the position of the anchor.

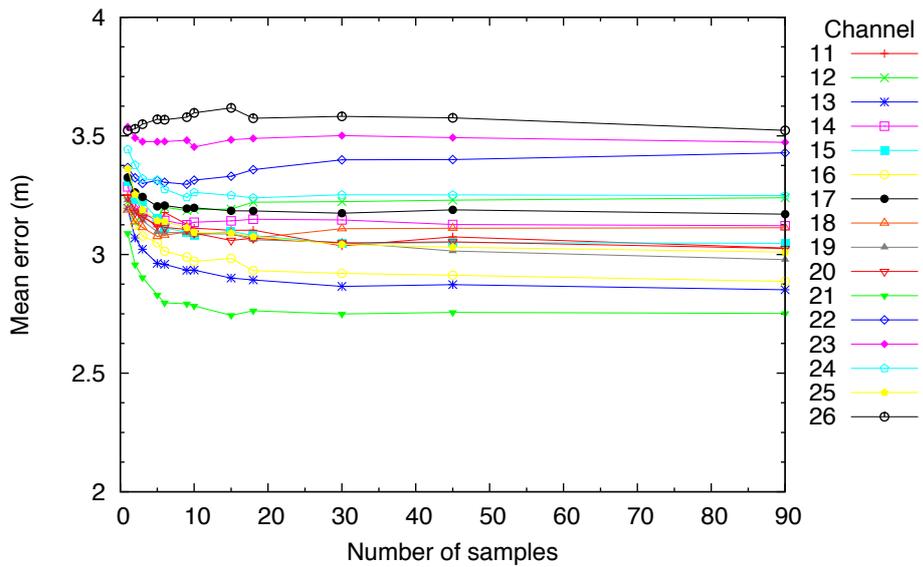


Figure 6.11: Mean error vs. the number of samples for PM baseline.

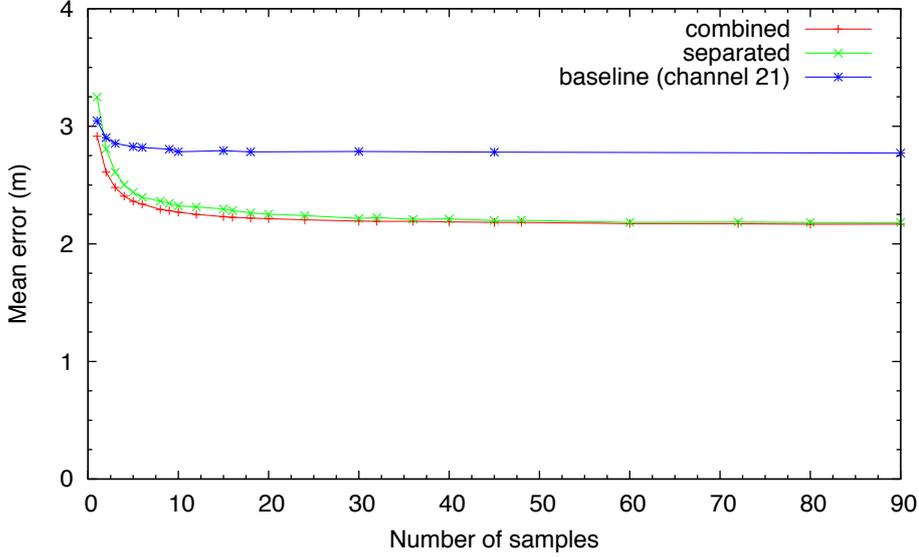


Figure 6.12: Mean error vs. the number of samples for PM *combined*, *separated* and *baseline* (channel 21).

Table 6.1: Overview of the mean error of k -NNS and PM for *baseline*, *combined* and *separated*, with $k = 4$, $s = 30$. The 90th percentile error is given in parentheses.

	<i>baseline</i> (ch 21)	<i>combined</i>	<i>separated</i>
k -NNS	2.75 m (4.55 m)	2.43 m (4.56 m)	2.43 m (4.47 m)
PM	2.75 m (4.60 m)	2.19 m (3.75 m)	2.22 m (3.71 m)

Table 6.2: Computation time per averaged sample, with $k = 4$, $s = 30$.

	<i>baseline</i> (ch 21)	<i>combined</i>	<i>separated</i>
k -NNS	2.18 ms	2.14 ms	37.1 ms
PM	6.04 ms	6.22 ms	41.1 ms

Conclusions and Future Work

7.1 Conclusions

The goal of this thesis was to create an application that can determine the location of a mobile node in indoor environments. The challenge was to achieve a sub-room level accuracy of around 3 meters, without any additional hardware. As a secondary goal we wanted to get practical experience with indoor localization.

From a review of the localization literature, we selected two algorithms: Radio Interferometric Positioning System (RIPS) and Concentric Anchor Beacons (CAB). RIPS was selected since it was tested in practice, used no additional hardware besides the radio and achieved a remarkable accuracy of 3 centimeters. CAB was selected as it is a range-free localization algorithm and for its simplicity.

Regarding RIPS there were two questions that needed to be answered. Firstly, RIPS was only tested outside in an open field. There were known problems with using RIPS in an indoor environment, due to multipath effects. A solution has been proposed in the form of measuring Doppler shifts instead of phase shifts, but this has not yet been tested in practice. The second question was whether RIPS could be applied to our target platform. The radio used in the experiments with RIPS was a CC1000, while we wanted to use a CC2420 radio. Although the CC2420 met some of the fundamental requirements, it was not accurate enough to be used with RIPS. We did try to use RIPS with nodes that have a CC1000 radio, but the hardware we had at our disposal differed too much in terms of operating frequency. The source code was too convoluted to port to our hardware. RIPS was therefore not a viable option.

The second candidate, the CAB algorithm, was only tested in simulation. One of the first steps was to test whether one of the fundamental assumptions of CAB holds in practical settings. CAB broadcasts beacons with different power levels and with the effective range of that particular setting. With this, a node can calculate in which ‘ring’ from the anchor it resides. During a practical experiment, we discovered that this assumption did not hold. Beacons that were transmitted on a lower power level did have a smaller range compared to beacons transmitted with a higher power level. There were however gaps in the range where beacons could, for example, be received up to 34 meters, between 38 and 40 meters, between 44 and 46 meters and on 50 meters. CAB was therefore also not a viable option.

We learned from these experiments that multipath effects have a major influence on the radio signals. If we want to use signal strength based localization, we need a method to reduce these multipath effects. We chose a technique called ‘multiple

frequencies' [16]. By transmitting not on a single channel, but on multiple frequencies and simply averaging over the RSSI of the received beacons, multipath effects can be significantly reduced. Rather than modeling these multipath effects, we could use these effects to improve the performance. We verified this effect with an experiment that showed some promising results.

The use of multiple frequencies was tested in a practical setting using a simple proximity algorithm. Unfortunately, the accuracy did not improve as we expected, but was actually a bit worse. One advantage of using multiple frequencies though, was the increased robustness. Over time, the estimated position was more consistent than when using a single channel.

From these experiments we concluded that localization algorithms that use signal strength as a direct distance metric are not feasible for our target application. Additionally, these methods rely on setting up a proper propagation model, which is quite difficult for an indoor environment.

During the literature review we dismissed the technique of fingerprinting for being too labor intensive and not robust enough. Since multiple frequencies can improve the robustness, we had a second look at fingerprinting to see how it performs in practice. We evaluated two different estimation methods: the deterministic k -nearest neighbor search and the probabilistic probability maps. Since we used multiple frequencies, we devised two approaches in storing the fingerprint database. One approach where per position the RSSI values are averaged over all channels, resulting in a relative small database, comparable to using a single channel. The other approach was to consider the channels separately, resulting in relative large database. The general consensus in fingerprint systems is that the more signal characterizes available, the better the performance. We therefore expected that the large database would perform better.

With experiments in an office setting we discovered that using a single channel for fingerprinting already met the requirements of the target application, if the right channel is chosen. Using multiple frequency, however, does improve the accuracy considerably. In addition, it provided a better robustness. One noticeable result was that the small database performed better than the large database in terms of accuracy. The extra channel information did only slightly improve the robustness for temporal disturbances. In terms of latency, averaging over more than 30 samples did not improve the accuracy much anymore. Recording 30 samples takes around 940 milliseconds and the position calculations are in the order of milliseconds. The whole localization process therefore can be done within 1 second, which is clearly below the 3 seconds that was required for the target application. The mean error was 2.19 meters, which is below the required 3 meters.

Although constructing the fingerprint database is a labor intensive process, it can be simplified by applying 'kriging'. Kriging is a geospatial interpolation technique that is mainly used in the mining industry. Some initial research in this direction has been done by Li et al. [19], showing some very promising results. Kriging can significantly reduce the number of samples needed and increase the accuracy.

7.2 Future Work

During the writing of this thesis, there were several directions which could not be explored due to time constraints. Despite all the hardware dependencies, RIPS is still an interesting technique. A direction for future work could be to get a radio with Doppler support. For example, a radio with two antennas, or by connecting two radios

to a node. While Doppler limits the localization algorithm to moving objects, this is not a problem for localization in a mobile scenario. Some work in this direction has already been done by Sallai et al. [25].

A completely different approach, which can also be achieved by using a radio with two antennas, is beamforming. Beamforming is a signal processing technique where the directionality of a signal is changed by transmitting two radio waves with different phase and amplitude. With this approach a radio signal can be electronically steered in a certain direction, without using mechanically rotation directional antennas or sectored antennas. This opens up a whole new range of localization algorithms. A localization algorithm such as SeRLoc [18], which uses sectored antennas, can be used on standard hardware.

A subject that is not touched in this thesis, but comes to mind when one talks about navigating people is privacy. In the current setup, all the beacons are send to a central computer that estimates the position of the mobile node. This could be undesirable when navigating people. There are some localization algorithms that take privacy in consideration, like for example Cricket [28], but often privacy is not considered. A solution to this problem is that each node computes its position based on the infrastructure.

Bibliography

- [1] R. Abielmona and V. Groza. Indoor sensor networks: Localization schemes. In *CCECE 2007: Canadian Conference on Electrical and Computer Engineering, 2007*, pages 1078–1081, April 2007. doi: 10.1109/CCECE.2007.275.
- [2] J. Bachrach and C. Taylor. *Handbook of Sensor Networks: Algorithms and Architectures*, chapter Localization in Sensor Networks, pages 277–310. Wiley Series on Parallel and Distributed Computing. Wiley-Interscience, Hoboken, November 2005. ISBN 0-471-68472-4. doi: 10.1002/047174414X.ch9.
- [3] P. Bahl and V. N. Padmanabhan. RADAR: an in-building RF-based user location and tracking system. *IEEE Infocom 2000: Proceedings of the 19th annual joint conference of the IEEE Computer and Communications Societies*, 2:775–784, March 2000. doi: 10.1109/INFCOM.2000.832252.
- [4] M. Battelli and S. Basagni. Localization for wireless sensor networks: Protocols and perspectives. In *CCECE 2007: Canadian Conference on Electrical and Computer Engineering, 2007*, pages 1074–1077, April 2007. doi: 10.1109/CCECE.2007.274.
- [5] M. C. Bor. Tracking in wireless sensor networks. research report, April 2008.
- [6] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low-cost outdoor localization for very small devices. *IEEE Personal Communications*, 7(5):28–34, October 2000. ISSN 1070-9916. doi: 10.1109/98.878533.
- [7] T. L. Dinh, W. Hu, P. Sikka, P. Corke, L. Overs, and S. Brosnan. Design and deployment of a remote robust sensor network: Experiences from an outdoor water quality monitoring network. In *LCN 2007: 32nd IEEE Conference on Local Computer Networks, 2007*, pages 799–806, October 2007. ISBN 978-0-7695-3000-0. doi: 10.1109/LCN.2007.49.
- [8] F. Franceschini, M. Galetto, D. Maisano, and L. Mastrogiacomo. A review of localization algorithms for distributed wireless sensor networks in manufacturing. *International Journal of Computer Integrated Manufacturing*, pages 1–19, June 2007. doi: 10.1080/09511920601182217.
- [9] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher. Range-free localization schemes for large scale sensor networks. In D. B. Johnson, A. D. Joseph, and N. H. Vaidya, editors, *MobiCom 2003: Proceedings of the 9th annual international conference on Mobile Computing and Networking*, pages 81–95, San Diego, CA, USA, September 2003. ACM. ISBN 1-58113-753-2. doi: 10.1145/938985.938995.

- [10] *IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*. IEEE Computer Society, 3 Park Avenue, New York, NY 10016-5997, USA, 802.15.4-2006 edition, September 2006. URL <http://standards.ieee.org/getieee802/download/802.15.4-2006.pdf>.
- [11] J. M. Kahn, R. H. Katz, and K. S. J. Pister. Next century challenges: mobile networking for “Smart Dust”. In *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 271–278, New York, NY, USA, 1999. ACM. ISBN 1-58113-142-9. doi: 10.1145/313451.313558.
- [12] M. B. Kjærsgaard. A taxonomy for radio location fingerprinting. In J. Hightower, B. Schiele, and T. Strang, editors, *Location- and Context-Awareness*, volume 4718 of *Lecture Notes in Computer Science*, pages 139–156. Springer Berlin / Heidelberg, 2007. doi: 10.1007/978-3-540-75160-1_9.
- [13] B. Kusý, G. Balogh, J. Sallai, Á. Lédeczi, and M. Maróti. InTrack: High precision tracking of mobile sensor nodes. In K. Langendoen and T. Voigt, editors, *EWSN 2007: Proceedings of the 4th European Conference on Wireless Sensor Networks, Delft, The Netherlands, January 29–31, 2007*, volume 4373 of *Lecture Notes in Computer Science*, pages 51–66. Springer, 2007. ISBN 3-540-69829-9. doi: 10.1007/978-3-540-69830-2_4.
- [14] B. Kusý, Á. Lédeczi, and X. Koutsoukos. Tracking mobile nodes using RF doppler shifts. In *SenSys '07: Proceedings of the 5th international conference on Embedded networked sensor systems*. ACM, November 2007.
- [15] B. Kusý, J. Sallai, G. Balogh, Á. Lédeczi, V. Protopopescu, J. Tolliver, F. DeNap, and M. Parang. Radio interferometric tracking of mobile wireless nodes. In *MobiSys 2007: Proceedings of the 5th international conference on Mobile Systems, Applications And Services*, pages 139–151, New York, NY, USA, 2007. ACM Press. ISBN 978-1-59593-614-1. doi: 10.1145/1247660.1247678.
- [16] C. Ladha, B. Sharif, and C. Tsimenidis. Mitigating propagation errors for indoor positioning in wireless sensor networks. *MASS 2007: IEEE International Conference on Mobile Adhoc and Sensor Systems, 2007*, pages 1–6, October 2007. doi: 10.1109/MOBHOC.2007.4428680.
- [17] K. Langendoen and N. Reijers. Distributed localization in wireless sensor networks: a quantitative comparison. *Computer Networks*, 43(4):499–518, 2003. ISSN 1389-1286. doi: 10.1016/S1389-1286(03)00356-6.
- [18] L. Lazos and R. Poovendran. SeRLoc: secure range-independent localization for wireless sensor networks. In M. Jakobsson and A. Perrig, editors, *WiSe 2004: Proceedings of the 3rd ACM workshop on Wireless Security*, pages 21–30, New York, NY, USA, October 2004. ACM. ISBN 1-58113-925-X. doi: 10.1145/1023646.1023650.

- [19] B. Li, Y. Wang, H. Lee, A. Dempster, and C. Rizos. Method for yielding a database of location fingerprints in WLAN. *IEEE Proceedings - Communications*, 152(5):580–586, October 2005. ISSN 1350-2425. doi: 10.1049/ip-com:20050078.
- [20] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson. Wireless sensor networks for habitat monitoring. In C. S. Raghavendra and K. M. Sivalingam, editors, *WSNA 2002: Proceedings of the 1st ACM international workshop on Wireless Sensor Networks and Applications*, pages 88–97, New York, NY, USA, 2002. ACM Press. ISBN 1-58113-589-0. doi: 10.1145/570738.570751.
- [21] M. Maróti, P. Völgyesi, S. Dóra, B. Kusý, A. Nádas, Á. Lédeczi, G. Balogh, and K. Molnár. Radio interferometric geolocation. In *SenSys '05: Proceedings of the 3rd international conference on Embedded networked sensor systems*, pages 1–12, New York, NY, USA, 2005. ACM. ISBN 1-59593-054-X. doi: 10.1145/1098918.1098920.
- [22] R. Nagpal, H. Shrobe, and J. Bachrach. Organizing a global coordinate system from local information on an ad hoc sensor network. In *IPSN 2003: Proceedings of Information Processing in Sensor Networks*, volume 2634 of *Lecture Notes in Computer Science*, pages 333–348. Springer Berlin / Heidelberg, 2003. ISBN 978-3-540-02111-7. doi: 10.1007/3-540-36978-3.
- [23] D. Niculescu and B. Nath. DV based positioning in ad hoc networks. *Telecommunication Systems*, 22(1-4):267–280, January 2003. ISSN 1018-4864 (Print) 1572-9451 (Online). doi: 10.1023/A:1023403323460.
- [24] N. B. Priyantha. *The Cricket Indoor Location System*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, June 2005.
- [25] J. Sallai, P. Völgyesi, and Á. Lédeczi. Radio interferometric quasi doppler bearing estimation. In *IPSN '09: Proceedings of the 8th international conference on Information Processing in Sensor Networks*, San Francisco, CA, April 2009.
- [26] A. Savvides, C.-C. Han, and M. B. Strivastava. Dynamic fine-grained localization in ad hoc networks of sensors. In *MobiCom 2001: Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 166–179, New York, NY, USA, 2001. ACM. ISBN 1-58113-422-3. doi: 10.1145/381677.381693.
- [27] *SiRFstarIII GSC3e/LP & GSC3f/LP – Flagship Performance, Low Power GPS Solution*. SiRF Technology, Inc, 2007.
- [28] A. Smith, H. Balakrishnan, M. Goraczko, and N. Priyantha. Tracking moving devices with the cricket location system. In *MobiSys 2004: Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 190–202, New York, NY, USA, June 2004. ACM. ISBN 1-58113-793-1. doi: 10.1145/990064.990088.
- [29] R. Stoleru, J. A. Stankovic, and S. H. Son. Robust node localization for wireless sensor networks. In C. J. Sreenan, P. Levis, and J. Paradiso, editors, *EmNets 2007: Proceedings of the 4th workshop on Embedded networked sensors*, pages

- 48–52, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-694-3. doi: 10.1145/1278972.1278984.
- [30] *CC2420: 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver*. Texas Instruments, Gaustadalléen 21, N-0349, Oslo, Norway, SWRS041b edition, 2007. URL <http://www.ti.com/lit/gpn/cc2420>.
- [31] *NEO-5 – u-blox 5 GPS modules data sheet*. u-blox AG, Zürcherstrasse 68, 8800 Thalwil, Switzerland, 2009.
- [32] V. Vivekanandan and V. W. Wong. Concentric anchor beacon localization algorithm for wireless sensor networks. *IEEE Transactions on Vehicular Technology*, 56(5):2733–2744, September 2007. ISSN 0018-9545. doi: 10.1109/TVT.2007.899962.
- [33] R. Want, A. Hopper, V. Falão, and J. Gibbons. The active badge location system. *ACM Transactions on Information Systems*, 10(1):91–102, 1992. ISSN 1046-8188. doi: 10.1145/128756.128759.
- [34] A. Ward, A. Jones, and A. Hopper. A new location technique for the active office. *IEEE Personal Communications*, 4(5):42–47, October 1997.
- [35] A. M. Youssef and M. Youssef. A taxonomy of localization schemes for wireless sensor networks. In H. R. Arabnia, V. A. Clincy, and L. T. Yang, editors, *ICWN 2007: Proceedings of the 2007 International Conference on Wireless Networks, June 25-28, 2007, Las Vegas, Nevada, USA*, pages 444–450. CSREA Press, 2007. ISBN 1-60132-039-6.
- [36] L. Yu, N. Wang, and X. Meng. Real-time forest fire detection with wireless sensor networks. In H. Zhou, J. Liu, and R. Conners, editors, *WiCOM 2005: Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing*, volume 2, pages 1214–1217, September 2005. ISBN 0-7803-9335-X. doi: 10.1109/WCNM.2005.1544272.
- [37] P. Zhang, C. M. Sadler, S. A. Lyon, and M. Martonosi. Hardware design experiences in ZebraNet. In J. A. Stankovic, A. Arora, and R. Govindan, editors, *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 227–238, New York, NY, USA, November 2004. ACM. ISBN 1-58113-879-2. doi: 10.1145/1031495.1031522.

