The Always Best Positioned Paradigm for Mobile Indoor Applications

Dissertation
zur Erlangung des Grades
des Doktors der Ingenieurwissenschaften (Dr.-Ing.)
der Naturwissenschaftlich-Technischen Fakultäten der Universität des Saarlandes

vorgelegt von
Tim Schwartz

Saarbrücken
4. Januar 2012

Dekan:
Prof. Dr. Holger Hermanns
Vorsitzender des Prüfungsausschusses:
Prof. Dr. Philipp Slusallek
Berichterstatter:
Prof. Dr. Dr. h.c. mult. Wolfgang Wahlster
Prof. Dr. Antonio Krüger
Promovierter akademischer Mitarbeiter der Fakultät:
Dr. Jörg Baus
Tag des Kolloquiums:
31. Januar 2012

Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet.

Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form in einem Verfahren zur Erlangung eines akademischen Grades vorgelegt.

Tim Schwartz, Saarbrücken, 4. Januar, 2012

Short Abstract

In this dissertation, methods for personal positioning in outdoor and indoor environments are investigated. The Always Best Positioned paradigm, which has the goal of providing a preferably consistent self-positioning, will be defined. Furthermore, the localization toolkit LOCATO will be presented, which allows to easily realize positioning systems that follow the paradigm. New algorithms were developed, which particularly address the robustness of positioning systems with respect to the Always Best Positioned paradigm. With the help of this toolkit, three example positioning-systems were implemented, each designed for different applications and requirements: a low-cost system, which can be used in conjunction with user-adaptive public displays, a so-called opportunistic system, which enables positioning with room-level accuracy in any building that provides a WiFi infrastructure, and a high-accuracy system for instrumented environments, which works with active RFID tags and infrared beacons. Furthermore, a new and unique evaluation-method for positioning systems is presented, which uses step-accurate natural walking-traces as ground truth. Finally, six location based services will be presented, which were realized either with the tools provided by LOCATO or with one of the example positioning-systems.

Kurzzusammenfassung

In dieser Doktorarbeit werden Methoden zur Personenpositionierung im Innen- und Außenbereich von Gebäuden untersucht. Es wird das „Always Best Positioned" Paradigma definiert, welches eine möglichst lückenlose Selbstpositionierung zum Ziel hat. Weiterhin wird die Lokalisierungsplattform LOCATO vorgestellt, welche eine einfache Umsetzung von Positionierungssystemen ermöglicht. Hierzu wurden neue Algorithmen entwickelt, welche gezielt die Robustheit von Positionierungssystemen unter Berücksichtigung des „Always Best Positioned" Paradigmas angehen. Mit Hilfe dieser Plattform wurden drei Beispiel-Positionierungssysteme entwickelt, welche unterschiedliche Einsatzgebiete berücksichtigen: Ein kostengünstiges System, das im Zusammenhang mit benutzeradaptiven öffentlichen Bildschirmen benutzt werden kann; ein sogenanntes opportunistisches Positionierungssystem, welches eine raumgenaue Positionierung in allen Gebäuden mit WLAN-Infrastruktur ermöglicht, sowie ein metergenaues Positionierungssystem, welches mit Hilfe einer Instrumentierung aus aktiven RFID-Tags und Infrarot-Baken arbeitet. Weiterhin wird erstmalig eine Positionierungsevaluation vorgestellt, welche schrittgenaue, natürliche Bewegungspfade als Referenzsystem einsetzt. Im Abschluss werden 6 lokationsbasierte Dienste vorgestellt, welche entweder mit Hilfe von LOCATO oder mit Hilfe einer der drei Beispiel-Positionierungssysteme entwickelt wurden.

Acknowledgments

First and foremost, I want to express my gratitude to Prof. Wahlster for giving me the opportunity and support to work on my PhD thesis. I truly enjoyed the times in which I could discuss my work and worries with him, and I am grateful for the encouragement, the good ideas and sound advice he has given to me. Although I learned a lot in the whole course of preparing and writing my PhD thesis, I feel that I learned the most when closely working with him.

I also want to thank Prof. Slusallek for encouraging me to start a PhD thesis, as well as Prof. Krüger and Prof. Butz who sparked my interest in Artificial Intelligence and who are great sources of inspiration. Likewise, I want to thank Christian Müller who always had an open ear for my concerns. This work would not have been possible without a great working environment with supportive, open minded and creative colleagues. It seems to me that the Chair of Prof. Wahlster and the IUI group at DFKI act as magnets for these special kind of people, who are usually hard to find: Doris Borchers, the good soul of the Chair, Jörg Baus, the 'Swiss Army knife' and stabilizing element of the Chair who provided me with sound advice and encouragement, Christoph Stahl, with whom I worked closely to realize LocaTo and with whom I founded a start-up company, Boris Brandherm, with whom I developed and discussed the concept of geoDBNs, Mira Spassova and Gerrit Kahl, who incorporated LocaTo into the IRL, Dominik Heckmann, who provided UbisWorld and UbisEarth, Ralf Jung, my long-term office mate who realized an location-adaptive audio notification service using Loriot, Michael Schmitz, with whom I had the pleasure to supervise an interdisciplinary course at the school of fine arts, Alexander Kröner, Jens Haupert and Matthieu Deru with whom I supervised two Android seminars, Angela Mahr, Sandro Castronovo, Christoph Endres and Michael Feld, with whom I worked in the project sim^{TD}. Thank you for the great time!

Of course I also want to thank my family for the love and support during my whole life and especially during the last months of this thesis, in which I was terribly absent-minded.

Although my MacBook seemed to overheat a few times in the last couple of months, it is still the most reliable computing machine I ever had. So yeah, thank you Steve Jobs:

Here's to the crazy ones ...

Wo laufen sie denn? Wo laufen sie denn hin, mein Gott?

Bernhard Victor (Vicco) Christoph Carl von Bülow alias "'Loriot"' - AUF DER RENNBAHN

Contents

List of Figures

List of Tables

Part I

Introduction

1 Introduction and Motivation

Prologue

‘Oh yes’, said Fred, smirking. ‘This little beauty’s taught us more than all the teachers in this school.’
He took out his wand, touched the parchment lightly and said, ‘I solemnly swear that I am up to no good.’
And at once, thin ink lines began to spread like a spider’s web from the point that George’s wand had touched. They joined each other, they criss-crossed, they fanned into every corner of the parchment; then words began to blossom across the top, great, curly green words, that proclaimed:

Messrs Moony, Wormtail, Padfoot and Prongs
Purveyors of Aids to Magical Mischief-Makers
are proud to present
THE MARAUDER’S MAP

It was a map showing every detail of the Hogwarts castle and ground. But the truly remarkable thing was the tiny ink dots moving around it, each labelled with a name in minuscule writing. Astounded, Harry bent over it. A labelled dot in the top left corner showed that Professor Dumbledore was pacing his study; the caretaker’s cat, Mrs Norris, was prowling the second floor, and Peeves the poltergeist was currently bouncing around the trophy room. And as Harry’s eyes travelled up and down the familiar corridors, he noticed something else.

This map showed a set of passages he had never entered. And many of them seemed to lead –

‘Right into Hogsmeade,’ said Fred, tracing one of them with his finger.

As Harry stood there, flooded with excitement, something he had once heard Mr Weasley say came floating out of his memory.

Never trust anything that can think for itself, if you can’t see where it keeps its brain.

This map was one of those dangerous magical objects Mr Weasley had been warning against . . . *Aids to Magical Mischief-Makers* . . . but then, Harry reasoned, he only wanted to use it to get into Hogsmeade, it wasn’t as though he wanted to steal anything or attack anyone . . . and Fred an George had been using it for years without anything horrible happening . . .

Harry traced the secret passage to Honeydukes with his finger.

Then, quite suddenly, as though following orders, he rolled up the map, stuffed it inside his robes, and hurried to the door of the classroom. He opened it a couple of inches. There was no one outside. Very carefully, he edged out of the room and slipped behind the statue of the one-eyed witch.

What did he have to do? He pulled out the map again and saw, to his astonishment, that a new ink figure had appeared upon it, labelled ‘Harry Potter’. The figure was standing exactly where the real Harry was standing, about halfway down the third-floor corridor. Harry watched carefully. His little ink self appeared to be tapping the witch with his minute wand. Harry quickly took out his real wand and tapped the statue. Nothing happened. He looked back at the map. The tiniest speech bubble had appeared next his figure. The word inside said ‘*Dissendium*’.

‘*Dissendium!*’ Harry whispered, tapping the stone witch again.

At once, the statue’s hump opened wide enough to admit a fairly thin person.

Taken and compiled from: HARRY POTTER AND THE PRISONER OF ASKABAN by *Joanne K. Rowling* © J.K. Rowling 1999

1.1 Introduction

Many of the magical artifacts that Joanne K. Rowling describes in her world-famous Harry Potter novels can be achieved in the real world through the use of modern information technology. In general, movies or novels can be great inspirations for IT applications. As creative writers of screenplays or novels do not have to care much about the technical feasibility of their ideas, they are free to envision any features that are useful for their current plot. In [Schmitz et al., 2008] and [Endres et al., 2010], movies were analyzed to find new human-computer interaction paradigms and ways to personalize user-interfaces in automotive applications. The Marauder's Map from the excerpt above was, for example, the inspiration for a system called Marauder's Light ([Löchtefeld et al., 2009]).

The main topic of this thesis is positioning, and Harry Potter's 'Marauder's Map' from the excerpt above is an example for an application, which is based on positioning. In the novel, the Marauder's Map is a piece of parchment, which shows a map of Harry's school as well as a part of the neighborhood. Furthermore, the positions of all people wandering about in the school are shown, including the position of Harry himself. With the map, Harry is able to find a secret passage to a nearby village and in order to help him gain access to the passage the map assists him by showing the needed actions.

1.1.1 Context-Aware Applications

In general, such an application that uses the current context of a user, is called a *Context-Sensitive* application. [Schilit et al., 1994] identify three important aspects of context as: 'where you are, who you are with and what resources are nearby'. All three aspects are considered in the Marauder's Map, where the first two are covered by showing the positions of Harry and others, and the last one by showing one or more secret passages. It can also be seen that position or location is a very important aspect of context. A more general definition of context is given by [Dey, 2001]:

Definition 1.1 (Context) *'Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.'*

Services that specifically use the position of a user are called *Location-Based Services*, or LBS for short. A definition for LBS is given in [Virrantaus et al., 2001]:

Definition 1.2 (Location-Based Service) *'LBSs are services accessible with mobile devices through the mobile network and utilizing the ability to make use of the location of the terminals.'*

Location-based services are thus a subset of context-aware services or applications. When seeing the Marauder's map as a mobile device and abstracting away the mobile network as being some sort of magic, it fulfills the definition of an LBS.

A further interesting concept can be identified in the excerpt above: the map determines that Harry is not able to open the entrance to the secret passage and automatically presents useful information to assist him. Such behavior, i.e. giving further information without the user specifically asking for it, is called *proactive*. The opposite behavior, in which the user specifically requests information, is called *reactive*.

Prominent examples in the real world that come close to the Marauder's Map are navigation systems for cars. The route planning of such a navigation system is usually reactive, i.e. a user has to at least specify their destination. The navigation itself, i.e. giving navigation instructions and potential recalculation of the route because of deviations, is proactive. Based on map material and the current position of the vehicle, the system can derive when a user has to take a turn and can give them further assistance. If the current position veers away from the planned route, the system can infer that the user has possibly lost their way or is following a deviation unknown to the system and can thus calculate a new route from the current position to the destination.

Harry's magic map could use a similar approach to infer that he is having trouble opening the entrance to the secret passage: if the map knows Harry's destination – the next village in the excerpt – and detects that Harry stays in front of the entrance for a while, it could deduce that he needs further assistance. This example shows that position information is viable not only for navigation instructions, but also to infer further knowledge about the current context of a user.

1.1.2 Outdoor Positioning

Is the Marauder's Map thus already realizable with little technical effort? No. Today's navigation systems use satellite-based positioning systems, for example GPS (Global Positioning System). As will be seen in Section 2.3.1, the needed reception of satellite signals can already be disturbed by tall buildings, which form so-called urban canyons. Inside of a building, even highly sensitive satellite-receivers can only work satisfactorily near windows and exterior walls. Figure 1.1 shows the reported positions of a GPS receiver, resting on a windowsill in the first level of building E11 on campus of Saarland University. The shown measurements were obtained over

Figure 1.1: Positions determined over a course of 25 minutes by a GPS receiver resting on a windowsill inside building E11 on campus of Saarland University.

a course of 25 minutes after the first position was derived by the GPS receiver (a Holux[1] GPSlim 236). The maximum deviation from the real position in this example is 444 meters. A position determination with GPS deep inside a building – or in a castle as in the case of Harry Potter – is thus highly unrealistic. GPS, or more generally, Global Navigation Satellite Systems (GNSS), are thus so-called outdoor positioning systems.

1.1.3 Indoor Positioning

In order to realize positioning inside of a building, different approaches have been realized, which differ in several aspects: the technology that is used, the positioning accuracy that they are able to deliver, as well as the deployment and maintenance costs. These systems are called *indoor positioning* systems and a de facto standard – like GPS for outdoor positioning – is not available. This is mainly due to the investments a building owner or operator would have to make in order to realize such a system. Although it is possible to derive positions by using already existing infrastructure, these systems may not provide the needed position accuracy for a specific application.

[1] http://www.holux.com/

1.1.4 Privacy Protection

Personal position information is of course highly sensible. On the Marauder's Map Harry can not only see his own position, but also the positions of everybody else in his vicinity. Moreover, these people are not aware that their positions are revealed to somebody else. This is obviously a violation of privacy. Maybe this is why the map is secured with the passphrase 'I solemnly swear that I am up to no good.'. The name Marauder's Map, the term 'Magical Mischief-Makers' and the content of the passphrase itself are of course already hints that the map is intended for misuse. Mr. Weasley's remark to 'never trust anything that can think for itself, if you can't see where it keeps its brain' can also be seen as an appeal for the responsible use of intelligent systems.

The protection of privacy is thus an important issue when dealing with positioning systems. Ideally, a positioning system should be designed in such a way that as little information as possible is revealed to the outside world and thus making it hard for an interceptor to gain access to positioning information. Furthermore, the sharing of positioning information should be under full control of the user.

1.1.5 Design Criteria for Positioning Systems

With the considerations from above in mind, several criteria can be specified, which should be kept in mind when designing a positioning system.

- **Accuracy:** The accuracy of a positioning system describes how close the derived position is to the real position. In general, the needed accuracy of a positioning system depends on the application, which makes use of the position information. Ideally, a positioning system can be used for any application and thus its accuracy should be as high as possible.

- **Robustness:** In principle, a positioning system should always deliver position information. In practice however, this may not always be possible because the position determination depends on the available information, e.g. the number of receivable satellites or the signal quality. It may also depend on the reliability of hardware components, for example the accuracy of the clocks used in satellites, the sensibility of the used receiver, or even the mechanical or environmental resilience of components. A robust positioning system should thus be able to cope with technical failure as well as degraded signals.

- **Cost of Ownership:** The cost of ownership of a positioning system can be divided into the costs for the operator of such a system and the costs for a user, who wants to make use of the system. In both cases, the cost should be as low as possible.

 - **Infrastructure Cost:** If a positioning system needs a dedicated infrastructure, an operator has to compensate for the cost of installing and maintaining the needed infrastructure. This includes the cost of the required hardware as well as expenses for the power consumption and for potential replacements.
 - **Cost of Mobile Device:** A user of a positioning system may need additional hardware to make use of the system, e.g. a GPS receiver and additional computational hardware to visualize the current position or to give route instructions.

- **Usability and Applicability:** A positioning system should also be easy to use and should be applicable in any situation. For example, carrying a satellite dish with a diameter of several centimeters may increase signal quality but is hardly practical during an exploratory tour through a foreign city. Similar considerations come into play when deploying an infrastructure for a positioning system: voluminous hardware may be profitable for the accuracy or robustness of a positioning system, but may as well be impracticable due to space constraints or unsightliness (see also [Schwartz and Jung, 2006]). Several sub-criteria can be specified for usability and applicability:

 - **Weight:** At least on the user side, the needed hardware should be lightweight
 - **Size:** The needed hardware for the infrastructure and the user should be small in size
 - **Power Consumption:** Since the hardware for the user is in general mobile and thus runs on batteries, the power consumption should be as low as possible. In order to achieve low power consumption, the hardware itself and the computational complexity of the used algorithms on the mobile device have to be taken into account.

- **Privacy Protection:** As mentioned in Section 1.1.4, the privacy of a user of a positioning system should be protected as well as possible.

Some of these criteria are conflicting with each other, e.g. higher accuracy and higher robustness can often be achieved through a more expensive infrastructure or more expensive user hardware. In practice, trade-offs have to be made, which often depend on the main application or planned application of a positioning system.

1.1.6 The Always Best Positioned Paradigm

In general, the above mentioned trade-offs are made by the operator of a positioning system, who may either also provide the needed user hardware or may rely on already existent hardware on the user side. If a user wants to be able to find out about their position in as many situations as possible, a solution has to be found, which enables a preferably broad coverage of positioning in indoor as well as outdoor environments. A similar situation is on hand in the area of mobile internet and cell-phone connectivity. A large variety of technologies for data or speech connections is available to a mobile user, e.g. GSM (a second generation (2G) cell-phone technology), GPRS (sometimes dubbed 2.5G), UMTS (3G) and LTE (3.9G). At home and in public places and buildings like hotels, airports or libraries, so-called WiFi hotspots are often available, which allow users to obtain a wireless connection to the internet.

In [Gustafsson and Jonsson, 2003], the concept of *Always Best Connected* (ABC) was introduced: 'The Always Best Connected (ABC) concept allows a person connectivity to applications using the devices and access technologies that best suit his or her needs, thereby combining the features of access technologies such as DSL, Bluetooth, and WLAN with cellular systems to provide an enhanced user experience for 2.5G, 3G, and beyond.'. [Passas et al., 2006] add that the term *Best* in Always Best Connected 'is usually defined separately for each user, as part of his/her profile, and it can be a function of service quality, cost, terminal capabilities, personal preferences etc. [...] This should be performed with no or minimum intervention of the user, leading to what is referred to as ≪invisible network≫'. In other words, the switch between different connectivity technologies should happen proactively, if possible.

In analogy to the ABC concept, the *Always Best Positioned* paradigm (ABP paradigm) can be defined. Similarly to ABC, an ABP system always tries to determine the position of a user, using any means that are currently available to the system. If an ABP system has access to several positioning technologies at the current location, it will try to combine these technologies to achieve an even higher accuracy. Which technologies are used in a given situation generally depends on two factors:

- the technical resources directly available to the user, i.e. a mobile device such as a cellphone and the available senders and sensors of that device
- the available technical resources in the current environment, i.e. the technical infrastructure, which can also consist of senders and sensors

The potential to gain position information and the possible position-accuracy thus depends on the technical resources of the user's device and the environment. As it is the case with ABC, the switch between different positioning technologies should be proactive. An ABP system can thus be defined as:

Definition 1.3 (Always Best Positioned System) *An Always Best Positioned System tries to determine a position as accurately as possible in any situation and at any time using the resources that are accessible at the current location. The addition and omission of positioning technologies or the switch between positioning technologies should be proactive and seamless.*

The Always Best Positioned Paradigm thus tackles some of the design criteria for positioning systems from the user's side of view. In particular the robustness and accuracy, where the former is influenced by the ability of an ABP system to switch between several technologies and the latter by the ability to combine several technologies.

1.2 Research Questions

The main research question answered in this thesis is:

- **How can positioning systems be built according to the design criteria specified in Section 1.1.5?** As already mentioned, designing a positioning system requires trade-offs between single criteria. In Section 1.1.6, the Always Best Positioned Paradigm was already identified as being essential to address the user's need for accuracy and robustness of a positioning system. Therefore, methods will be investigated in this thesis, which help to realize the Always Best Positioned paradigm. In addition, a comprehensive toolkit will be designed and implemented, which allows to create positioning systems that can be tailored to the specific needs of the operator and its users.

In order to answer this main question, the following subquestions must be answered:

- **What are the basic methods for position determination in natural organisms?** In order to gain a basic understanding of how position determination can be achieved, interdisciplinary insights on neuropsychology and biology will be used to identify the basic needs and methods to obtain self-position awareness.
- **How can natural self-position awareness be replicated through methods of Artificial Intelligence?** A link between the natural methods and technical methods of position awareness and position determination will be established. This link will help to identify basic building blocks of positioning systems.

- **How can technical positioning methods be classified and what are the implications of the classification?** Based on the derived basic building-blocks of positioning systems, possible design-variations will be analyzed.
- **How should a positioning system be designed to protect the privacy of its users?** As the protection of the user's privacy is one of the design criteria of positioning systems, the design variations that give the best privacy protection should be identified. This will be done by analyzing the data flow in possible positioning-system architectures.
- **What are possible methods to build positioning systems following the Always Best Positioned paradigm?** In order to realize the Always Best Positioned paradigm, a preferably general solution to perform sensor fusion has to be found. This question will be answered by analyzing known methods for sensor fusion.
- **How far do state-of-the-art positioning systems comply with the specified design criteria and the Always Best Positioned paradigm?** This question will be answered by conducting a comprehensive analysis of state-of-the-art as well as classical positioning systems.
- **How can positioning systems be evaluated?** Since the accuracy is one of the main design criteria for positioning systems, preferably rigorous evaluation methods should be found, which take interferences into account instead of minimizing them. Using the designed toolkit, several positioning systems will be realized and rigorously tested regarding their accuracy.

1.3 Thesis Outline

In Chapter 2, the basic building blocks of positioning systems are derived by giving an overview on the neuropsychological view on perception. Based on these findings, the basic building blocks for a position-aware artificial agent are identified and possible variations are discussed. Furthermore, the chapter gives an introduction to position representation, the mathematical principles of positioning and an introduction to Bayesian filtering and Bayesian networks.

Chapter 3 first gives an introduction into GPS and Global Navigation Satellite Systems in general. Furthermore, the most widespread sender and sensor technologies will be discussed in detail and example positioning-systems for each technology will be explained. State of the art single-sensor and multi-sensor positioning systems will be analyzed and their compliance with the design criteria and the Always Best Positioned paradigm will be discussed.

Chapter 4 introduces the Localization Toolkit LOCATO. An overview on its capabilities will be given and its components will be explained in detail. Section 4.2 to Section 4.4 describe three positioning systems that were developed using LOCATO. Each system addresses different design criteria. Section 4.2 introduces OUT OF THE BLUE, a simple but cost-effective tracking system, with emphasis on usage in conjunction with public displays. UBISPOT, an opportunistic positioning system designed for Android devices that can provide position information in various environments without additional instrumentation, is described and evaluated in Section 4.3. Section 4.4 introduces LORIOT, a real-time capable positioning system with high accuracy, which was designed for instrumented environments. The evaluation of LORIOT will also be described in detail.

Chapter 5 presents six applications that were realized either with modules of LOCATO or by directly integrating one of the implemented positioning systems.

Chapter 6 summarizes the results of this thesis and gives an outlook on future work.

Part II

Theoretical Part: Foundations

2 BACKGROUND

2.1 The Advent of Position Awareness

This section describes a general concept, which enables a methodical analysis of ways how to accomplish the task of positioning.

2.1.1 A Naturalistic Perspective

Positioning is not a purely technical task: Even animals have ways to keep track of their own position. It can be argued that the problem of positioning, i.e. having some sense about one's own location, arises as soon as an organism gains the ability to propel itself in a controlled fashion.

To elaborate further on this thought, consider a hypothetical low life-form in the Panthalassic Ocean[1], with no means of self-locomotion or other ways to influence its own position. Furthermore, it has no sense of its surroundings, nor of its own position. Whether or not this life form has access to life-supporting and species-preserving resources would totally depend on external factors, e.g. ocean currents. Developing a sense of its own position would therefore not be beneficial for preserving its own life or for the survival of its own species (but it would be of no obvious disadvantage either). In terms of genetic evolution, a mutation of this life-form into a 'self-position aware' being would not give a survival benefit over its 'non self-position aware' congeners.

What if the life-form develops the ability to control its own movement and therefore its own position? In order to gain evolutionary advantage from this new ability, the

[1]The Panthalassic Ocean or Panthalassa (Greek: *all sea)* is the global ocean that surrounded the supercontinent Pangaea (Greek: *entire earth*) about 250 million years ago [van Waterschoot van der Gracht et al., 1928]

life-form would need a way to decide, in which direction it should move, i.e. it needs to be aware of the locations of species- and life-preserving resources (and/or the locations of endangering threats) and at least it has to have some indication if it is getting nearer to, or further away from, these locations. In other words, the life-form needs to be aware of its own position relative to locations of other life-forms or objects that are important for it. This kind of position is called *relative position* and such locations that are of particular interest are called *points of interest* (often abbreviated as *POIs*).

Definition 2.1 (Relative Position) *A point defined with reference to another position, either fixed or moving.*

Definition 2.2 (Point of Interest (POI)) *A specific point location that is interesting or valuable for an entity.*

2.1.1.1 Senses and Stimuli

In order to gain a relative position, the life-form needs to be able to sense the presence of other life-forms or objects. In biological terms, this is achieved with sensory receptors that are able to pick up signals that are emitted or reflected by other entities. In his treatise 'De Anima', Aristotle identified five senses: sight (ophthalmoception), sound (audioception), smell (olfacoception), taste (gustaoception) and touch (tactioception). He also argued that there are no other than these five senses. Modern science however recognizes more senses, like pain (nociception), temperature (thermoception), balance (equilibrioception) and kinesthetic sense (proprioception). Non-human senses also include magnetism (magnetoception), electrical fields (electroception), and polarized light.

There is no consensus about a definition of sense and thus the number of senses varies throughout the literature. Some researchers classify into exteroceptive and interoceptive senses. Exteroceptive senses are senses that react to stimuli that originate outside of the sensing entity. Interoceptive senses react to stimuli from inside the entity and can be further divided into proprioception, which senses relative positions of own body parts as well as their acceleration, and visceroception, which senses stimuli originating from internal organs, e.g. the perception of one's own heartbeat (cf. [Vaitl, 1996]). Obviously, exteroception provides the life-form with the necessary information to sense the presence of other entities. Proprioception, on the other hand, provides information about the configuration and orientation of its own body. However, the strict distinction between exteroception and proprioception is also disputed in literature. [Gibson, 1979] claims that 'all perceptual systems are propriosensitive as well as exterosensitive, for they all provide information in their various ways about the observer's activities' (page 115). He further argues, that the term *egoreception* is

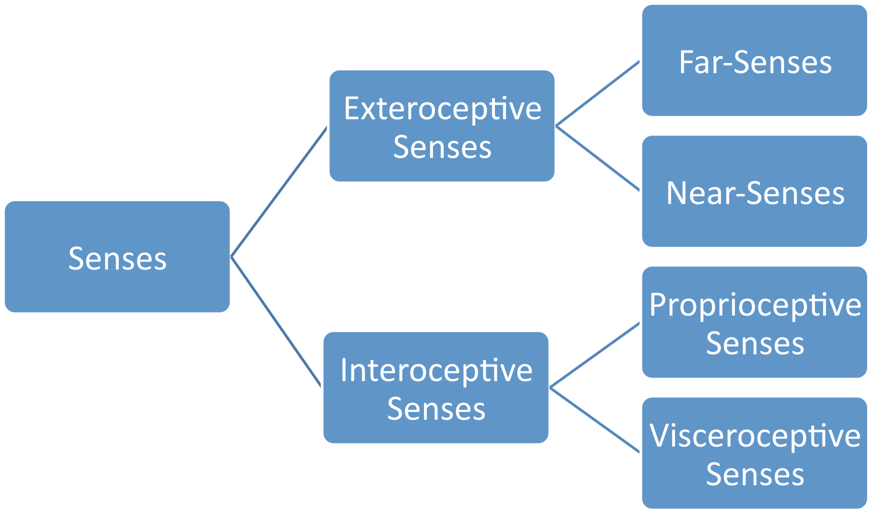


Figure 2.1: A classification of senses.

more appropriate than proprioception. This view is supported by the reported case of a 19 years old male who, due to injury, lost his proprioception. He never recovered from this loss, but managed to relearn how to walk and even how to drive a car by replacing his proprioception through vision. He reported that the mental effort in his daily life feels like having to do a daily marathon (cf. [Kolb and Whishaw, 2003], pp. 173–174 and [Cole, 1995]). From this episode it can be concluded that although proprioception might be substituted by exteroception, the combination of both helps to reduce the cognitive load.

Human senses can further be classified into near-senses and far-senses, depending on whether the perception is directly associated with the sensing organ (near-sense) or not (far sense). According to this definition, the only human far-senses are sight and sound and all other senses are near-senses. Figure 2.1 shows a classification of senses by combining the different existent classifications.

2.1.1.2 Combination of Senses

In general, the combination of different senses, may they be exteroceptive or interoceptive, plays an important role in self-positioning. For example, if one is standing in a bakery, the visual sense will provide cues like the shape and texture of different breads and cakes, olfacoception provides the characteristic smell of freshly baked bread and thermoception may provide a higher temperature due to the heated oven.

The technical term for such a combination of senses is sensor fusion and will be discussed in Section 2.6.

A more concrete example can be derived from fauna. As already mentioned above, some animals have a sense for polarized light, or more specifically for the direction of polarized light. This sense helps them to keep direction, since sunlight gets polarized through scattering and the polarization direction depends on the current position of the sun. The sense for polarization direction thus can help animals to keep their direction, although no direct view to the sun is available. This ability of keeping direction is also called celestial compass ([Wehner and Lanfranconi, 1981]). According to the classification above, the sense for polarization direction is an exteroceptive far-sense. This sense alone is not sufficient to determine a relative position. In [Wagner et al., 2006] the authors hypothesize that ants combine their celestial compass with some kind of step-counter, or odometer, to determine their current position relative to the nest. They tested this hypothesis by artificially shortening or lengthening the legs of Cataglyphis fortis, a foraging desert ant species, after they had arrived at a location outside of their nest. The results confirmed their theory in that ants with lengthened legs overshot while trying to return to the nest and ants with shortened legs stopped prematurely. The exact mechanism of the odometer is not known, but could be based on proprioceptive senses.

Although humans do not have a direct sense for polarized light, most can learn to identify the polarization direction by an entoptic phenomenon[2], called Haidinger's brush. This phenomenon is named after its discoverer Wilhelm Karl Ritter von Haidinger, an Austrian scientist, who realized that a sudden change of the polarization direction relative to the observer's eye results in a visible, faint yellow and blue pattern, whose orientation correlates with the polarization direction [Haidinger, 1844].

2.1.1.3 Dead Reckoning

In the context of animals, the above described method of self-positioning through the use of direction and distance information is commonly referred to as path integration. A more technical term for path integration is dead reckoning, which is claimed to be derived from the phrase 'deduced reckoning' (cf. [Kolb and Whishaw, 2003], page 560). Humans usually perform dead reckoning with the help of tools, like a compass and a means of measuring their speed to determine the traveled distance. Columbus is believed to have used dead reckoning while traveling to Central America. Dead reckoning still plays an important role in technical position determination and will be further discussed in Section 2.6.

[2] a phenomenon that is created in the eye itself

Although there is no evidence that humans use Haidinger's brush to accomplish path integration, it seems that the polarization effect was used by humans in maritime navigation. [Ropars et al., 2011] claim that the Vikings sailed to North America using a special crystal, the transparent common Iceland spar, as a depolarizer to detect the hidden sun and thus being able to keep their direction even when no visible landmarks were available. Ropars et al. conducted experiments showing that with the help of such a 'sunstone', the direction of the sun can be determined up to +/- five degrees, even under crepuscular conditions.

2.1.1.4 Landmarks

A further concept to gain more information about relative position is that of landmarks, i.e. objects with known positions that are easy to identify and that can be perceived over a large distance. Examples for landmarks are peculiar looking mountains or tree formations, but also star-formations and single stars (especially the sun), although the latter change their position over time. Besides natural objects, man made objects can act as landmarks as well and can be classified into landmarks that were built for the purpose of positioning or navigation, e.g. position fires, lighthouses and foghorns, and landmarks that were built for other purposes, e.g. skyscrapers or radio towers (see also Section 2.3.3). [Lynch, 1960] claimed that in order to navigate through cities, the memory of landmarks plays an important role. However, the usage of the term landmark varies in literature. [Sadalla et al., 1980] summarized that the term has been used to denote

(a) discriminable features of a route, which signal navigational decisions

(b) discriminable features of a region, which allow a subject to maintain a general geographical orientation

(c) salient information in a memory task.

In the context of positioning, (a) and (b) are the most appropriate interpretations. A definition close to (b) is given in [Allen et al., 1978]:

Definition 2.3 (Landmark) *Landmarks are environmental features that when recognized with a specific perceptual context, serve as reference points in large-scale space.*

Landmarks that more closely resemble interpretation (a), for example an oddly shaped tree leaning towards one path at a crossroad, are often called waymarks or routemarks, as they are directly related to a route. [Kray, 2003] proposed to distinguish between landmarks and routemarks by taking their proximity and visibility in relation to a position on a specific route into account.

In general, landmarks do not necessarily have to be visually perceptible, they can also be auditive, e.g. the sound of a waterfall, or perceptible by any other sense. The perception, identification and knowledge of the position of one landmark allows to draw conclusions in which area the current position is located. This area can further be diminished by estimating the distance to the landmark. If a landmark is perceived by a near-sense, it can be derived that it is fairly close. For far-sensed landmarks, a distance estimation is often possible because stimuli tend to degrade with increasing distance, e.g. a faraway waterfall sounds softer than a nearer one. The distance to visually perceived landmarks can be estimated by the perceived size or the perceived level of detail. The famous proverb 'Don't shoot until you see the whites of their eyes' is an example for a distance estimation using the perceived level of detail. If more landmarks and distance estimations are available, an even more accurate position determination is possible through so-called trilateration. These methods for position estimation will be discussed in detail in Section 2.5.1.

2.1.1.5 Non-Electronic Tools for Positioning

Humans began very early to develop various aids to enhance their capability of positioning and navigation. Especially on open sea, the lack of earthbound landmarks forced seamen to search for different solutions. Celestial navigation, i.e. the use of stars as landmarks, was and is typically aided by mechanical tools like a kamal, sextant or octant, which help to more accurately determine direction, angles and distance.

Early examples for artificial landmarks especially designed for maritime positioning were fires, which were lit at the coastlines especially near ports to allow navigation even at night. These fires eventually evolved into lighthouses, which have a higher visibility and are protected against weather influences. However, these visually enhanced landmarks have the disadvantage of being barely perceivable in foggy conditions. Foghorns were thus invented, to at least be able to warn ships of rocks or shoals, but these devices only give coarse information about the direction of the signal.

The topophone (see Figure 2.2a), was a purely acoustical appliance, which should help to determine the direction to a sound-source, e.g. a foghorn. Through turning the body, and thus turning two equidistant resonators attached to a shoulder rest, an

(a) The topophone was an acoustic based tool for direction estimation in maritime applications [Scientific American, 1880]

(b) Acoustic mirror near Kilnsea, UK (source: http://www.geograph.org.uk/photo/315865)

Figure 2.2: Acoustical based positioning tools.

increasing or decreasing level of sound could be perceived by the operator. When the operator was facing the direction of the foghorn, the highest volume was perceived[3] [Scientific American, 1880].

Between the first and second World War, so-called acoustic mirrors were used to detect and localize incoming military airplanes. An example of such a sound-mirror is shown in Figure 2.2b. These concrete monumental buildings should reflect and focus the sound of airplane engines and soldiers standing in front of the mirrors should try to estimate the incoming direction by moving in front of the mirrors [Scarth, 1999].

2.2 Human and Artificial Agents

The considerations about senses and perceptions can be transfered into the field of Artificial Intelligence through the use of the notion *agent* instead of life-form. [Poole et al., 1998, page 1] define the field of Computational Intelligence[4] as 'the study of the design of intelligent agents'. [Russel and Norvig, 1995, page 31] define the term agent in the following way:

[3] If the acoustical setup of tubes matches the wavelength of the sound signal, it can also happen that the signals of both resonators cancel each other out

[4] Poole et. al. prefer to use the term Computational Intelligence over the term Artificial Intelligence, and argue that the latter is a source of confusion

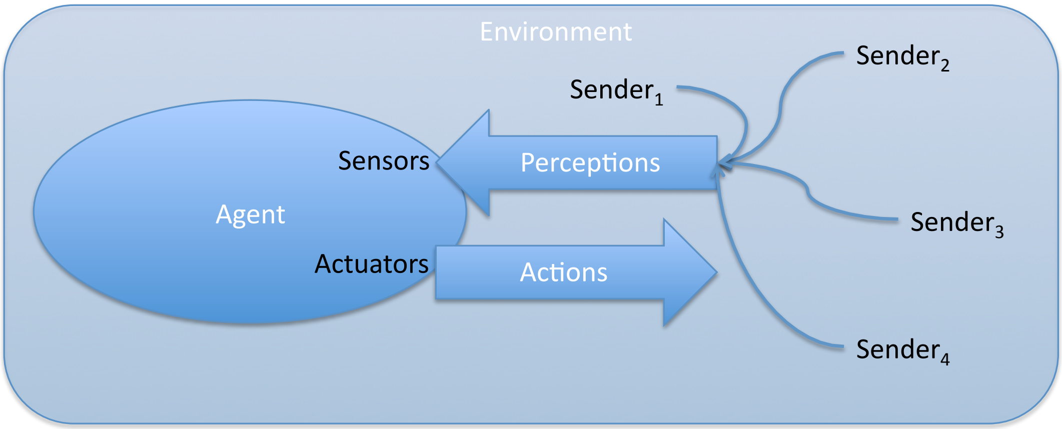


Figure 2.3: An agent has sensors to perceive parts of its environment and actuators to manipulate it.

Definition 2.4 (Agent) *An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors (see Figure 2.3).*

According to that definition, the life-form from Section 2.1.1 can be seen as an agent, with sensors emulating receptors to gain knowledge about landmarks in its vicinity and effectors to change its own position. It is however important to keep in mind that an agent does not necessarily have to have a robot-like appearance. It can also be a pure software-agent, that gets encoded bit-strings as perceptions and produces encoded bit-strings as actions. The idea of agents is becoming even more intriguing, if one considers a human agent carrying a mobile computational-device as a kind of symbiosis[5] between the human agent and the artificial agent: Instead of the artificial agent using effectors to change its position, it uses effectors – e.g. its screen or audio output – to influence the 'host' (the human) to change the position (see Figure 2.4).

This symbiosis can even be seen as mutualistic, when assuming the device's battery charge as the artificial agent's fitness criterion: providing useful location information ensures that the host will keep the device charged. Moreover, human agents can share their perceptions with the artificial agent, which in turn can take these into account to gain position information. Such a situation is depicted in Figure 2.4: the human agent provides its perception of a landmark via speech input. [Kray, 2003] described such a system, called SISTO, which can derive a coarse position through descriptions and tries to refine it by asking additional questions, e.g. showing pictures

[5]The term symbiosis is here used in the original sense of 'the living together of unlike organisms', which includes mutualistic, commensal and parasitic relationships.

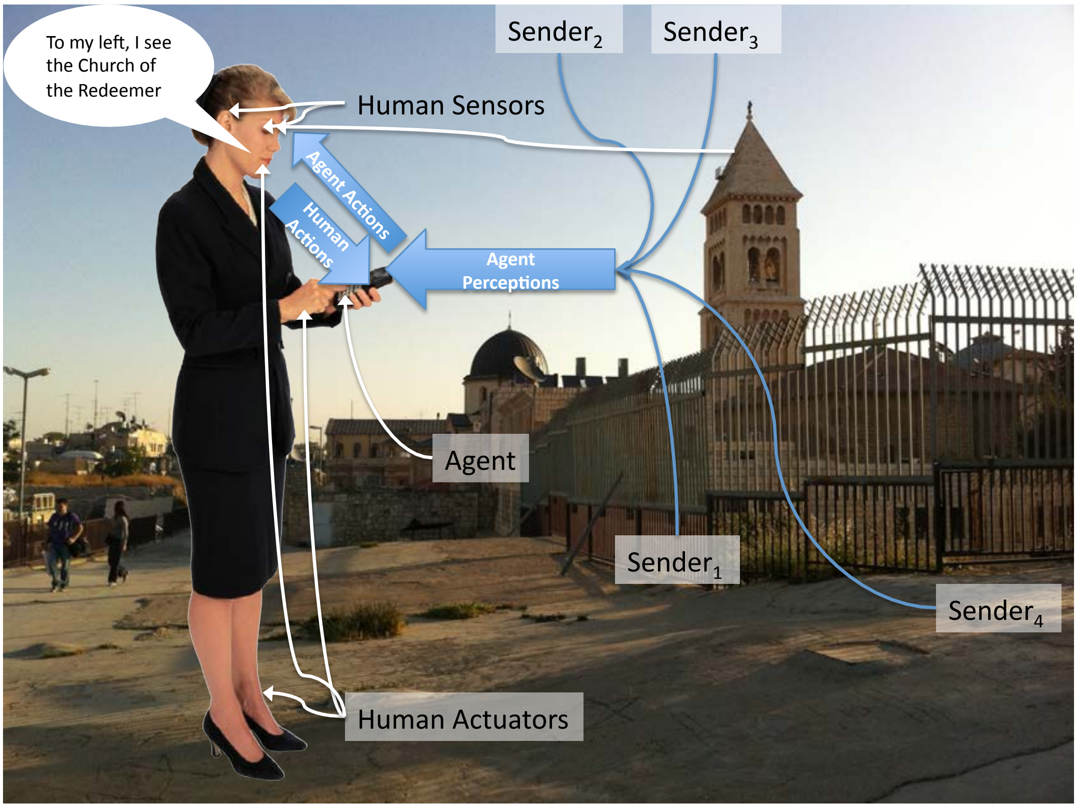


Figure 2.4: A symbiosis between a human agent and an artificial agent: Instead of using its effectors to directly change its environment, the artificial agent uses its effectors as a means of communication with its host to reach the common goal.

of additional landmarks and asking if they are visible. Such a symbiosis of a human agent and an artificial agent is the common scenario for personal positioning, i.e. the determination of one's own position by electronic means. In Figure 2.4, the human agent – or user – carries an agent running on a mobile device, such as a smart phone.

2.2.1 Sensors and Senders

Although the term sensor is used very vaguely in Definition 2.4, it is clear that a sensor acts as an input for some kind of signals.

A technical definition for sensors can be found in the Federal Standard 1037c [National Communications System Technology & Standards Division, 1996]:

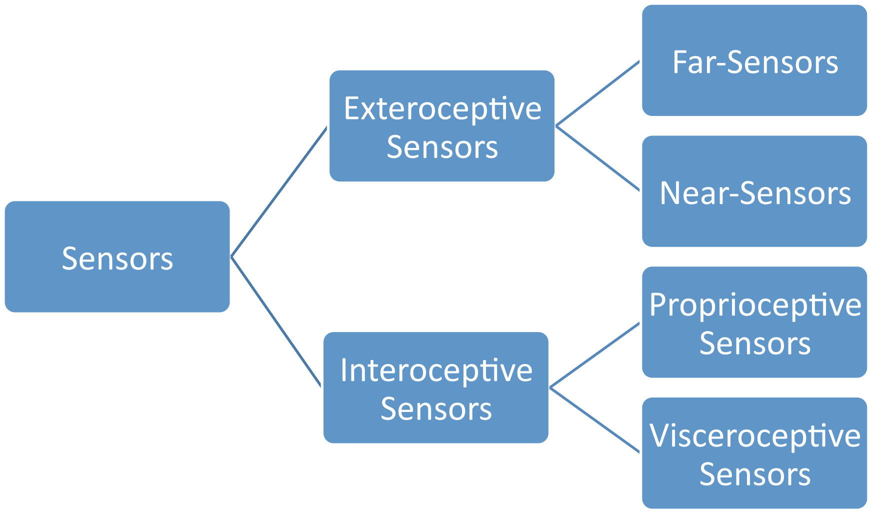


Figure 2.5: Classification of sensors.

Definition 2.5 (Sensor) *A device that responds to a physical stimulus, such as thermal energy, electromagnetic energy, acoustic energy, pressure, magnetism, or motion, by producing a signal, usually electrical.*

According to this definition, the analogy of sensors to sensory receptors becomes obvious. Examples for sensors mimicking human receptors are cameras and microphones, replacing ears and eyes. Although it is possible to use these sensors to pick up natural signals from the environment, the signal processing necessary to derive useful information for positioning can be quite complex and demands high computational power. Most practical attempts for positioning thus use specialized senders, acting as artificial landmarks that broadcast designed signals, which are easier to handle by a machine and can contain data that is tailored to the task of positioning.

Moreover, sensors can be classified in analogy to the classification of senses in Section 2.1.1.1. The sensor classification, as shown in Figure 2.5, is from the perspective of an agent to which the sensors are attached. In that sense, exteroceptive sensors are sensors that pick up signals from external senders. The discrimination between far-sensors and near-sensors depends on the reach of a sensor, i.e. the maximum distance from a sensor to a sender. Examples for far-sensors are cameras and microphones, but also radio technologies like WiFi (see Section 3.1.3). A touchscreen is an example for a near-sensor. Technically, sensors can often reduce their range, which allows them to switch between being far-sensing and near-sensing.

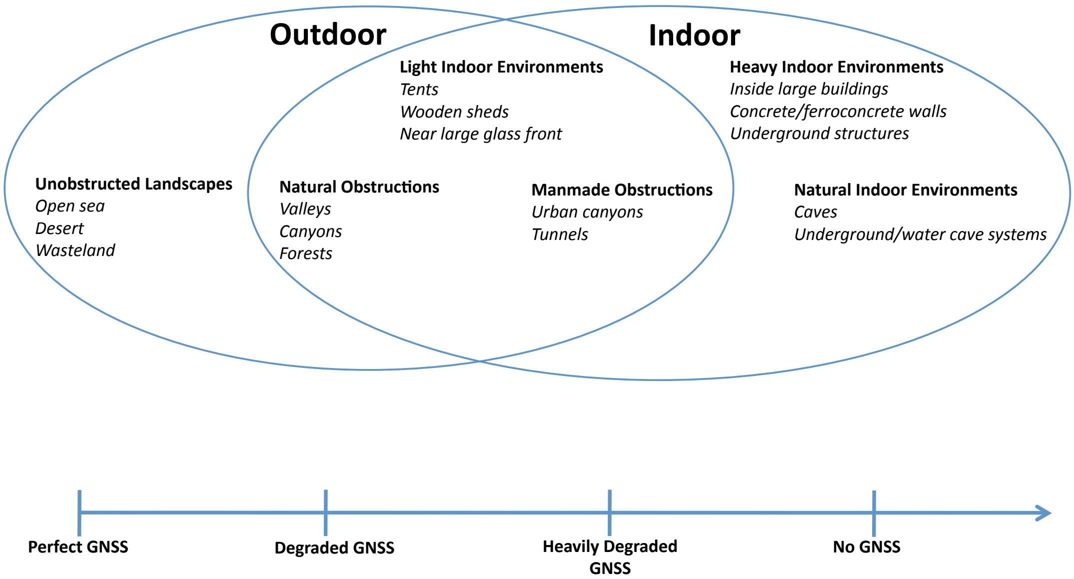


Figure 2.6: Classification of outdoor and indoor situations with examples.

Interoceptive sensors are sensors that react to signals from 'inside' an agent. Proprioceptive sensors are sensors that report about the agent's own spatial configuration, like orientation, acceleration. Many modern smart phones have proprioceptive sensors that replicate human proprioception, for example accelerometers that report acceleration on different axes, and gyroscopes that report the spatial attitude. Visceroceptive sensors report about internal processes or vital internal signals, e.g. the remaining battery-power or the current CPU load.

2.3 Classification of Positioning Systems

2.3.1 Indoor versus Outdoor Positioning

In literature, the terms outdoor and indoor positioning are often found. The need for this distinction arises from the advent of Global Navigation Satellite Systems (GNSS, see also Section 3.1.1), such as GPS. Although these systems are designed to cover the whole globe, a clear line of sight (LOS) to at least four satellites is needed in order to determine a position. The availability of a sufficient number of satellites can be achieved by a high density of satellites, but the prerequisite of a clear line of sight can already be violated in a steep valley or canyon. In general, objects or different transmission media, such as ionosphere and stratosphere, result in so-called fading, i.e. attenuation, scattering, reflection and diffraction of the signals. This

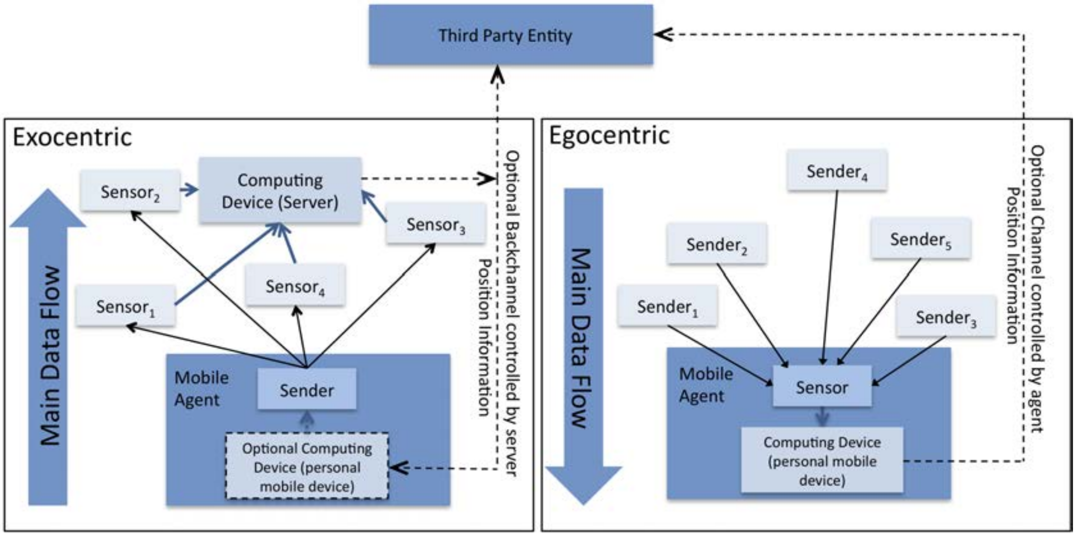


Figure 2.7: Exocentric and egocentric positioning.

fading leads to inaccuracies or even a complete inability to determine a position. This is especially true inside of large buildings, where walls, floors and the roof cause massive degradations of the satellite signals. In general, GNSS does not work inside buildings. Nonetheless, having position information indoors is still desirable, e.g. in shopping malls, airport terminals, museums or large fairs. To overcome these problems of GNSS, special positioning systems are developed, which are called Indoor Positioning Systems (IPS).

Although the terms indoor and outdoor suggest a strict distinction between inside and outside of a building, the term indoor is often used more loosely in literature, describing any system that can be used when GNSS fails, e.g. inside of tunnels, caves, underground or in urban environments where large buildings and structures inhibit satellite signals. In analogy to natural canyons, the latter situation is often called urban canyon. Figure 2.6 shows a classification of indoor and outdoor situations. Perfect GNSS positioning is only achievable with clear LOS. With increasing obstructions, the GNSS signal quality decreases. In light indoor environments, GNSS positioning can be possible with the use of high sensitivity GNSS receivers (see Section 3.1.1.2).

2.3.2 Egocentric and Exocentric Positioning

Through the clear distinction between sensors and senders, positioning techniques can be roughly classified into two categories: *exocentric positioning* (or tracking) and *egocentric positioning* (or self-positioning). Figure 2.7 shows an abstraction of both categories.

2.3.2.1 Exocentric Positioning

In exocentric positioning, the mobile agent is equipped with some kind of sender (or senders) that is broadcasting a specific identification signal. Sensors are installed in the environment and send their readings to a centralized computing device (a server). The sensors do not necessarily have to be connected directly to the server but can also form a sensor network. The centralized computing device collects all sensor readings, which can already be preprocessed, and calculates the resulting position.

With the exocentric approach, the main data flow is from the agent to the environment, meaning that the agent is constantly giving away information. In order to give the agent access to his own positioning information some sort of back channel has to be used. If a third party entity wants to spy on position information, it can gain access to this information by intercepting this back channel or by attacking the server. The exocentric approach is depicted in Figure 2.7 on the left side.

2.3.2.2 Egocentric Positioning

An egocentric positioning system uses the reversed approach: Senders are installed in the environment and the mobile agent is equipped with one or more sensors. The senders broadcast signals into the environment, which are collected by the mobile agent's sensors. The agent can then calculate its own position. The main difference to the exocentric approach is that the main data flow is from the environment to the agent, meaning that it is the environment, or parts of the environment, that is giving information to the agent. The positioning calculation is thus literally in the hands of the user.

In the case of egocentric localization a back channel, which is usually controlled by the agent, can be used to voluntarily give a third person access to the position information. A malicious third party would have to attack the mobile agent directly. The egocentric approach is shown on the right side of Figure 2.7.

2.3.2.3 Hybrid Approaches

Besides the two basic cases, it is also possible to have both, senders and sensors, on the same side, i.e. in the environment or at the agent. If such a combination of sender and sensor share parts of their circuitry, it is called a *transceiver*. If both parts are

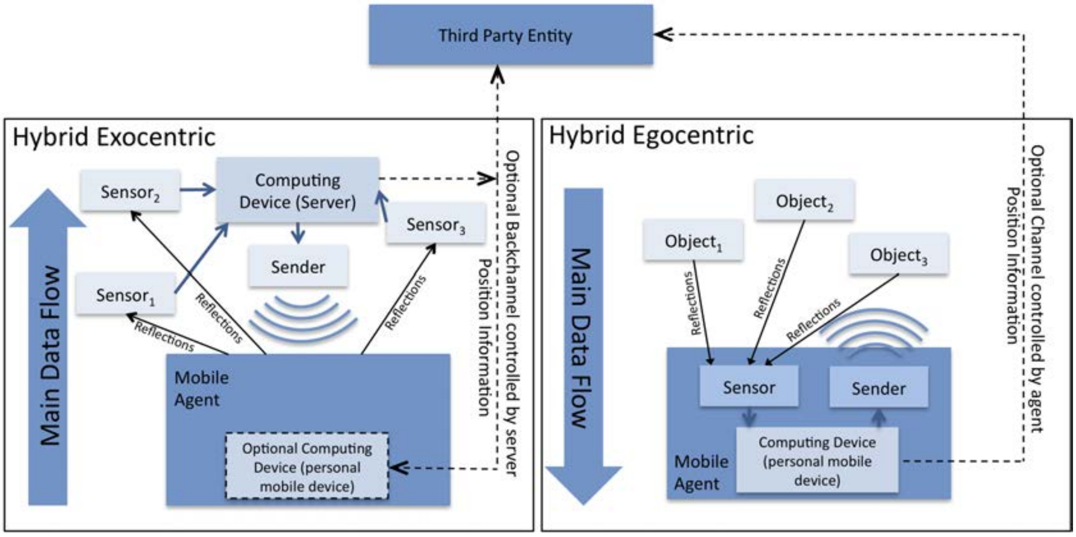


Figure 2.8: Hybrid exocentric and egocentric positioning. Sender and sensor are both installed either in the environment or at the agent.

completely separated, it is called a *transmitter-receiver*. When such a combination is used, the senders usually broadcast a high entropy signal and the receivers pick up the same signal or reflections of the signal. The received signal may have a lower entropy and can thus contain data useful for identification or position determination. A simple example for such a sender/sensor combination is a light barrier, which can only detect if something breaks the light ray.

A more elaborated example is a laser range scanner, which can also measure the distance to the reflecting object. An example for changing the entropy of a signal is a camera with a flash light: the emitted white light is an evenly distributed mixture of frequencies in the visible light range, and has thus a high entropy. Objects absorb some of these frequencies and reflect others and thus reduce the entropy. The result, as most should know from experience, is a picture containing a massive amount of data.

If such a combination is installed in the environment, the system is classified as exocentric, as it is still the agent who will reveal information to the environment, by either reflecting or absorbing the signal.

A system where the agent is equipped with the sender/sensor combination is egocentric, although a third party entity might try to use the broadcast signal to track the agent. However, such a third party tracking system will then again be exocentric. Both variants of hybrid methods are depicted in Figure 2.8.

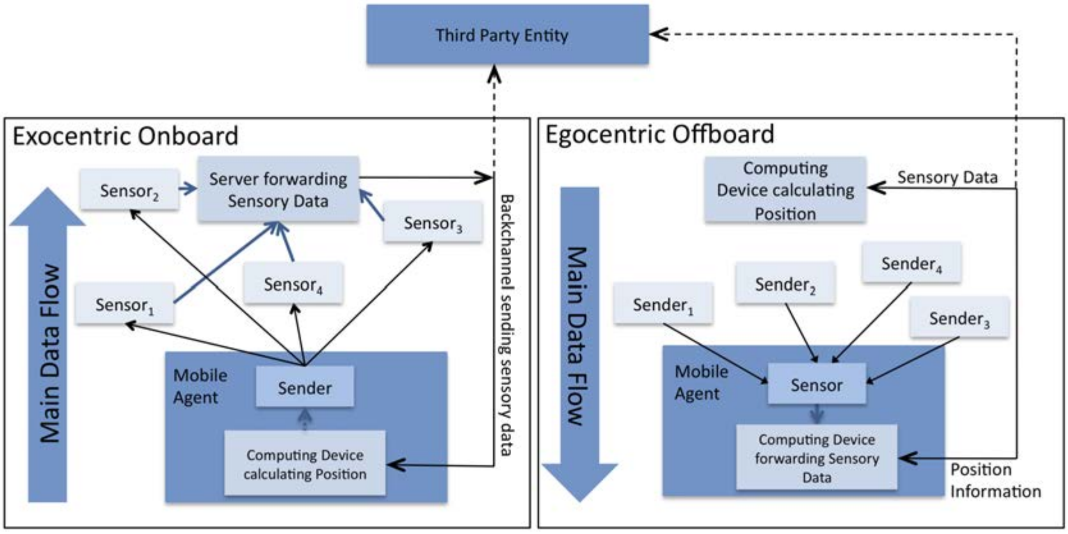


Figure 2.9: Onboard exocentric and offboard egocentric positioning. Sensory data is sent from the environment to the agent or from the agent to a computing device in the environment.

2.3.2.4 Onboard and Offboard Calculation

With an exocentric system, it is also possible that sensors send their collected data back to the mobile agent, which then does the actual position determination. Although this would in a way protect privacy, the user would have to trust the system to not further distribute the collected sensory data.

The same approach is possible with an egocentric system, i.e. the mobile agent collects sensor data and forwards it to a centralized server, for example to reduce its own computational load. In this case, the privacy could be protected through the use of an anonymized and encrypted protocol to the server, but this is again a matter of trust on the user's side.

With these considerations in mind, the terms onboard and offboard can be used to further classify positioning systems. Onboard systems calculate the position on the user's personal device, i.e. the agent determines its own position, while offboard systems use a computation device outside of the user's control. Exocentric onboard and egocentric offboard are visualized in Figure 2.9. In the case of egocentric offboard, two channels or a bidirectional channel have to be used to communicate sensor data to the computing device and to receive position data from the computing device. Exocentric offboard and egocentric onboard are the default cases of exocentric and egocentric as shown in the previous Figure 2.7.

	Exocentric	Egocentric
Offboard	Low Privacy	Reduced Privacy
Onboard	Reduced Privacy	High Privacy

Table 2.1: Privacy levels for different configurations of positioning systems.

2.3.2.5 Discussion

As described above, the main difference between both positioning techniques lies in the direction of the main data flow. From agent to environment in the exocentric case and from environment to agent in the egocentric case. Offboard exocentric positioning is therefore the preferred approach for tracking people or objects, since in that case neither the agent nor the object is interested in the position but a third person wants to find out about the whereabouts of somebody or something else. For the purpose of personal navigation, which requires the knowledge of the own position, an exocentric system has to be extended with a back channel to either send back the derived position or to enable onboard calculation, which can increase the cost of such a system.

Onboard egocentric positioning is the choice for self-localization (e.g. for navigational purposes), since the agent itself is interested in its own position and passing the positioning information on to third persons is often perceived as a violation of privacy. An offboard egocentric positioning system or an onboard egocentric system with a back channel can also be used for tracking purposes. As in the case of an added back channel in exocentric positioning, this can lead to further costs but has the advantage that users can either use the system for their private purposes (by switching off the back channel) or to share their positioning information with other users or systems.

The different privacy levels depending on ego/exocentric and on/offboard are summarized in Table 2.1.

In general, onboard positioning systems have to deal with higher technical-resource limitations of mobile device in comparison to stationary devices, e.g. restricted computational power, restricted memory capacity and restricted power supply. Algorithms for onboard positioning thus have to be optimized according to those restrictions. An onboard system can thus be easily converted into an offboard system, in the simplest case by using the same computational hardware and installing it into the environment. In addition, sensor data has to be communicated to the now stationary device, if the system is egocentric.

On the other hand, converting an offboard system into an onboard system is usually not an easy task (unless one is willing to carry a bulky stationary device with heavy batteries). Depending on the used algorithms, even a high optimization may not be sufficient to cope with the resource restrictions of mobile devices.

2.3.3 Instrumented Environments and Opportunistic Positioning Systems

As already indicated above, positioning is also possible using natural signals, i.e. without special senders. Examples for such natural signals are heat radiation of a human body, which can be picked up by infrared sensors, or the noise produced by the engine of an airplane, which can be picked up by a microphone or by sound mirrors (see Section 2.1.1.5). Systems that use these kinds of signals are called passive positioning systems.

Most positioning systems however need additional infrastructure installed in the environment. For example, the well-known GPS requires specialized satellites acting as senders in the orbit around the earth (see also Section 3.1.1.1). The infrastructure, or instrumentation, for a such a system is thus specially designed to enable positioning. As already indicated in Section 2.1.1.5, humans very early began to build such instrumentations in form of lighthouses, fog-horns and buoys. Early electronic systems for position determination in maritime and aviation applications were the Britsh GEE system and the American Loran (LOng RAnge Navigation) system, which were both developed during World War II ([Appleyard et al., 1988]). The GEE system was the first hyperbolic positioning system and was mainly used for aircrafts. The instrumentation consisted of stations that were installed on the ground and were organized in chains. Each chain had a master station that was responsible for the synchronization of three slave stations, dubbed B, C and D. Pilots could determine their position by tuning in to these stations and using multilateration (see Section 2.5.1.4). Even today similar systems are still in use for aviation purposes as Instrument Landing Systems (ILS). Besides the use of terrestrial radio signals, bright light-arrays are often used in addition.

Besides these especially for positioning designed instrumentations, it is also possible to use instrumentations that were originally set-up for a different purpose. In general, technical services or applications often need a special infrastructure, e.g. mobile phones need cell towers; wireless internet needs access points. If a positioning system uses the already existing infrastructure of a different service, it is called an opportunistic positioning system. WiFi-based positioning systems are an example for such opportunistic approaches (see also Section 2.5.3 and Section 3.1.3).

Positioning systems with designed instrumentation often achieve higher position accuracy, but also increase the cost of deploying and maintaining the system. Opportunistic positioning systems on the other hand, help to keep the infrastructure-costs low because no additional instrumentation has to be provided, but often have the disadvantage of a lower position accuracy and need an initial training phase. However, this initial training phase can sometimes also be combined with other activities, like it was done with Google Streetview[6], where cars used to actually take geo-referenced pictures simultaneously collected geo-referenced WiFi data. This method of collecting geo-referenced WiFi data with a car is called war-driving, as opposed to war-walking, where the same is done by pedestrians.

2.4 Position Representation

In order to realize a positioning system, positions have to be represented by some means. A straightforward mathematical way is to use a Cartesian coordinate system. In Section 2.1.1 the definition for relative positions was given. As a matter of fact, any position is a relative position and a position given in a Cartesian coordinate system is a relative position with reference to the origin of the coordinate system. Such a position is represented by a multidimensional vector, containing the distance on each Cartesian axis to the origin. However, if a certain coordinate system is defined and agreed upon, positions in reference to the origin of that coordinate system are often called absolute positions (in respect to that coordinate system), and relative positions are positions that are given in reference to such an absolute position.

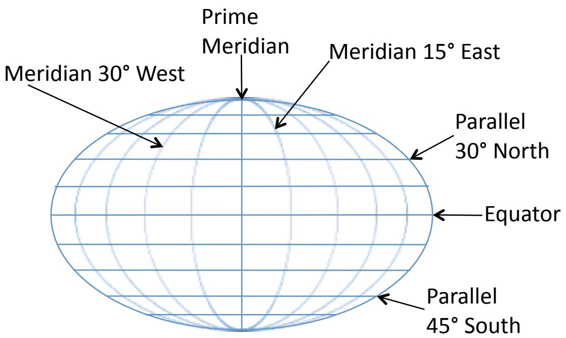
Cartesian coordinate systems are easy to handle and sufficient to describe small areas, such as a building or campus. However, when larger areas have to be covered, such as a country or even a whole planet, the fact that the Earth is not flat can no longer be ignored.

2.4.1 World Geodetic System WGS84

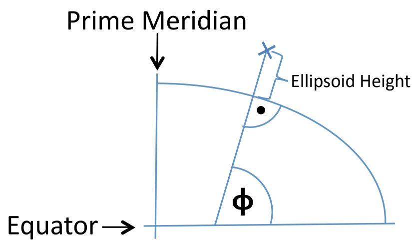
There are several coordinate systems that deal with covering large areas or the whole Earth. The World Geodetic System (WGS) in its iteration WGS84 is widespread, as it is the reference coordinate system of GPS (see also Section 3.1.1.1). WGS84 was established, as the name implies, in 1984.

In order to represent a position on Earth, a geometrical model is needed that describes its shape. In WGS84, the Earth's shape is approximated by a biaxial ellipsoid, with a major radius of $6,378,137$ meters (equator) and a minor radius of $6,356,752.314245$ meters (rotational axis) ([National Imagery and Mapping Agency, 2000]). The center of that ellipsoid lies at the Earth's center of mass. In order to describe a position

[6]http://maps.google.com/streetview/



(a) Meridians and parallels on the ellipsoid



(b) Determination of latitude and ellipsoid height

Figure 2.10: WGS84 coordinates are expressed in latitude, longitude and altitude.

on the surface of the ellipsoid, it is organized into meridians, which run from north to south, and parallels, which run from east to west (see Figure 2.10a). One meridian is chosen as the prime meridian, and the equator of the ellipsoid is chosen as the reference parallel. Coordinates are expressed with two angles: latitude and longitude, where the prime meridian is assigned to 0° longitude and the equator is assigned to 0° latitude. An arbitrary meridian is identified by the angle between this meridian and the prime meridian, where angles go from 0° to 180° East and from 0° to 180° West. The longitude of a position on the ellipsoid is thus determined by finding the meridian that passes through that position.

The latitude of a position is determined by the angle ϕ between the plane that is described by the equator and the line that is perpendicular to the ellipsoid and passes through the position (see Figure 2.10b). Similar to longitude, latitude angles are divided into North and South and are thus expressed as values between 0° and 90° North or South.

A position that lies directly on the ellipsoid can thus be described by latitude and longitude. In order to express the height of a position, an appropriate reference has to be found. If the ellipsoid itself is used as a reference, the third coordinate is named ellipsoid height and denotes the distance from the ellipsoid surface to the position along a line perpendicular to the ellipsoid surface. Although latitude, longitude and ellipsoid height (λ, ϕ, H) uniquely define a position with respect to the ellipsoid, the ellipsoid height does not really correspond with the actual height of a position, i.e. to reach a position with the same ellipsoid height one may have to climb a hill.

A better measure of height can be derived by using a Geoid. The basic idea of a Geoid is best explained by imagining that all of the Earth's oceans were connected through canals and that influences like weather and sea currents were not present. The surface of the water would then describe the perfect Geoid of the Earth and would directly depend on the Earth's gravitational field and the centrifugal force of its rotation. The surface of the Geoid is thus highly irregular, as the gravitational field is not regular. In practice, a Geoid is derived by measurements from a dense net of reference points and modeled through a spherical harmonics representation, which allows to approximate the gravitational potential at a particular position on the ellipsoid. With the help of such a Geoid, the ellipsoid height can be transferred into a Geoid height (sometimes also called orthometric height). Since an accurate Geoid model is rather complex and thus needs additional memory, GPS receivers usually report the ellipsoid height, which can then be transformed by additional hardware. Currently WGS84 uses the 1996 Earth Gravitational Model (EGM96).

If a position expressed in geodetic coordinates latitude, longitude and ellipsoid height is to be converted into a three-dimensional Cartesian system, the origin of the Cartesian coordinate system has to be defined, along with the orientation of the axes in relation to the Earth's surface, and the used ellipsoid has to be taken into account. This set of information is called a geodetic datum or Terrestrial Reference System (TRS). In order to show geodetic coordinates on a map, the coordinates have to be projected onto a two-dimensional Cartesian coordinate system. This is done through a map projection and the resulting coordinates are usually called eastings, for the x-coordinate, and northings, for the y-coordinate (cf. [Ordnance Survey, 2010]).

2.4.2 Semantic Representation

Coordinates are a mathematical expression of a position, but are usually meaningless to humans unless indicated on a map. In day-to-day conversations, semantic descriptions are used to indicate a position, e.g. 'I'm in the kitchen' or 'I'm currently in Babylon'. Of course such descriptions can be underspecified or can only be disambiguated with further knowledge about the context or dialog discourse. For example if somebody has received a phone call on their landline telephone and utters 'I'm in the kitchen' it can be derived from the context that the person is in the kitchen of his home. If somebody mentions on a mobile phone that they have just landed at JFK and that they are now in Jamaica, it can be inferred from the discourse and world knowledge that the person is most probably in Jamaica, in the borough of Queens in New York City.

To represent such a semantic description of a position, a hierarchical spatial ontology can be used, which describes the spatial relations between different locations. The example position above could be expressed as Earth → North America → USA → New York → New York City → Queens → Jamaica. In general, such an ontology can be stored in a tree, where each node represents an area and the areas denoted by child nodes are part of the areas denoted by their parent node.

(a) Top nodes of UBISEARTH

(b) Subnodes can be freely added, even down to single drawers

Figure 2.11: UBISEARTH is a spatial ontology in UBISWORLD.

2.4.2.1 UBISWORLD

Such a spatial ontology is provided by UBISWORLD[7], which was developed by [Heckmann, 2006]. UBISWORLD is a cloud service for ubiquitous user modeling that follows the Web 3.0 paradigm, i.e. the combination of Web 2.0 ideas with methodologies of the Semantic Web, as described by [Wahlster et al., 2006]. The basic idea of ubiquitous user modeling is to build and keep an up-to-date user profile, which is accessible anytime and anywhere and contains viable information that can be used for user adaptation. Of course such information is highly private and thus appropriate filtering and security mechanisms have to be taken into consideration.

The current position of a user or a history of positions is of course viable information for a user profile. UBISWORLD uses ontologies to build its knowledge base and the ontology that is used to represent positions is called UBISEARTH. This ontology contains over 28 million places all over the Earth. Figure 2.11a shows the first layers of the hierarchy of UBISEARTH: Earth, which is divided into Oceans and Continents. The hierarchy is then further divided into Countries, Regions, Cities, Buildings and Rooms. Since the ontology is user editable, further subnodes can be defined, for example cabinets and drawers of cabinets (see Figure 2.11b). Countries, Regions etc. are roles, and Germany, Saarland etc. are instances of these roles.

[7] http://www.ubisworld.org/

Every instance in UBISWORLD has a unique identifier, called UbisPointers. Instances can also have properties, which contain further information. In Figure 2.11b, the properties of the instance Saarbrücken can be seen. Among other information like zip-codes and dialing-codes, a WGS84 coordinate is given. With this information it is possible to convert a semantic description or a UbisPointer into geodetic coordinates and vice versa.

As a cloud-service, UBISWORLD provides sophisticated interfaces, which allow to efficiently search, modify and refine the ontologies while keeping the data traffic low. With these interfaces, it is possible for a mobile agent to download a specific part of the UBISEARTH ontology that is currently relevant to the user (cf. [Heckmann et al., 2005a, Heckmann et al., 2005b, Schwartz et al., 2006, Loskyll et al., 2009]).

2.4.3 Positioning in a Moving Reference System

An interesting situation arises when a user is inside a larger, moving object, for example a train, a plain or a cruise ship. A cruise ship is basically a swimming multistory-building and passengers might be mainly interested in services that are related to their position inside the ship, e.g on which deck they are, where the casino is located or how to find back to their cabin. However, in some circumstances, for example to pinpoint the exact moment when one is crossing the equator for the first time, they might be interested in their exact WGS84 coordinates.

In ships or airplanes, the current position of the vehicle is determined for navigation purposes. Thus, a solution this situation is to use a static coordinate system for the interior of the vehicle itself, with a datum (see Section 2.4.1) that ideally coincides with a reference point for the vehicle's own position determination, e.g. a GPS antenna.

Mobile agents can use this static coordinate system to determine their position within the vehicle. If a global position is needed, the mobile agent needs access to the vehicle's position and can then determine its own global position. If a system like UBISWORLD is used, the vehicle would constantly update its own position in the spatial ontology, and a passenger's position would include the vehicle in the description hierarchy, e.g. MS Ejemplo → Deck 2 → Dining Room → Table 2 → Seat 4. The properties of subtree of the vehicle, which contain the geodetic coordinates of each position, would then be automatically updated by UBISWORLD.

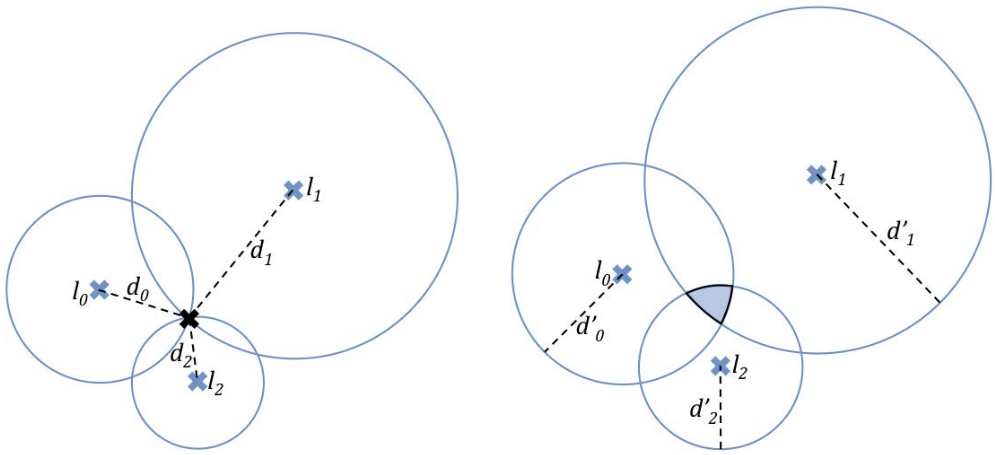
2.5 Basic Mathematical Principles of Positioning

With the considerations from Sections 2.3 and 2.4 in mind, the basic building blocks for a positioning system can be derived. Regardless if a positioning system is egocentric, exocentric, onboard or offboard, position determination is always done on the basis of sensor data. Using the raw sensor output, a positioning system has to derive a position representation. In this section, standard mathematical position determination methods are introduced, which mostly rely on a position representation through numerical coordinates.

2.5.1 Trilateration and Multilateration

Trilateration is a method to determine the position of an object with the use of three distance measurements to three known locations. In two-dimensional space, one such distance measurement d_0 to a known location l_0 leads to the conclusion that the searched position is somewhere on the circle around l_0 with radius d_0. Adding a second distance measurement d_1 to known location l_1 reduces the possible positions to two points, described by the intersection points of the two resulting circles. A third measurement finally disambiguates between the two positions and thus determines the correct position in the plane that is defined by the three known locations (see Figure 2.12a).

Besides the location in the plane, also the height above the plane can be computed, which generally has two solutions representing mirror images with respect to the



(a) Trilateration with accurate measurements (b) Trilateration with inaccurate measurements

Figure 2.12: Trilateration with accurate and inaccurate measurements.

plane. The correct solution cannot be determined mathematically and must therefore be solved by using additional information, e.g. testing if one solution would lead to a position inside the earth ([Fang, 1986]) or by adding a fourth distance measurement.

In practice, measurements are noisy and thus introduce inaccuracies, which lead to an area instead of one point in which the searched position can lie. Figure 2.12b shows an example where the distance measurements d'_0, d'_1 and d'_2 are erroneous. The resulting area of possible solutions is shaded and marked with a black line.

In general, trilateration can be expressed as the problem of finding the solution to a system of quadratic equations, where each equation describes a sphere around a known location with the measured distance to that location as radius ([Thomas and Ros, 2005]):

$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = d_0^2 \tag{2.1}$$
$$(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2 = d_1^2 \tag{2.2}$$
$$(x - x_2)^2 + (y - y_2)^2 + (z - z_2)^2 = d_2^2 \tag{2.3}$$

Here, x, y, z denote the coordinates of the searched position, x_i, y_i, z_i, $i = 1, 2, 3$ are the coordinates for the known locations l_i and d_i are the measured distances.

In [Murphy and Hereman, 1995], the authors tested different approximations for trilateration and found out, that a nonlinear least squares method gave the most accurate position calculation. This method results in the exact position if the exact distances are known and in a reasonably accurate position if only approximate distances are known.

In order to obtain the needed distance measurements, several standard methods can be found throughout the literature, which will be described in the following.

2.5.1.1 Signal Strength

As already indicated in Section 2.1.1, signals or stimuli tend to degrade with increasing traveling-distance. Sensors can often derive an indication for the received signal strength, the so-called Received Signal Strength Indicator (RSSI). Technically this property can be used to estimate the distance to a sending object, when a propagation model for the signal-type is available. Generally, signal loss follows the inverse-square law, which states that the strength of a signal is inversely proportional to the square of the distance it has traveled.

$$p \sim \frac{1}{d^2} \tag{2.4}$$

Using more elaborated path-loss models, the distance to the sender can be approximated. RSSI based distance calculations are however highly influenced by various environmental factors, like air humidity, temperature, refraction and reflection.

2.5.1.2 Time of Arrival (TOA)

Time of Arrival describes the determination of the distance between a sender and a sensor by measuring the travel time t of the signal. When the travel velocity v is known, the distance d can be easily calculated by

$$d = v * t \tag{2.5}$$

The travel time, sometimes also called Time of Flight (TOF), can be measured by incorporating a time-stamp into the signal, which indicates when the sender started to transmit the signal. In this case, the clocks of all receivers and senders need to be tightly synchronized with each other. Another way to accomplish TOA is to send a signal with embedded time-stamp and have the receiver immediately send the same signal back (optionally with added delay information) and then measuring the so called round-trip time of the signal, which results in twice the travel time (minus the added delay).

TOA can be used for exo- and egocentric applications. In the case of exocentric positioning, at least three sensors with known positions are needed, each one determining the distance to the sender and sharing this information with each other or an additional instance that then performs the trilateration.

For egocentric positioning, at least three senders need to be detectable by the sensors of the self-locating entity. Moreover, the location of the senders has to be known, e.g. by sending this information along with the time-stamp or by a map stored on the locating device.

Although simple in theory, the practical application of TOA has some drawbacks: the distance error depends highly on the accuracy of the time measurement. In the case of radio signals, which are traveling at the speed of light, a measurement error of $1\mu s$ results in about 300 meters distance error. Also, the velocity of a signal depends on the materials it travels through. A signal passing through different or unknown materials can therefore also have an effect on the accuracy of the distance calculation.

2.5.1.3 Pseudorange

In the context of satellite based positioning, the term pseudorange or pseudoranging is used to describe the above mentioned problem of inaccuracies in TOA measure-

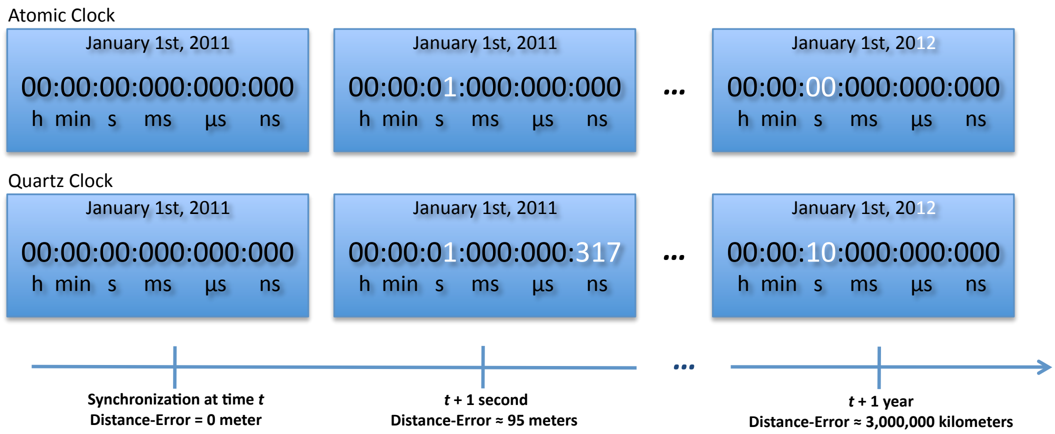


Figure 2.13: Inaccuracies of a quartz timer in comparison to an atomic clock.

ments due to unsynchronized or loosely synchronized clocks. More specifically, the receiver's clock is usually based on a quartz oscillator, whereas the satellites use atomic clocks. In comparison to an atomic clock, a quartz oscillator is either a bit too fast or too slow, which results in a timer-offset between both clocks. For example, a typical quartz watch has an accuracy of about ± 10 seconds per year:

$$\frac{10s}{1a} \approx \frac{10s}{365 * 24 * 60 * 60s} \approx 0.317\mu s/s \tag{2.6}$$

This means that if an atomic clock and a quartz clock are perfectly synchronized at time t, then at time $t + 1s$ the quartz based clock will be $\approx 0.317\mu s$ before or behind the atomic clock. As radio signals travel with the speed of light c, the resulting error in the distance calculation according to Equation 2.5 is $0.317\mu s * c \approx 95m$. The timer-offset will increase over time; after one hour, the error will be over 300 kilometers and after one year 3 million kilometers (see Figure 2.13). The term pseudorange is used to describe an uncorrected distance measurement, i.e. pseudorange $= (signal\_traveltime + timer\_offset) * c$. Since the satellites are tightly synchronized among themselves, the offset between a receiver and any satellite is constant at a given point in time. To correct the resulting distance error, the timer-offset can be computed through a fourth TOA measurement, as described in [Teunissen and Kleusberg, 1996].

2.5.1.4 Time Difference of Arrival (TDOA)

In contrast to TOA, Time Difference of Arrival does not measure the absolute travel time of signals from senders to receivers, but the time difference of either the arrival of a signal on at least two different receivers with known locations (exocentric), or the

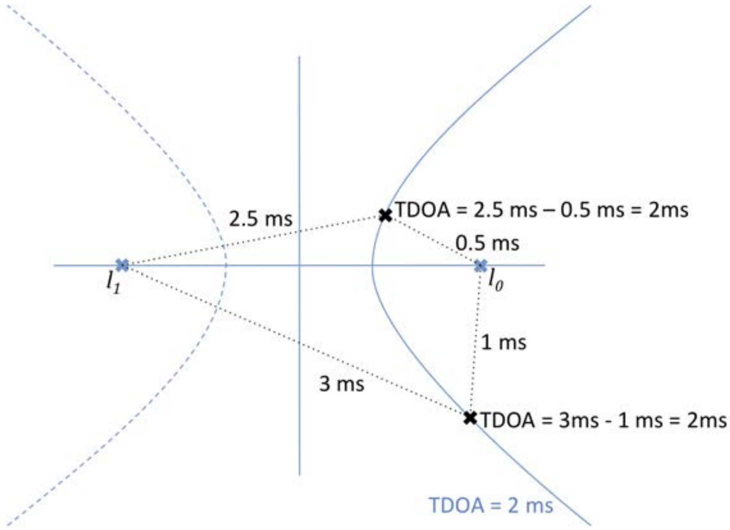


Figure 2.14: A single TDOA measurement results in a hyperbola with two known locations as focus points. Only one branch of the hyperbola has to be considered (marked with a solid line).

arrival of signals sent simultaneously from at least two senders with known locations (egocentric).

In the exocentric case, the needed time difference can be determined by a cross-relation process, which means that the receivers need synchronized clocks and a way to exchange the measured signals. Determining one such time difference leads to a hyperbola with the known locations of the receivers at its two focal points ([Bucher and Misra, 2002]). Unless the sender has the exact same distance to the receivers, the receiver nearest to the sender will detect the signal first. The position of the sender is thus somewhere on the branch of the hyperbola that has the nearest receiver as focal point. Figure 2.14 exemplifies such a hyperbola with two known locations l_0 and l_1. The relevant branch is drawn as solid line. Every point on the hyperbola results in the same TDOA measurement (2 ms in the example).

In two-dimensional space, a position can be fixed by intersecting at least two hyperbola obtained from two TDOA measurements. To pinpoint a location in three dimensions, three TDOA measurements are required, resulting in three hyperboloids (instead of hyperbola) intersecting in one point. As it was the case with TOA, measurement errors lead to an area of possible locations instead of a single point. Positioning with the help of TDOA is also often called *multilateration* or *hyperbolic positioning*.

For egocentric positioning, pairs of senders must be synchronized to ensure that their signals are sent simultaneously. Furthermore, the receiver must be capable of identi-

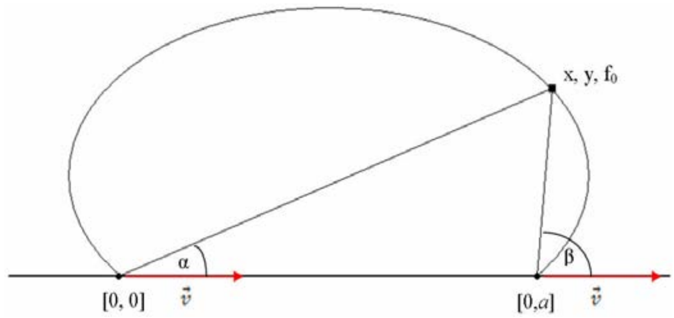


Figure 2.15: An iso-Doppler contour results from the measurement of one FDOA measurement ([Vesely, 2010]).

fying the origin of each signal and the locations of the senders must be known. With this setup, the receiving unit can determine TDOA measurements between pairs of senders and calculate its own position with the same principles as described in the exocentric case.

The main advantage of TDOA over TOA is that only the deployed infrastructure has to maintain synchronization, i.e. either the installed senders or the installed receivers (cf. [Appleyard et al., 1988], pp. 76–79). Due to this property, TDOA can also be used to locate unknown signal origins, for example cosmic gamma-ray bursts as described in [Klebesadel et al., 1982].

2.5.1.5 Frequency Difference of Arrival (FDOA)

If senders and receivers are in relative motion to each other, Frequency Difference of Arrival can be applied. The relative movement causes a signal shift in the frequency domain – the so-called Doppler shift – that can be observed on the receivers' end. The FDOA is derived by subtracting the Doppler shifts of different sensors or signals ([Mušicki and Koch, 2008]). In order to determine a position, the relative velocity as well as the locations of the senders or receivers have to be known. One FDOA measurement results in a so-called iso-Doppler contour, as shown if Figure 2.15. Again, a position can be fixed by intersecting at least two iso-Doppler contours in two dimensions or three contours for three dimensions.

2.5.2 Triangulation

In the context of positioning, *triangulation* is the process of calculating the coordinates of an object by measurement of angles and at least one distance in a triangle.

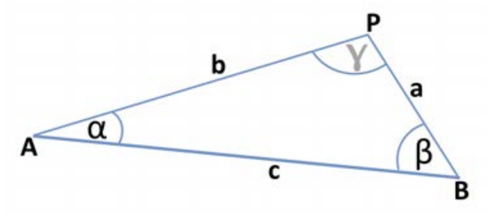


Figure 2.16: The principle of triangulation

The basic idea of triangulation is depicted in Figure 2.16. If A and B are known points, then P can be calculated by measuring the angles α, β and by using the law of sines ($\frac{a}{\sin\alpha} = \frac{b}{\sin\beta} = \frac{c}{\sin\gamma}$) and the fact that the three angles in a triangle sum up to 180 degrees. In two-dimensional space the calculation is as follows:

$$\gamma = 180 - \alpha - \beta \tag{2.7}$$

$$a = \frac{c}{\sin\gamma}\sin\alpha \tag{2.8}$$

$$b = \frac{c}{\sin\gamma}\sin\beta \tag{2.9}$$

2.5.2.1 Angle of Arrival (AOA)

To accomplish triangulation, the angles of arriving signals have to be determined. This can be accomplished by the use of a rotating antenna with a highly directed field of view. By internally tracking the rotation angle of the antenna, the angle of the incoming signal can be determined by observing at which angle the highest signal strength is reached.

Alternatively, a static antenna array can be used, consisting of several spatially arranged antennae. These antennae can either be directed, with each antenna pointing to different directions, or undirected antennae are used.

In the first case, the TOA can be determined by observing which antenna receives the highest signal strength. The angle can be fine-tuned by interpolating between all antennae that receive the signal and weighting according to the received signal strength. In the latter case, the spatial arrangement often dependents on the frequency or frequency range of the expected signals, such that phase-differences can be measured. The AOA can then be derived by measuring these differences between the individual antennae, which usually also correlates with the TDOA.

2.5.3 RSS Fingerprinting

Fingerprinting differs from the above approaches in that it is not purely geometrically based. The main idea here is to use previously made relative signal strength (RSS) measurements as indicators for the current position. In general, a fingerprinting approach is divided into two phases: a so-called training-, calibration-, or offline-phase and an actual positioning- or online-phase.

In the egocentric case, the calibration is done by taking repeated measurements at a number n of reference points p_n with known coordinates (x_n, y_n). These measurements include IDs of the received senders as well as the measured RSS of each sender. The IDs and their averaged RSS are stored in a database, together with the coordinates of the reference point. These are the so-called reference fingerprints and the resulting database is a fingerprinting map. More formally, a reference fingerprint rf_i for a reference point p_i is a vector of averaged RSS measurements for each detected sender $rf_i = [r_{i0}, r_{i1} \ldots r_{im-1}]$, with $0 \leq i \leq n$ and m detected senders. In the exocentric case, a sender is placed at the reference points and the sensors in the environment report the measured RSS of the sender to a centralized server, which then creates the fingerprint map, again consisting of the coordinates of each reference point, the averaged RSS and the IDs of the sensors that sensed the sender.

In the actual position phase, a fingerprint $f = [cr_0, \ldots cr_m]$ is made at the current location and the positioning algorithm tries to estimate which reference fingerprint rf_i most closely resembles the currently measured one. A simple approach to do this, is the Nearest Neighbor algorithm: The Euclidean distance D_i from the current fingerprint f to each reference fingerprint rf_i is calculated with

$$D_i = \sqrt{\sum_{j=0}^{m-1} (cr_j - r_{ij})^2} \tag{2.10}$$

The fingerprint with the smallest distance D_i is assumed to indicate the correct position and thus the reference point associated to that fingerprint is returned. This method has the disadvantage that no intermediate coordinates, i.e. coordinates that lie between reference points, can be returned. The k-Nearest Neighbor (kNN) algorithm overcomes this restriction by returning the mean of k reference points with the lowest calculated distance ([Laoudias et al., 2011]). A further variation is the k-Weighted Nearest Neighbor algorithm, which calculates the weighted mean of k reference points, where the inversed Euclidean distance of each reference point can be used as weight ([Chernoff and Nielsen, 2010]).

2.6 Methods for Sensor Fusion

As indicated in Section 2.1.1.2, a combination of different senses is often used in natural positioning. The technical term for such a combination is *Sensor Fusion*. For positioning systems, sensor fusion is generally used to gain higher position accuracy, when data from different types of sensors are available at the same time. This is one of the key-features needed in order to realize the Always Best Positioned paradigm (see Definition 1.3). Furthermore, an ABP system has to be able to work with any subset of the used sensors, in particular if only data from one sensor is available. In the following, three methods for probabilistic position determination will be described. All of them are based on Bayesian inference.

In general, a positioning system, may it be egocentric or exocentric, can be regarded as a system that represents the position of an agent as a state vector $s_t \in \mathbb{R}^n$, where $t \in \mathbb{N}$ denotes a so-called time-slice and $n \in \mathbb{N}$ denotes the dimension of the state vector. The state can contain more information than just the agent's position, i.e. its current velocities. A concrete example is $s_t = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^T$, where x, y, z denote position coordinates and $\dot{x}, \dot{y}, \dot{z}$ denote velocities; ^{T} indicates a translation. Moreover, a state vector could also contain semantic descriptions instead of numerical coordinates.

The task of a positioning system is to estimate the current (position) state given a series of observations or measurements $z_{1:t} = \{z_1, z_2, \ldots z_t\}$, where each measurement $z_i \in \mathbb{R}^m$ describes a measurement vector with m dimensions at the ith time-slice.

The estimation of such a system will contain inaccuracies introduced through measurement errors, noisy sensors et cetera. These inaccuracies can be expressed through probabilities. In the following, the notation $P(A)$ will be used to denote the probability of an event A, e.g. $P(AtWork) = 1.0$ means that the probability of the event

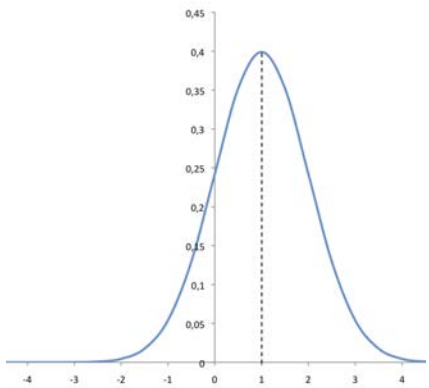


Figure 2.17: An example for a probability distribution function.

$AtWork$ is 100%. A probability density function (PDF) is a function that describes the probability of a random variable, or of a vector of variables, at a given point. A PDF of a random variable r will be denoted as $p(r)$. Figure 2.17 shows an example of a PDF. More specifically, it shows a Gaussian distribution $N(\mu, \sigma)$ with mean $\mu = 1.0$ and standard deviation $\sigma = 1.0$. If this were the PDF $p(s_t)$ of a one-dimensional state vector s_t, it would indicate that the probability of being at coordinate 1.0 is $P(s_t = [1.0]) = N(1.0, \mu, \sigma) \approx 0.4$. The probability of being at coordinate 0.0 would be $P(s_t = [0.0]) = N(0.0, \mu, \sigma) \approx 0.24$ and $P(s_t = [8.0]) \approx 910^{-12}$. In other words, with the help of a PDF, probabilities for all locations can be derived. The location with the highest probability can be considered as the current position.

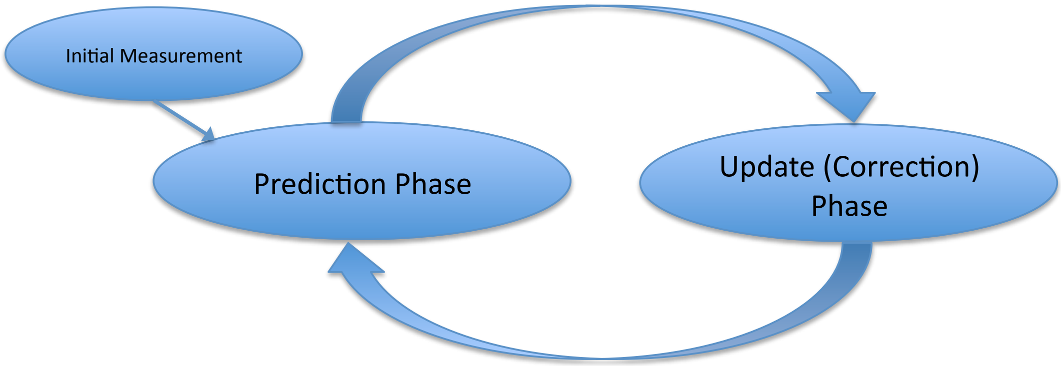
One way to accomplish the task of estimating the current state given a series of measurements is to derive the PDF $p(s_t \mid z_{1:t})$, i.e. the PDF of the current state under the condition of observed measurements $z_1, z_2; \ldots z_t$. Generally, the measurement z_0 is regarded as being an empty measurement as it is commonly used to derive an initial state s_0, i.e. $p(s_0 \mid z_0) \equiv p(s_0)$. The PDF $p(s_t \mid z_{1:t})$ is called the posterior PDF, as it includes all observed measurements up to time t. From this PDF, the position with the highest probability can be derived and reported as the current position. This method is used by Kalman filters and particle filters.

2.6.1 Kalman Filter

The Kalman filter was introduced in [Kálmán, 1960] and belongs to the family of Bayesian estimators. It uses a prediction model, a measurement model and error models for the measurement noise as well as for the error of the prediction to calculate the current posterior PDF. The basic principle of the Kalman filter is depicted in Figure 2.18a: an initial state $\hat{s}_0$ will be derived from a first measurement z_0. After this initialization, the filter will enter a recursive loop consisting of a prediction phase and an update or correction phase. In the prediction phase, the filter tries to estimate the next state using the prediction model. In the update phase, a new measurement will be used to correct the prediction from the previous phase. This corrected state will then be used for the next prediction and so on (see Figure 2.18a).

The Kalman filter has some restrictions: the prediction model and the measurement models must be linear and all PDFs can only be expressed as Gaussian densities. If these restrictions are met, the Kalman filter is an optimal estimator, i.e. it minimizes the mean square error of the estimated parameters.

Because of the required linearity of the prediction and measurement models, both models can be expressed through matrices. The prediction model tries to estimate the next state of the system $\hat{s}^-_{t+1}$, which is called the a-priori state as it does not yet contain a measurement for time $t + 1$. The a-posteriori state $\hat{s}_{t+1}$ is computed



(a) Kalman Filter Loop

Figure 2.18: The Kalman filter loop: after an initial measurement, prediction phase and update phase will be repeatedly executed.

in the update phase by taking a new measurement z_{t+1} into account. Because the Kalman filter is recursive, the prediction of the next a-priory state is based on the previous a-posteriori state. The prediction is calculated by using a state prediction model $A_t \in \mathbb{R}^{n \times n}$ and a probability variable v_t, which represents the inaccuracies introduced by A, the so-called process noise:

$$\hat{s}^-_{t+1} = A_t \hat{s}_t + v_t \tag{2.11}$$

v_t is assumed to be white noise with a noise covariance Q and normal probability distribution: $p(v_t) \sim N(0, Q)$. In practical applications, the process noise is often guessed or fine-tuned after a running system is implemented. In other words, the next state follows from the previous one by a translation plus the inaccuracies introduced by the translation itself. These inaccuracies are reflected in the so-called prior PDF $p(s^-_{t+1} \mid z_{1:t-1})$. In a positioning system, A_t would include the speed and direction of an agent at time t to compute the next state, i.e. the next position, by using Newtonian physics.

From the calculated a-priori state, a prediction for the next measurement $z'_{t+1} \in \mathbb{R}^m$ can be made:

$$z'_{t+1} = H_t \hat{s}^-_{t+1} + n_t \tag{2.12}$$

Here, $H_t \in \mathbb{R}^{m \times n}$ translates a state into a measurement and n_t is a random variable representing the measurement or sensor noise with noise covariance R. $\hat{s}^-_{t+1}$ can be seen as a hypothesis and z'_{t+1} describes which measurement is needed in order to confirm the hypothesis. Again, the added noise has an effect on the resulting PDF $p(z'_{t+1} \mid \hat{s}^-_{t+1})$. In general, a probability that an observation occurs under the condition that a defined state is given, is called *likelihood*. This in contrast to a probability that a state occurs under the condition of a specific observation, which is called *belief*.

(a) Initial Measurement (b) Prediction (c) New Measurement (d) Update (e) Prediction

Figure 2.19: Example of a Kalman filter for positioning.

In the update phase an actual measurement $z_{t+1} \in \mathbb{R}^m$ is taken. The discrepancy between the actual measurement and the predicted measurement, i.e. $z_{t+1} - z'_{t+1}$, is called the measurement innovation. The a-posteriori state is calculated as a linear combination of the a-priori state and the measurement innovation:

$$\hat{s}_{t+1} = \hat{s}^{-}_{t+1} + K_{t+1}(z_{t+1} - z'_{t+1}) \tag{2.13}$$

$K_{t+1} \in \mathbb{R}^{n \times m}$ is called the Kalman gain and is computed from the prior and likelihood PDFs. The Kalman gain also changes in every iteration of the filter and thus the weight between the a-priori state and the measurement innovation is shifted accordingly. With very noisy sensors and a good prediction model, the weight will be gradually shifted towards the a-priori state (cf. ([Maybeck, 1979, Welch and Bishop, 2006]).

The Kalman filter can be used to fuse the measurements of a position-giving system, i.e. a system that derives a position out of measurements (like a GPS receiver), with velocity- and direction-giving sensors, e.g accelerometers, by using the latter to predict the next position. Figure 2.19 exemplifies this approach: (a) a first GPS position is measured and the sensor-noise model is used to construct an initial posterior PDF (black). (b) the Kalman recursion is entered and a prediction for the next position is made. The resulting prior PDF (red) is flattened out in comparison to the initial posterior PDF (black) because of the process noise. (c) A new GPS measurement is taken. The PDF of the measurement (green) has the same shape as the initial posterior, since the same sensor-noise model is used. (d) The update phase provides a new position, which lies between the first measured position and the predicted position according to the Kalman gain. The resulting posterior PDF (black) is slightly sharper than the prior and the measurement PDF. (d) The Kalman filter enters the next iteration and generates a new prediction (red).

The method used in the example is called a loosely coupled Kalman filter. Here, the derived position of a position-giving system is treated as measurement and the distance between the predicted position and the measured next position is used as measurement innovation. A tightly coupled Kalman filter predicts the actual measurements of the position-giving system, e.g. TOA measurements and uses the difference between the predicted measurements and the next measurements as measurement innovation.

As already mentioned above, the Kalman filter has some restrictions. The Extended Kalman Filter (EKF) reduces some of these limitations by allowing non-linear models through the use of local linearization. However, the state PDF is still limited to be Gaussian.

In [Wolpert and Ghahramani, 2000] the model of a Kalman filter is used to explain how humans are capable to compensate for sensorimotor delays and noise inherent in sensory and motor signals. For example, a visual perception can easily be delayed by 100 ms, from the moment stimuli hit the retina until the signal reaches the according regions of the brain. According to the authors' theory, the brain uses a copy of a motor command, a so-called efference copy, and a model to predict the current state from the previous state, from which then the expected sensory feedback is predicted. The error between this prediction and the actual sensory feedback is then used to correct the estimate.

2.6.2 Particle Filter

As indicated above, the use of the Kalman filter is restricted to applications, where the system-state PDF can be described by a Gaussian density function. In practice however, this is often not the case, e.g. when different sensors report different positions. Figure 2.20 shows a non-Gaussian pdf, indicating two different one-dimensional positions. In general, the problem with non-Gaussian distributions is that they cannot be expressed through a uniform description and thus providing a general algorithm proofs somewhat difficult.

Particle filters tackle this problem by using Independent and Identically Distributed (IID) random samples of the system-state PDF. In the following, the same notation as in Section 2.6.1 is used, i.e. $s_t \in \mathbb{R}^n$ represents a system state at time t, $z_t \in \Re^m$ represents a measurement taken at time t. $s_{0:t}$ and $z_{1:t}$ represent the set of all system states and measurements until time t.

As it was the case with Kalman filters, a particle filter has a prediction model and a measurement model. Since particle filters can also deal with non-linear state models,

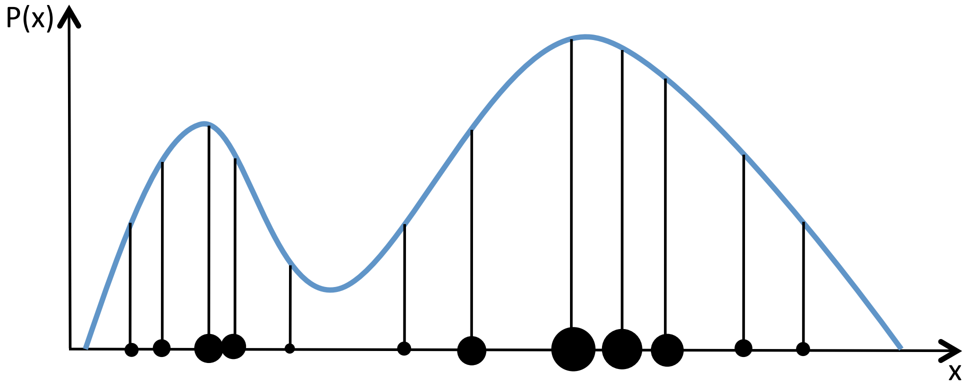


Figure 2.20: A non-Gaussian PDF and its approximation through weighted random samples (particles).

both models can be expressed through the use of functions:

$$s_{t+1} = f_t(s_t, v_t) \tag{2.14}$$

is the prediction model, with $f_t : \mathbb{R}^n \times \mathbb{R}^{n_v} \rightarrow \mathbb{R}^n$ and v_t being a noise vector with dimension n_v. With the help of Equation 2.14, the PDF $p(s_t \mid s_{t-1})$ can be computed.

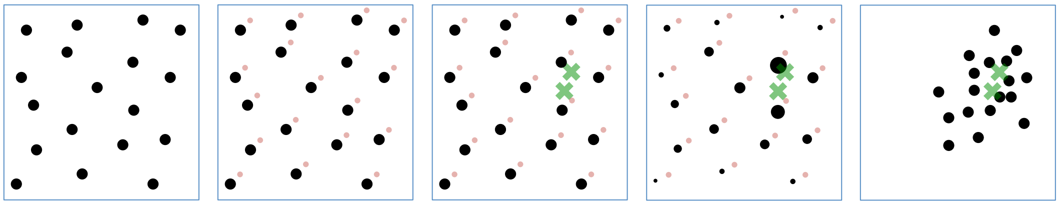
$$z_{t+1} = h_t(s_t, n_t) \tag{2.15}$$

is the measurement model, with $h_t : \mathbb{R}^n \times \mathbb{R}^n_n \rightarrow \mathbb{R}^m$ and taking a state description s_t and a noise vector n_t with dimension n_n as input to predict a measurement.

A particle consists of a so-called support point $s^i_{0:t}$, which represents a partition of the system state at time t, and an associated weight w^i_t. A set of particles $\{s^i_{0:t}, w^i_t\}^{N_p}_{i=1}$ is used to characterize the posterior probability function $p(s_t \mid z_{1:t})$, where N_p is the number of used particles. The weights are normalized, i.e. all available weights sum up to 1 and each weight w_i is proportional to $p(s^i_t \mid z_{1:t})$. The posterior probability can then be approximated as the weighted sum over the contributions of each support point to the complete system state, which can be calculated using the Dirac delta measure δ:

$$p(s_{0:t} \mid z_{1:t}) \approx \sum_{i=1}^{N_p} w^i_t \delta(s_{0:t} - s^i_{0:t}) \tag{2.16}$$

A set of particles thus divides a state PDF into discrete partitions. In terms of positioning, each particle $s^i_{0:t}, w^i_t$ is a hypothesis stating that the current position is at the particle's support point $s^i_{0:t}$. The particle's weight $w^i_{0:t}$ is proportional to the probability that this hypothesis is true. Through the use of the prediction model, the



(a) Initial state (b) Prediction (c) New Measurement (d) Update (e) (optional) Resampling

Figure 2.21: Example of a particle filter for positioning.

measurement model and a new measurement z_{t+1}, the weight of each particle is updated proportionally to the probability that the next position will be at the particles support point. Particles that are far away from the newest measurement, i.e. hypotheses that do not have a high support from the taken measurement, will thus have a lower weight than those that are closer.

There is however a problem: the state PDF is usually not directly accessible. In order to solve that problem, a so-called importance density $q(\cdot)$ is used, which has to be proportional to the state PDF. In practical applications, the last posterior PDF is often used as importance density. As it was the case with the Kalman filter, particle filters are executed recursively, thus an initial posterior PDF has to be created. Figure 2.21 shows an example in the positioning domain. The boxes represent a finite two-dimensional state space. In (a) an initial state is created by randomly distributing particles, indicated as black dots, over the state space. Each particle has the same weight and thus an evenly distributed PDF is represented. (b) Using the prediction model and the measurement model, predictions for each particle are created (marked in red). These 'prediction particles' are for illustration purposes only. (c) A new measurement is taken by two different sensors, which results in two possible positions (marked as green crosses). (d) The weights of each particle are adjusted according to how well each particle fits the new measurement. The new weights are normalized, so that they sum up to 1 again. Note that the particles do not change their position, they still represent the probability that the current position is at each particles initial support point. Step (e) will be explained further below.

It becomes clear from this example, that the number of particles – and thus the particle density – has a high impact on the accuracy of the position determination. Moreover, a central problem of this approach is visible: after a small number iterations, a few particles will gain most of the weight, while the other particles' weight will quickly become insignificant. This effect is called the degeneracy problem. Since computations have to be performed for every particle, most of the computations will then not really contribute to the position determination and thus the effectiveness of the particle filter is impaired. A measure for the effectiveness of a particle filter $N_{\text{eff}}(t)$ at time t can be derived from the summed squares of all weights:

$$N_{\text{eff}}(t) = \frac{1}{\sum_{i=1}^{N_p} (w_t^i)^2} \tag{2.17}$$

The lower $N_{\text{eff}}(t)$, the less effective is the particle filter at time t. To overcome the problem of degeneracy, resampling can be used when $N_{\text{eff}}(t)$ falls below a defined threshold. In the resampling step, N_p new particles are drawn out of the current state PDF, which can be approximated by Equation 2.16. The basic idea is to keep the particles with high weight and to shift those with low weight closer to the high weight ones. This is sometimes also referred to as 'survival of the fittest'. The weight of all particles in the new particle set is reset to $1/N_p$, i.e. the weight is equally distributed. The effect of such an additional resampling step is depicted in Figure 2.21e: The particles form a cloud around the high probability area of the state space. Resampling thus not only helps to keep the efficiency up, but also increases the accuracy as the distances between the hypotheses are reduced (cf. [Arulampalam et al., 2002, Gordon et al., 1993, Gustafsson et al., 2002]).

Particle filters are often used because of their ability to cope with non-Gaussian state PDFs and non-linear prediction- and measurement-models. The main disadvantages are the high computational complexity and the determination of a sufficient number of particles. Both parameters interact with each other: the higher the number of particles, the higher the computational complexity.

2.6.3 Bayesian Networks

Since Bayesian Networks, and moreover, dynamic Bayesian Networks play an important part in Chapter III, they will be described in more detail in the following. As a matter of act, Kalman filters and particle filters are subsets of dynamic Bayesian Networks ([Diard et al., 2003]).

Bayesian Networks (BNs) and their extension – Dynamic Bayesian Networks (DBN) – are a computational framework for the representation and the inference of uncertain knowledge via probability theory. As a matter of act, Kalman filters and particle filters are subsets or special applications of dynamic Bayesian Networks. The term 'Bayesian Networks' and their basic concept was introduced by [Pearl, 1985]. In the aforementioned article, Pearl argues that the straightforward way of implementing probability-based reasoning by using a joint probability distribution quickly runs into complexity problems. The main reasons for this being the exponential memory requirements to store the joint probability table for n propositions $x_1 \dots x_n$, and the

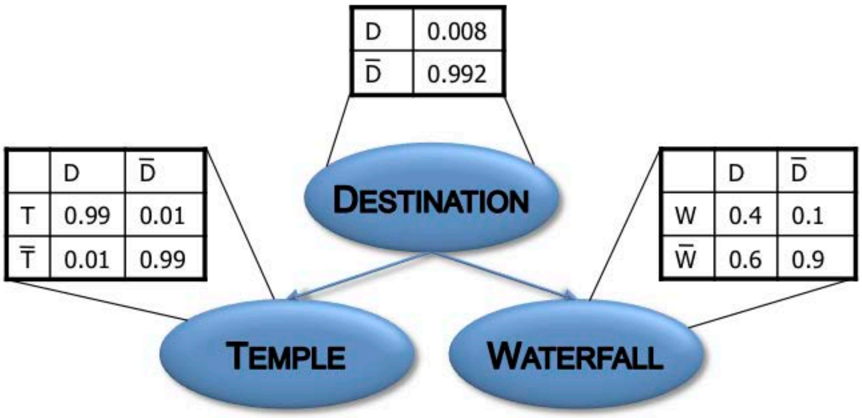


Figure 2.22: An example for a Bayesian Network showing the directed acyclic graph and the conditional probability tables.

exponential runtime to calculate the marginal (i.e. unconditional) probability for any proposition x_i ($1 \leq i \leq n$), which would be needed twice to compute conditional probabilities such as $P(x_i \mid x_j)$. Furthermore, Pearl elaborates that humans are in general very good at judging the dependence or independence between propositions, while being reluctant when asked about numerical estimates for conditional probabilities of propositions. Therefore, his proposed approach consists of a *qualitative* part, which describes the dependency between propositions, and a *quantitative* part, which describes the (estimated) numerical probabilities.

To model the qualitative part, directed acyclic graphs (DAGs) are used, where the nodes represent random variables and the directed edges represent direct dependencies between these variables. An example of such a DAG is shown Figure 2.22. The shown graph models the situation, where an agent tries to reach a certain destination, which it can identify by two routemarks. As explained in Section 2.1.1.4, routemarks differ from landmarks in that they are closer to a particular point on a route. The top node, labeled 'Destination' represents the probability whether the agent has reached its destination or not. The node thus contains two states: 'true' and 'false'. The bottom left node, labeled 'Temple' represents if the agent has visually perceived the temple, which marks his destination. It contains two states 'true' and 'false'. The last node, 'Waterfall', represents the event of acoustically perceiving a waterfall, which is also close to the destination but visually hidden in a dense forest. Like the other two nodes, it contains the states 'true' and 'false'.

Whether or not the agent sees the temple, directly depends on whether or not it has reached its destination, thus a directed edge leads from the node 'Destination' to the node 'Temple' (the direction is indicated by the arrow in the graph). The same is true for the event of hearing the waterfall. Thus 'Waterfall' also directly depends on

'Destination' and also has an edge starting from 'Destination'.

More formally, a directed graph G is defined by set of nodes, or vertices, V and a set of directed edges E ($G = (V, E)$). An edge can be represented by an ordered pair $(v, w) \in E$, with $v, w \in V$. A graph is acyclic if no path exists that starts and ends at the same node. A node v_p that directly influences another node v_c, i.e. there is an edge $(v_p, v_c) \in E$, is called parent of v_c.

In order to describe the *quantitative* part of the network, each node contains a conditional probability table (CPT), which describes the effects of the parent nodes on this node. The number of entries in each CPT is determined by the number of parent nodes and the number of states of the parent nodes and the node itself. In the case of the example network, the node 'Destination' does not have any parent nodes. Its CPT therefore only contains two entries, describing the probabilities whether the agent has reached its destination or not. Because of the lack of parent nodes, these probabilities are unconditional, i.e. they are a-priori probabilities. In this example, a very low a-priori probability of $P(Destination = true) = 0.8\%$ is assumed for the event that the agent is at its destination. Terms like $P(Destination = true)$ or $P(Destination = false)$ are often shortened to more readable notations like $P(D)$ and $P(\overline{D})$. Since a CPT represents the exhaustive set of cases for a node, it follows that the a-priory probability that the agent has not reached its destination is $P(\overline{D} = 1 - P(D) = 99.2\%)$.

The CPT of the node 'Temple' contains four entries, since it is directly influenced by the node 'Destination'. Thus, its entries describe the probabilities of seeing the temple under the condition that the agent did or did not reach its destination. Using the abbreviations T for $Temple = true$ and $\overline{T}$ for the $Temple = false$, the CPT contains values for $P(T \mid D), P(T \mid \overline{D}), P(\overline{T} \mid D)$ and $P(\overline{T} \mid \overline{D})$. The reliability of tests or sensors is often expressed in terms of *sensitivity* and *specificity*, e.g. the percentage of cases where the sensor correctly classifies the temple as temple and the percentage of cases where the sensor correctly classifies other objects as not being the temple. These values can be retrieved through evaluations under laboratory conditions. For this example a sensitivity and specificity of 99% in each case is assumed, i.e. a highly reliable sensor. The sensitivity equates to $P(T \mid D)$ and the specificity equates to $P(\overline{T} \mid \overline{D})$. The values for $P(T \mid \overline{D})$ (false positives) and $P(\overline{T} \mid D)$ (false negatives) can be calculated by $1 - P(T \mid D)$ and $1 - P(\overline{T} \mid \overline{D})$, respectively.

The CPT of the 'Waterfall' node represents the probabilities of hearing or not hearing the waterfall under the condition that the agent has reached its destination or not. For the sake of an example, it is assumed here, that the auditive sensor is highly unreliable, and more than that, the possible perception of the waterfall highly depends on environmental factors, like the amount of water the river carries, which can make the sound of the waterfall louder or softer, and the current wind speed, which can drown out the sound. In such cases, the probabilities are often estimated, which also allows to incorporate influences that are not explicitly modeled in the DAG by adjusting the probabilities accordingly. In this example, a 10% chance is assumed that the agent's audio-sensor mistakingly reports a waterfall although the agent is not at its destination ($P(W \mid \overline{D}) = 10\%$). By analyzing the current and past weather condition, the agent may adapt this value accordingly. Again, the value of $P(\overline{W} \mid \overline{D})$ can be computed by $1 - P(W \mid \overline{D})$. For this example, the probability of hearing the waterfall when being at the destination, $P(W \mid D)$, is estimated to be a low 40%, and thus the probability of missing the sound of the waterfall is $P(\overline{W} \mid D) = 60\%$.

According to [Russel and Norvig, 1995], Bayesian Networks can be summarized as 'a graph in which the following holds:

1. A set of random variables makes up the nodes of the network
2. A set of directed links or arrows connects pairs of nodes. The intuitive meaning of an arrow from node X to node Y is that X has a direct influence on Y.
3. Each node has a conditional probability table that quantifies the effects that the parents have on the node. The parents of a node are all those nodes that have arrows pointing to it.
4. The graph has no directed cycles (hence is a directed, acyclic graph, or DAG).'

Inference in Bayesian Networks

Having such a graph and the associated CPTs, queries can be answered. For example the question 'How high is the probability that the agent sees the temple when being at its destination?' can be translated into $P(T \mid D)$ and the answer (99%) can be directly retrieved from the CPT of the 'Temple' node. In this example query, the value of the node 'Destination' was observed, i.e. its state was known. In general, such a node, whose value can be observed, is called evidence node. Nodes whose values are unknown are called hidden or latent nodes [Ben-Gal, 2008].

The question 'How high is the probability that the agent will hear the waterfall, not knowing if the agent is at its destination or not?' translates into the marginal probability $P(W)$, which is not directly accessible through the CPTs. However, the query can be answered by summing up the joint probabilities over all outcomes of the influencing node 'Destination'. This method is also called marginalization:

$$P(W) = P(W \mid D)P(D) + P(W \mid \overline{D})P(\overline{D}) = 10.24\%$$

Note that all probabilities needed to perform that calculation can again be retrieved from the CPTs of the network.

A more interesting question would be ‘If the agent sees the temple, how high is the probability that it is at its destination?’, which translates into $P(D \mid T)$. In contrast to the first example query, in which the evidence node was a parent to the queried node, the evidence node is now a child to the queried node. This type of inference is called *bottom-up reasoning* and can be performed by using Bayes rule, which is also the reason for the term ‘Bayesian Network’. The general form of Bayes rule is:

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} \tag{2.18}$$

Applied to the query above, this leads to

$$P(D \mid T) = \frac{P(T \mid D)P(D)}{P(T)}$$

Since $P(T)$ is not directly accessible from the CPTs, it has to be computed via marginalization, leading to

$$P(D \mid T) = \frac{P(T \mid D)P(D)}{P(T \mid D)P(D) + P(T \mid \overline{D})P(\overline{D})} = 44.39\% \tag{2.19}$$

The outcome is surprising given the high accuracy that is suggested by the assumed 99% sensitivity and specificity of the test. This is however the effect of the low a-priory probability of $P(D) = 0.8\%$ (when assuming that the a-priory probability of being at the destination is 20%, the probability $P(D \mid T)$ rises to 96.12%).

If the node ‘Waterfall’ is also observed, $P(D \mid T \wedge W) = 76.15\%$ or $P(D \mid T \wedge \overline{W}) = 34.74\%$ can be computed (here the original probability of $P(D) = 0.8\%$ was used), again by applying Bayes rule and marginalization. Especially the bottom-up reasoning makes Bayesian Networks a very powerful tool.

Of course the outcome of a Bayesian Network can only be as good or exact as its modeling, including the quality of the CPT entries. As stated above, missing or unknown influences in the graph can be compensated for in the CPT entries. Nonetheless, such a network can only represent the view or the belief of its architect. Thus, Bayesian Networks are also called *Belief networks* and computed evidences are called *beliefs*, e.g. the computed probability of $P(D \mid T \wedge W)$ is called the belief that the agent is at its destination under the observation of seeing the temple and hearing the waterfall. Other terms for Bayesian Networks are *Influence network* and *Causal network*.

To summarize, Bayesian Networks approach the problem of storing a complete joint probability table by decomposing it into CPTs, e.g. a complete joint probability table for 30 two-valued variables would need $2^{30} > 1$ billion table entries, whereas

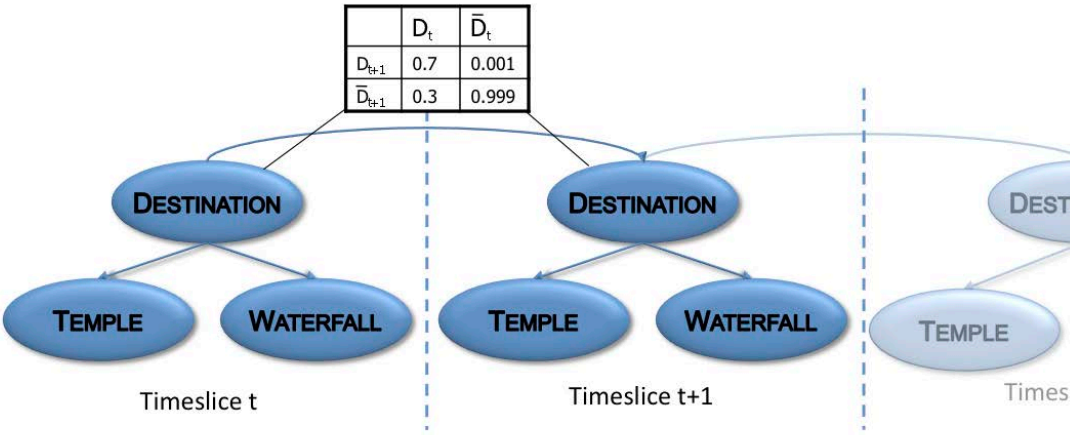


Figure 2.23: An example for a dynamic Bayesian Network including two time-slices and an inter-time-slice CPT.

a Bayesian Network consisting of 30 nodes, where each node has at most 5 parents only needs at most $30 \times 2^5 = 960$ CPT entries ([Russell and Norvig, 2003]). However, in general the exact inference in a Bayesian Network is still an NP-hard problem ([Cooper, 1990]), but efficient algorithms exist for certain network topologies, e.g. networks that only have at most one undirected path between any two nodes, so-called *polytrees* or *singly-connected networks*. [Pearl, 1986] describes such an algorithm, which is based on message passing, a technique that was also used by [Lauritzen and Spiegelhalter, 1988]. For large, multiply-connected networks, approximate inference can be used, e.g. Monte Carlo sampling ([Pearl, 1987]), logic sampling ([Henrion, 1986]) or importance sampling ([Fung and Chang, 1989]).

2.6.3.1 Dynamic Bayesian Networks

Dynamic Bayesian Networks (DBNs) are an extension of Bayesian Networks. With a DBN, it is possible to model dynamic processes: Each time the DBN receives new evidence a new time slice is added to the existing DBN. Figure 2.23 shows an example DBN with two time-slices and a single edge leading from time-slice t to time-slice $t + 1$. In this example, the CPT of this inter-time-slice edge models the probabilities for a change of states in the node 'Destination' from one time-slice to another. D_t and $\overline{D}_t$ denote the states of being at the destination in time-slice t and not being at the destination in time-slice t. Likewise, D_{t+1} and $\overline{D}_{t+1}$ denote the states of being or not being at the destination in time-slice $t + 1$. $P(D_{t+1} \mid D_t)$ thus is the probability that the agent will still be at the destination in the next time-slice under the condition that it is already at the destination in the current time-slice.

In the example, this probability is set to 70%. The probability that the agent will not be at the destination anymore in the next time-slice $P(\overline{D}_{t+1} \mid D_t)$ is thus 30%. The probability that the agent will reach the destination in the next time-slice although it is not there at the current time-slice is set very low: $P(D_{t+1} \mid \overline{D}_t) = 0.1\%$. The probability that it will still not reach the destination in the next time-slice is thus set to 99.9%. The CPT can be seen as an equivalent to the prediction model or motion-model in the Kalman filter and particle filter as it predicts the future state of the 'Destination' node based upon the current state. And as in the Kalman and particle filter, this prediction will be updated or corrected by new measurements, i.e. by setting new evidences in the nodes 'Temple' and 'Waterfall'.

DBNs can have an arbitrary number of inter-time-slice edges and these edges are not restricted to lead from one time-slice to the immediately following one. In principle, DBNs can be evaluated with the same inference procedures as normal BNs, but their dynamic nature places heavy demands on computation time and memory. This complexity can be greatly reduced, by applying roll-up procedures that cut off old time slices without eliminating their influence on the newer time slices. [Brandherm, 2006] describes sophisticated algorithms to apply these roll-ups and introduces a tool that automatically generates Java code for graphically modeled DBNs. In Chapter III a new positioning method based on DBNs is elaborated, which combines the principles of particle filtering with the power of DBNs.

3 RELATED WORK

3.1 Positioning with a Single Sensor Technology

3.1.1 Global Navigation Satellite Systems (GNSS)

The most prominent example for an outdoor positioning-system is the Global Positioning System (GPS), which is based on satellites orbiting the earth. In general, positioning systems based on satellites and pseudolites are called Global Navigation Satellite System (GNSS). The satellites of a GNSS usually act as senders, at least broadcasting a time-stamp that encodes the exact point in time when the broadcast began. A receiver on earth can then calculate its own position through TOA and trilateration, as described in Section 2.5.1. In theory, three satellites with known positions are enough to determine the receiver's position on the surface of the earth. However, as indicated in Section 2.5.1.3, the inaccuracies of the quartz timer on the receiver's end only leads to pseudoranges. These inaccuracies can be resolved by using a fourth satellite. In general, more satellites lead to a more accurate positioning. Because the data-flow in a GNSS is from the satellites to the receiver, GNSSs are egocentric positioning systems.

3.1.1.1 NAVSTAR GPS

As already indicated, GPS is the most well known GNSS system, with various applications in military as well as civil domains. The full name of the system is Navigation System with Timing and Ranging – Global Positioning System, or NAVSTAR GPS. It was deployed by the US Department of Defense, and reached its initial operation capability in 1993. Full operation capability was declared in 1995 ([Roth, 2005], p. 284).

GPS satellites operate on two frequencies called $L1$ and $L2$, with $L1 = 1575.42$ MHz and $L2 = 1227.6$ MHz. The $L2$ frequency is reserved for the so-called Precise Positioning Service (PPS), which is encrypted and can only be used by military applications. The $L1$ frequency is available for PPS as well as the Standard Positioning Service (SPS), which can also be used by consumers. Until 2000, the consumer SPS signals were artificially degraded trough a method called Selective Availability (SA) to ensure that only the US and NATO military could use high accuracy positioning. Since May 2000, SPS and PPS deliver the same basic accuracy, but since the PPS signals are sent on $L1$ and $L2$ frequencies, military applications can perform an ionospheric correction, which results in a higher accuracy.

Each satellite sends its own, unique Pseudo Random Noise (PRN) code, onto which additional data is modulated. A GPS receiver can simulate the PRN code of each satellite and thus knows how the signal looks like at the time the satellite sent it. The actually received PRN code is thus delayed against the simulated signal and the TOF (see Section 2.5.1.2) can be derived by shifting the simulated signal until it correlates with the received one. The accuracy of the TOF determination depends on how fast the bits of the PRN code are transmitted. In the case of SPS, a time resolution of 0.01 μs is possible, which leads to an accuracy of 3 meters for one distance measurement.

The bandwidth of the modulated information is only 50 bits per second and the payload is divided into three parts: clock-correction data, which contains the current number of the week and time; the ephemeris, which contains the orbit and health status of the satellite; and the almanac, which contains the coarse orbit and health status of other satellites. A GPS receiver typically has to build an internal database from the received data before it can provide a first position. The time that passes until the first position can be delivered is called Time To First Fix (TTFF). The TTFF depends on how much the satellite constellation has changed since the receiver was last turned on. In extreme cases, e.g. when the receiver has been brought to a different continent while being switched off, the complete database has to be rebuilt. This so-called cold-start takes 12.5 minutes when a clear view to one of the satellites is given. A warm-start is possible, when large parts of the database are still up-to-date, e.g. if the receiver has changed its position less than 300 kilometers since the last fix. A warm-start can be as fast as 45 seconds. A hot-start only takes 15 to 20 seconds and is possible when the database is up-to-date ([NavCen, 1996]).

Assisted GPS (AGPS)

A long TTFF is prohibitive for most mobile location-based services. To overcome this problem, assisted GPS downloads almanac data via a cell phone network or WiFi access from an assistance server. This server can also provide precise time and information about local ionospheric conditions, which can be used to derive a higher position accuracy. Furthermore, complex calculations can be offloaded from the receiver to the server, which allows to reduce the computational power of an AGPS device. In the case of AGPS via cell phone network, the cell phone provider can roughly estimate the current position of the device by the cell in which it is logged in, which also helps to reduce the search area for satellite signals. An AGPS system can bring the TTFF down to about one second.

In [Waters et al., 2011], a method is proposed in which several devices equipped with GPS and WiFi (see also Section 3.1.3) can assist each other similarly to AGPS. The idea here is that GPS devices that have already determined the correct GPS system-time, can act as access points according to the upcoming IEEE 802.11v WiFi standard, and insert the correct system-time in their data-packages. As the provided system-time has to be as accurate as possible, the timing of the transmission itself is highly critical. The authors propose, that integrated chipsets, which contain WiFi and GPS capabilities should be used, to minimize the transmission time.

Differential GPS (DGPS)

Differential GPS is a method to further increase the position accuracy through static reference stations with precisely known positions. These reference stations determine their own GPS position and use the difference to their exact position to calculate correction data. This correction data is usually broadcast via terrestrial radio and a DGPS enabled device can use the delivered data to correct its own position. The achieved accuracy depends on the distance from the receiver to the reference station and in order to be able to use the correction data, the DGPS receiver must use the same satellites as the reference station. According to [NavCen, 2001], the accuracy at the reference station is below 1 meter and degrades about 1 meter for each 150 kilometer of increased distance between receiver and reference station.

Wide Area Augmentation System (WAAS)

A Wide Area Augmentation System is based on the same methods as DGPS, but uses satellites instead of terrestrial radio to broadcast the correction data. Usually, a WAAS has a master control station, which collects the data of several reference stations. The calculated correction data is then sent to a geostationary satellite, which broadcasts the data on the $L1$ frequency using its own PRN identification. The advantage of WAAS over DGPS is, that no additional antenna has to be used to receive the correction data.

Standard GPS	2.0 - 8.76 m
Differential GPS	1 - 5 m
Wide Area Augmentation System	1 - 3 m
RTK GPS	$\approx$ 1 cm

Table 3.1: Accuracies of standard GPS, AGPS, DGPS, WAAS and RTK GPS.

Realtime Kinematic GPS (RTK GPS)

As mentioned earlier, the accuracy of the distance measurements from the receiver to each satellite is restricted through the bit-rate with which the PRN codes are transmitted. A way to further improve the positioning accuracy is to use the carrier-signal itself, which has a higher bit-rate, instead of the PRN code to measure the TOF and thus the distance to each satellite. However, since the carrier-signal is missing direct information that is needed to align the measured delayed signal with a simulated one, a statistical approach is needed to find this alignment. First approaches to this problem needed hours of carrier-signal measurements from one static position, which were then post-processed offline on a desktop computer. This method was called Static Surveying. Rapid Static Surveying was an improvement of this method and only needed a few minutes of measurements, but still had to be post-processed. Kinematic Surveying relied on Rapid Static Surveying but allowed the receiver to be moved after an initialization phase as long as it kept using the same satellites. With increasing computing power, it was eventually possible to perform the needed calculations on site and in realtime, thus the name Realtime Kinematic GPS (RTK GPS) ([van Diggelen, 1997]).

RTK is a refinement of DGPS, in that it also needs a precisely positioned reference station, but has higher restrictions. The reference station must be in a range of max. 50 kilometers of the moving receiver and at least 5 satellites must be available. With a clear LOS to at least 6 satellites, an accuracy of 1 centimeter can be reached.

Table 3.1 shows a comparison of the accuracies achieved by the different GPS based systems.

3.1.1.2 GPS Indoors

Like all known GNSS systems, GPS generally does not work inside buildings. This is mainly due to high attenuation of the satellite signals caused by exterior and interior walls. Although highly sensitive GPS receivers can sometimes pick up enough satellite signals next to windows or thin walls to be able attempt a position, the received signals are often distorted by reflection and diffraction. This distortion can cause a direct signal to be weaker than indirect ones and thus decreases the accuracy.

Ghinamo et al. In [Ghinamo and Gangyi, 2011], the authors tested GPS positioning in a light indoor-environment (see Section 2.3.1) and propose to use a particle filter approach, based on an empirical error distribution. They compared their approach with a weighted least squares method, and found out that the particle filter improves the accuracy in those conditions, where the error distribution is non-Gaussian. According to their experiment, conducted with a Sirf Star III high sensitivity receiver, they could achieve a position accuracy between 0.35 and 1.04 meters. The exact evaluation process, i.e. if only one position was measured or a moving receiver, was not disclosed.

3.1.1.3 GLONASS

GLONASS (Globalnaya Navigatsionnaya Sputnikovaya Sistema) is a Russian GNSS, which was completed in 1995. With the Russian financial crisis in 1998, the system could not be kept fully operational and in 2000 only 10 of the originally 24 satellites were still active. At the end of 2003, the restoration of GLONASS began, and since November 2011, the system is fully operational again.

As it is the case with GPS, GLONASS provides two different positioning accuracies: Standard Precision (SP) and High Precision (HP). In contrast to GPS, all satellites use the same PRN, but send on different frequencies. The center frequencies are $L1$ and $L2$, but different channels are used by adding offsets to them, i.e. $L1_{ch} = L1 + ch * 562.5$ kHz and $L2_{ch} = L2 + ch * 437.5$ kHz, were $ch = -07..06$ denotes a channel number. With 4.46 to 7.38 meters, the accuracy of an unassisted GLONASS receiver is in general slightly less than that of unassisted GPS.

3.1.1.4 Galileo

Galileo is a planned European GNSS, which is inter-operable with GPS and GLONASS. In contrast to the other two systems, Galileo will be completely under civilian control. Furthermore, the satellites will be placed in orbits at a greater inclination to the equatorial plane, which should increase the coverage in northern Europe and other areas with high latitude. The first Galileo test satellite, GIOVE-A, was launched in 2005, followed by GIOVE-B in 2008. With the launch of the first two of four navigation satellites in 2011, Galileo reached the third phase. At the end of the fourth phase, which is planned for 2014, 18 satellites will be in orbit and first services will be available. The system will be fully deployed in 2020, with 30 satellites in three circular medium earth orbit planes.

Galileo shares the $L1$ frequency with GPS, but uses a different modulation scheme. Additionally, the $L5$ frequency at 1176.45 MHz is used instead of $L2$. According to the official Galileo webportal[1], Galileo will provide several different services: the Open Service (OS), which is free of charge and provides 'position and timing performance competitive with other GNSS systems'; the Safety-of-Life Service (SoL), which will provide a service guarantee and can warn users when the accuracy drops below a threshold and a Commercial Service (CS), which will be encrypted and provides a higher accuracy. Officially, no accuracies are given, but Galileo is expected to provide the same accuracy as GPS.

3.1.1.5 BeiDou

The Chinese GNSS BeiDou (chin. for the Big Dipper asterism, formed by the seven brightest stars of Ursa Major) was declared operational for the region of China and its surrounding areas on December 27th 2011. In contrast to GPS, GLONASS and Galileo, BeiDou uses five geostationary satellites in addition to conventional non-geostationary satellites, and uses the China Geodetic Coordinate System 2000 (CGCS2000) instead of WGS84. BeiDou satellites operate on a carrier frequency of 1561.098 MHz, which is called the $B1$ frequency. The current system is also referred to as Compass or BeiDou-2 and is planned to reach completion in 2020, then comprising of 37 satellites and being operational world wide with a nominal accuracy of 10 meters ([China Satellite Navigation Office, 2011]).

The previous system, called BeiDou Satellite Navigation Experimental System or BeiDou-1, consisted of three satellites and differed from BeiDou-2, GPS, GLONASS and Galileo by being an offboard/exocentric positioning system: a user's terminal broadcasts a signal to the satellites, which in turn send the measured time of arrival to a terrestrial ground-station. The ground station determines the position of the terminal and sends this information back to the terminal via the satellites[2].

3.1.1.6 Pseudolites

Although global navigation satellite systems work well in outdoor scenarios with a free line of sight to four satellites, they have problems in urban canyons, and usually do not work within buildings. A proposed solution consists of so-called pseudolites, which can be received with the same hardware as the orbital satellites, but are installed terrestrial in said urban canyons or inside buildings. Although the idea sounds effective, it has some disadvantages, most notably the near-far problem: depending on the range between the receiver and the pseudolite, the signal can become more

[1] http://www.esa.int/esaNA/SEMTHVXEM4E_galileo_0.html; visited November 29, 2011

[2] http://www.cnsa.gov.cn/n615708/n620172/n677078/n751578/62676.html; visited December 29, 2011; translated with Google Translate

powerful than that of satellite and thus jam the receiver ([Ndili, 1994]). Solutions to this problem either modify the used frequency, pulse the signal or change the data protocol of the pseudolites, which also leads to hardware and software updates on the receivers side. In [Borio et al., 2011], a theoretical framework is presented that allows simulating and quantifying the signal loss of satellite signals, if pulsed pseudolites are used in addition to satellite signals. With the help of such a simulation, the deployment of pseudolites in buildings can be tested and optimized in terms of interference with satellite signals.

Indoor Messaging System (IMES)

A more concrete attempt to pseudolites is the Indoor Messaging System (IMES). IMES is a part of Japan's Quasi-Zenith Satellite System (QZSS), which itself is an extension of GPS through three additional satellites, which should improve positioning accuracy in Japan. The IMES specification states that in order to receive IMES data, only a small customization of existing GPS receivers has to be made ([QZSS, 2009]). The pseudolites use the $L1$ frequency with an offset of 8.2 kHz and use specially assigned PRN codes. Instead of the ephemeris data sent by GPS satellites, IMES pseudolites can send different data, e.g. longitude/latitude and floor-data or even a simple ID that refers to a database entry, which can then be accessed via a network. An IMES capable receiver usually does not use triangulation, but just adopts the received location of the pseudolite. According to [Dempster, 2009], IMES has to face several potential problems: pseudolites have to be installed every 20-30 meters, which could result in high costs; the pseudolites are likely to jam the reception of GPS satellites and could thus influence outdoor positioning; a seamless handover from pseudolite to pseudolite or to satellites could provide difficulties because of the near-far problem.

Kohtake et al. [Kohtake et al., 2011] describe a seamless indoor and outdoor positioning system based on IMES and GPS. They installed IMES pseudolites at various locations in a shopping mall, each one covering a distance of 10 to 20 meters and sending out a unique database ID. The used GPS receivers can read position information sent by the nearest pseudolite and measure the received signal strength of that satellite. If a receiver is in the middle of two pseudolites, both signals can be read and measured, which allows for a higher position-accuracy in those overlap zones. For testing purposes, the authors constructed a cart containing a GPS chipset receiver with modified software, a GPS antenna and a cell phone. The authors do not give an evaluation of the achieved position accuracy.

Sakamoto et al. [Sakamoto et al., 2011] use IMES pseudolites for robot localization. Since an accuracy of tens of meters is not applicable for robots and trilateration is impossible with IMES, they use Doppler shift effects to improve the positioning accuracy. In order to produce the needed Doppler shifts, they use two receivers, one with a stationary antenna and a second one with a movable antenna. Both receivers share the same clock and are thus tightly synchronized. The position of the movable antenna is always known by the robot in its own local coordinate system (LCS). The robot tries to determine its coordinates in a world coordinate system (WCS) by rotating the movable antenna around the fixed one, while the robot itself remains stationary. The authors conducted two experiments, in which either the rotation radius or the rotation angle was varied. In both experiments only one IMES pseudolite was used. The resulting measurements were stored and the positions were determined in an off-line phase. The resulting accuracy was highest with the longest rotation radius (300 millimeters) and biggest rotation angle (360 degrees) and resulted in $\approx$ 17 centimeters. However, because of the rotating antenna the proposed approach is not suitable for personal positioning.

3.1.2 Cellular Based

Modern mobile phones rely on a cellular network, consisting of radio towers. As already indicated in Section 2.3.3, opportunistic positioning systems use already existing infrastructure that was originally set up for a different purpose. Since cellular networks are widespread, they are often used for opportunistic positioning.

Cell phone standards are categorized in generations by the International Telecommunication Union (ITU). For each generation, key-features and requirement are defined, on which new standards are developed. The development of first generation (1G) mobile phones and networks started in the 1950s and was based on analog technology. In Germany, the first consumer-usable mobile phone network was available in 1958 and was called A-Netz (A network). The switch to digital technology, and thus to the second generation (2G), took place in 1991, with the start of the Global System for Mobile Communications (GSM) standard in Finland. Besides telephony, the second generation also introduced the capability of transferring data packets. The third generation (3G) was launched in 2001 by NTT DoCoMo in Japan and implemented the Universal Mobile Telecommunications System (UMTS). Candidates for the fourth generation (4G) are Long Term Evolution Advanced (LTE Advanced) and Worldwide Interoperability for Microwave Access 2 (WiMAX 2). Both standards have predecessors (LTE and WiMAX), which do not completely fulfill the 4G requirements, and are thus dubbed near-4G systems or 3.9G. Since 2G the main differences between each generation are higher data bandwidth and new frequency bands. The different generations are non-backwards compatible in transmission technology.

As the name implies, a cellular network consists of individual cells, each one covering a limited area. One such cell is created by at least one base station, consisting of a transceiver that operates on a certain radio frequency and that is able to maintain a connection to a limited number of mobile phones. In this context, terminal devices, such as a mobile phone, are called User Equipment (UE). Signals that are sent from a base station to an UE are called down-link signals and signals from an UE to a base station are called up-link signals. Adjacent cells, which can overlap, operate on different frequencies or frequency-bands, to minimize interference. However, if two cells are far enough apart, the same frequency-band can be reused. Through the partition of the whole network into cells, several problems are addressed. Firstly, the distance between an UE and a base station can be kept low, which is important for battery operated mobile devices, since the power consumption of such a device should be low. Secondly, if a sufficient number of cells is available, more UEs can connect to the network at the same time. However, for both solutions to work, a sufficient number of base stations has to be available, which can drive up the cost of deploying and maintaining such a network. To address this problem, several cell sizes can be used, e.g. large cells for rural areas and several small cells for urban environments (cf. [Roth, 2005], pp. 46). In the telecommunications domain, the names for different sized cells are derived from International System of Units (SI) prefixes:

- *Macro*cells are cells with a coverage up to 35 kilometers and are usually deployed in rural areas
- *Micro*cells cover up to 2 kilometers and are ideal for urban and suburban areas
- *Pico*cells have a range up to 200 meters and can be deployed in high-density areas, such as shopping malls or large office buildings
- *Femto*cells have a range up to 10 meters and are usually deployed and maintained by consumers

As indicated in the list, femtocells are operated by consumers. Because picocells and femtocells are especially designed for indoor usage and have a relative small range, they are particularly interesting for indoor positioning (see also Section 3.1.2.7).

The need for positioning in cellular phone networks was largely motivated by governmental regulations, such as E112 in Europe, E911 in North America and 110 in China. These regulations demand that network operators must be capable to locate any emergency caller within a given accuracy, i.e. 95% of all E911 callers within 150 meters. The following standard methods are often used to determine the position of an UE.

3.1.2.1 Cell ID (since 2G)

UEs connect to the base station that is nearest to them, i.e. the one that provides the device with the highest signal strength. This cell is called the Cell of Origin (COO) and can be used for positioning, either ego- or exocentric. In the exocentric case, the network operator can check in which cell the UE is logged in, and can thus determine the area in which it is located. A simple egocentric approach is based on the information each cell broadcasts, i.e. the Cell ID. By having a table stored on the device that contains the location for each cell ID, the device itself can determine its current area. A network provider can also broadcast this location info for each base station, which obsoletes the provision of such a table. The accuracy of these simple approaches depends on the size of the current cell and ranges from 200 meters (picocells) to 35 kilometers (macrocells) ([Singh and Ismail, 2005]).

3.1.2.2 Cell ID + Timing Advance (since 2G)

The accuracy of Cell ID based systems can be increased by taking additional measurements into account. For example the received signal strength on either the UE or the base station can be used to further narrow the radius of the circle around the base station describing possible locations.

GSM uses a Time Division Multiplex Access (TDMA) method to share one frequency with several UEs, i.e. each UE gets assigned to a specific time slot inside a time frame. In order for a base station to receive a data-package in the correct time-slot, the UE has to compensate for the signal delay due to the distance between itself and the base station. In order to determine this delay, the base station measures the Round-Trip Time (RTT) between itself and an UE, i.e. the time that is needed for a signal to be received by the UE plus the time that is needed to receive an acknowledgment signal from the UE.

A quantized value of this measurement, called the Timing Advance (TA) value, is sent back to the UE. TA values range between 0 and 63 and each step represents a time-step of 3.7 μs ([3GPP, 1999, 3GPP, 2010]). A TA value of 1 thus represents the distance of $\frac{3.7\mu s}{2} * c \approx 550m$ and a value of 63 represents the maximum GSM cell-size of 35 kilometers. The TA thus already provides an approximation of the UE's distance to the base station and since the value is known to both, the network operator and the UE, it can be used for exo- and egocentric positioning.

This kind of enhanced Cell ID is called Cell ID+TA. However, since Cell ID+TA can only provide a resolution in 550 meter-steps, this method only provides an advantage over simple Cell ID in macro and micro cells, where the maximum cell range exceeds a single step.

3.1.2.3 Cell ID + Round Trip Time (since 3G)

3G systems, like UMTS, can determine the RTT with higher accuracy. This improvement is achieved through the larger bandwidth and thus shorter time-slots of the protocol and can be further improved by applying optional Location Measurement Units (LMU) to the base stations (called Node B since 3G). Moreover, 3G cells can be divided into sectors by using directional antennae, which can be used to further narrow down the possible area of an UE. According to [Borkowski and Lempiäinen, 2006], distance measurements with an accuracy between 5 and 36 meters can theoretically be achieved by using oversampling at Node B measurements. Practical measurements however achieve an accuracy between 150 and 450 meters.

Third generation networks also provide features called softer handover and soft handover. As the names imply, these features ensure a seamless handover of a moving UE's connection from one Node B to another. During softer handover, the UE gains access to two Cell IDs and one RTT value. When the UE is in soft handover, it has access to two or more Cell IDs and to two or more RTT values. In these cases, a more precise positioning is possible, by using trilateration (see Section 2.5.1). In [Borkowski et al., 2004] a comparison of Cell ID+RTT was performed, including single Cell, softer handover and soft handover scenarios. According to the authors, the accuracy heavily depends on the network topology and lies in the range of 16 to 440 meters. As reported above, single Cell ID provided a range estimation between 150 and 450 meters. Softer handover resulted in positions with an accuracy between 50 and 100 meters, and soft handover could achieve a rather constant accuracy of 16 meters.

3.1.2.4 Observed Time Difference of Arrival (OTDOA) (since 2G)

In the context of positioning in cellular networks, the term Observed Time Difference of Arrival (OTDOA) is often used for a multilateration based on Time Difference of Arrival (see also Section 2.5.1.4). OTDOA can be used for exocentric and egocentric positioning. The latter is dubbed Down-Link Observed Time Difference of Arrival (DL-OTDOA) and is only possible if the UE is capable of measuring signals from different base stations.

OTDOA measurements are already performed in GSM networks to realize the so-called pseudo-synchronous handover between two base stations, which requires the determination of a new TA for the new base station. According to [Silventoinen and Rantalainen, 1996], the rate with which OTDOA measurements are performed in a GSM network is too low for positioning. They propose a software change on the network and UE side to increase the measurement rate and tested their approach with a simulation. A mean positioning accuracy between 100 and 200 meters could be measured. However, the authors admit that the measurement error might be much higher in some real-world environments. [Singh and Ismail, 2005] report a medium accuracy between 100 and 500 meters in urban environments (due to closer distances to several base stations) and a few kilometers in rural areas.

In 3G networks, OTDOA measurements for egocentric positioning face the near-far problem (see also Section 3.1.1.6), where the serving Node B drowns signals from other base stations. A solution to this problem is called Time Aligned Idle Period Downlink (TA-IPDL), where each Node B suspends its transmissions for a given amount of time, which enables the UE to receive signals from distant Node Bs. According to [Borkowski and Lempiäinen, 2006], with TA-IPDL an accuracy between 30 and 100 meters can be achieved.

3.1.2.5 Angle of Arrival (AOA, only with additional hardware)

To enable exocentric positioning through triangulation within a cellular phone network, the base stations must be equipped with antenna arrays to be able to derive the AOA (see also Section 2.5.2.1). Egocentric positioning through trilateration is uncommon in cellular networks, since standard UE hardware is not capable of deriving AOA measurements. The accuracy through exocentric triangulation based on AOA measurements ranges between 100 and 500 meters ([Singh and Ismail, 2005]).

3.1.2.6 Positioning in 4G

The 3rd Generation Partnership Project[3] (3GPP) is an international consortium of telecommunications associations, which produces technical specifications. Originally the consortium was founded to define standards for 3G cellular networks, but is now also involved in 4G standardization.

3GPP specified three positioning methods for 4G:

- Network-assisted GNSS methods
- Cell ID + enhancements like RTT
- OTDOA

The network-assisted GNSS methods, e.g. AGPS, are specified as the main positioning method, while Cell ID methods and OTDOA are specified as fallback solutions. The motivation for this specification of positioning methods is mainly

[3] http://www.3gpp.org

motivated by the E911/E112 regulations (see Section 3.1.2), although the expected higher data-rates of 4G (up to 1GBit/s in LTE Advanced) are also envisioned to provide new opportunities for location based services. However, an increase in position accuracy of the proposed standard solutions is not expected (cf. [Tam and Lee, 2009, Ranta-aho, 2010]).

Pereira F. et al. In [Pereira et al., 2011b], an approach is described how to use GSM fingerprinting to derive position information in the Large Hadron Collider (LHC) tunnel, located near Geneva, Switzerland. The LHC is a particle accelerator ring with a perimeter of 27 kilometers, lying 100 meters below the surface. Inside the tunnel, positioning information could be helpful for the radiation protection group, who has to perform frequent radiation surveys involving radiation measurements at thousands of points.

Due to the location deep under ground, no GPS signals can be received inside the tunnel. However, GSM is provided inside the tunnel through several so-called leaky-feeder cables. The tunnel is divided into eight sections and for each section two GSM cells are created through the leaker-feeder cables, such that when following along the tunnel in one direction, the signal strength of one GSM cell rises and the other gets attenuated.

The authors used a modified Nokia 6150 mobile phone connected to a laptop to log the signal strength of different GSM cells and to create a fingerprint map. To find corresponding locations, a weighted k-nearest neighbor approach was used. The system was tested using test-locations inside the tunnel, and measurements were taken under optimal conditions (nobody was near the measuring equipment), sub-optimal conditions (at least one person was standing near the measuring equipment) and realistic conditions (a person was holding the equipment during measurement).

As expected, the signal variations where minimal in the optimal condition ($\pm$ 2.5 dBm) and were highest in the realistic condition ($\pm$ 6 dBm). The authors further noticed that significant differences arose between different measurement sessions, which may be due to magnets being powered in the particle accelerator equipment.

The measured accuracy of the positioning algorithms was determined being between 20 and 280 meters, taking all three conditions into account. The authors conclude that their system provides accuracy within an acceptable range (for their purposes) and that they will be able to enhance the system through a higher resolution of the fingerprinting map, a better signal measurement process and through the application of filtering techniques.

3.1.2.7 Indoor Positioning with Femtocells and Picocells

As already mentioned above, femtocells and picocells are especially interesting for indoor positioning applications, due to their short sending range. In contrast to picocells, femtocells are operated by consumers and bridge the user's cell phones to their cellular network operator by using the consumer's DSL or other broadband Internet access. As an incentive, operator companies usually offer reduced call rates or lower fees for their services. According to [Haddad and Porrat, 2009], cellular network operators consider femtocells as a solution for two problems: low signal reception of cell towers indoors, which usually degrades voice quality and leads to low data throughput, and losing profit because users tend to use WiFi and Voice over IP at home instead of the cellular network. The idea to use privately operated base stations – also called Home Base Stations (HBS) in contrast to the network operated base stations – has already been proposed in [Silventoinen et al., 1996] to connect GSM based phones with a fixed telephone line. The current concept of femtocells is promoted since 2007 by the Femto Forum, a "not-for-profit membership organization which seeks to enable and promote femtocells and femto technology worldwide"[4]. The first femtocell standard was published in 2009 and was the result of a three-way cooperation between 3rd Generation Partnership Project (3GPP), Femto Forum and Broadband Forum.

Dempsey et al. [Dempsey et al., 2011] describe a testbed, in which they implemented a Customer Relation Management (CRM) system, which should help to increase a company's interaction with customers, clients and sales prospects. Among other features, the CRM system provides location information for subscribers and a way to share location information with third parties. The hardware consists of three 2.5G proprietary GSM picocells, although the authors argue that the same principles can be used for 3G or 4G pico- and femtocells. The testbed also contains a centralized server to which the picocells are connected and which can gain information on the Cell ID and signal strength of every subscriber's mobile phone. Furthermore, the current connection status of a subscriber can be extracted, e.g. whether they are currently talking on the phone or not. The CRM system can poll this information in intervals of 20 seconds and also has access to the calendars of every subscriber.

Coarse position information is derived through the Cell ID information. Motion is inferred through changing signal strength between polling intervals and direction is estimated through Cell ID changes. So-called presence agents try to infer a more precise position or sub-status by combining the measured information with calendar entries through the use of Bayesian Networks. For example, if a subscriber has a scheduled meeting at 8:30 in room A on the ground floor and the measured cell at

[4] http://femtoforum.org/

8:25 is the one on the ground floor, the agent refines the position information to room A and changes the sub-status to 'in a meeting'. If the subscriber's cell is on the first floor, with an inferred direction downstairs (e.g. the previous cell was on the second floor), the sub-status can be changed to 'on the way to a meeting in room A'. The authors conclude that such a pico/femtocell based infrastructure provides 'cheap context' as the cost of a femtocell is less than $100.

3.1.3 WiFi Based

The term WiFi (or Wi-Fi) is a trademark of the Wi-Fi Alliance[5] and an abbreviation for Wireless Fidelity. It is used to describe the set of IEEE 802.11 standards ([IEEE, 1999]) for wireless data connections into Local Area Networks (LAN) and the Internet ([Patton et al., 2005]). Often, the term WLAN, for Wireless Local Area Network is used interchangeably. According to the IEEE 802.11 standard, wireless data connections can be established via infrared or radio communication. However, nowadays the terms WiFi and WLAN almost always describe the access via radio communication. As indicated above, the IEEE 802.11 is a set of standards that has evolved over time. The original 802.11 was proposed in 1997 and allowed data-rates up to 2 MBits per second on the unlicensed 2.5 GHz frequency band. In 1999, the extensions 802.11a, with data rates up to 54 MBits/s on the unlicensed 5 GHz band, and 802.11b, with data rates up to 11 MBits/s on the unlicensed 2.4 GHz band, where introduced ([Roth, 2005], page 81). Since then several extensions followed, where the most important ones are 802.11g, 802.11h and 802.11n, which further increase the data rates on the 2.4 GHz and 5 GHz bands. A special extension, named 802.11p, was proposed for vehicular applications and is a very important key-element for Car-to-Car (Car2Car) and Car-to-X (Car2X) applications.

All 802.11 standards support two basic connection modes: ad-hoc mode and infrastructure mode. In infrastructure mode, WiFi compliant Access Points (AP) are deployed into the environment. Mobile or stationary devices that connect to such an access point are called stations. An access point can handle connections to several stations and can thus create a network between these stations. Usually, an AP can also connect to a wire-based network or a broadband Internet access, and can thus integrate wireless stations in an already existing LAN or provide wireless Internet access. In the ad-hoc mode, a network is formed between wireless stations without the use of any AP. All stations must therefore be in appropriate range to each other.

Since WiFi uses unlicensed frequency bands, an infrastructure can be easily deployed by consumers and companies at relative low cost. WiFi access points in public places, like airports, trains stations, shopping malls or even parks and other municipal areas,

[5] http://www.wi-fi.org

are called hot spots and are often freely accessible. Nowadays, WiFi infrastructures are an integral part of universities, companies and even in many private homes. Because of this, WiFi infrastructures are a good choice for opportunistic positioning systems (see 2.3.3).

Each WiFi capable station and each AP has a unique 48 bit wide Media Access Control address (MAC address) that is used to identify each entity. APs also broadcast a Service Set IDentifier (SSID), which consists of up to 32 characters and can be freely chosen. The SSID is therefore not necessarily unique and is mainly used to name an AP or a specific WiFi network. The SSID can be hidden, but this is merely a matter of filtering out the information by the protocol on the receiving end. APs repeatedly broadcast their MAC address, SSID and additional data in so-called beacon frames to advertise their presence and services.

A scanning process on an AP reveals all available WiFi stations in reach and results in a list of their MAC addresses and the measured Relative Signal Strength (RSS) to each station (see also Section 2.5.1.1). A scanning process on a station results in a list of MAC addresses of nearby APs and, depending on the hardware, also the measured RSS values for each AP. With the use of this information, several positioning approaches are possible. Each of them can be used for egocentric or exocentric positioning:

- **Single AP:** If the location of a single AP is known, a simple proximity positioning is possible.
- **Single AP+RSS:** If the RSS is known in addition, the area of possible positions can be further reduced.
- **Multiple APs+RSS**: If the locations of several APs are known, trilateration is possible by using the RSSs to estimate distances (see Section 2.5.1).
- **RSS Fingerprinting**: If enough APs are available, signal strength fingerprinting can be used (see Section 2.5.3).

The first three methods require the knowledge about the positions of APs and the fingerprinting approach requires fingerprint maps. Usually this data is stored on a database, either on a server or directly on a device. In [Gschwandtner and Schindhelm, 2011] the authors propose to include additional data into the 802.11 beacon frames. This additional data includes the coordinate of the broadcasting AP itself or data that enables a mobile station to construct a fingerprint map. With this extension, the authors hope to minimize the deployment effort and needed storage capacities of WiFi-based positioning systems. The proposed protocol was implemented on a modified OpenWRT based WiFi access point, which proved the feasibility of the approach.

Bahl et al. [Bahl and Padmanabhan, 2000] describe a system called RADAR, which was one of the first WiFi based positioning systems using a fingerprinting approach, although they classified it as being based on triangulation. Besides the construction of a fingerprint map trough empirical measurements, they also proposed to use a radio propagation model as an alternative. The average positioning accuracy of RADAR lies between 2 and 3 meters.

Ledlie et al.: Molé In [Ledlie et al., 2011] an onboard/egocentric positioning system called Molé is presented, which relies on signal strength fingerprinting. Molé uses semantic descriptions instead of coordinates, which are organized in a hierarchy. This hierarchy limited to five levels: country, region, city, area and unique place. Trained fingerprints can be retrieved from a cloud-service and users can train unknown places and upload their fingerprints to the cloud. Molé thus uses some of the ideas already published in [Dimitrov, 2007] and [Schwartz et al., 2010b], which will be discussed in detail in Section 4.3.

Signal-strength fingerprints in Molé are expressed as a list of triplets $\langle w_i, \mu_i, \sigma_i \rangle$ for each detected WiFi access point AP_i at a particular position. Here, μ_i and σ_i describe the observed signal strength over a time period as a single Gaussian with mean μ_i and standard deviation σ_i. w_i is a weighting factor, which is derived from the observation how often an access point was measured during a time period. The weighting factors are normalized, i.e. all w_i in one fingerprint sum up to 1. The similarity of two fingerprints is computed by comparing each access point in the list and computing an overlap coefficient from both Gaussian distributions. For missing access points a penalty value is subtracted. The weighted sum over all access points is used as a confidence level. The authors call this method MAximum Overlap localization (MAO).

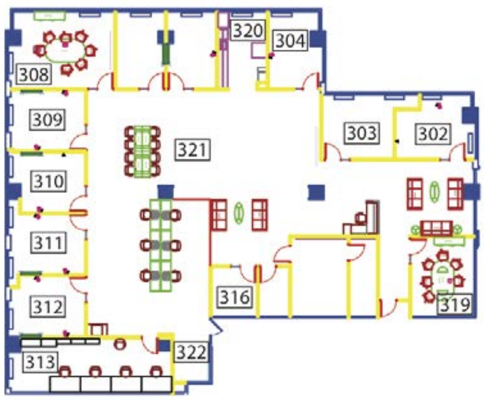


Figure 3.1: A test environment for Molé [Ledlie et al., 2011].

Molé was tested in different setups. The most meaningful, because densely located rooms were tested, was conducted in a lab of which the floor plan is shown in Figure 3.1. Fourteen Nokia N900 tablets with Molé running were placed in the 14 rooms indicated. The system was trained 24 hours prior to the evaluation. As Molé provides semantic descriptions of positions, the evaluation tested how often the correct room was guessed. Several variations of the algorithm were tested, including using histograms instead of a Gaussian distribution to describe the signal strength. The highest average hit-accuracy achieved was 93.16%, using MAO with signal strength histograms, where the rooms 309, 310, 311 and 312 had the highest miss-rates. The MAO with a Gaussian description resulted in an average hit-accuracy of 80.46%; again rooms 309 to 312 were the most problematic. Obviously, those rooms provide the highest challenge for such a system, as they are directly adjacent. When averaging the hit-accuracies over those four rooms, the average hit-rate drops to 76.57% for MAO with histograms and 41.98% for MAO with Gaussian distributions.

3.1.4 Bluetooth Based

Bluetooth is an open wireless data-transmission standard for short distances. The standard was initially started by Swedish company Ericsson after conducting a feasibility study on how to create a technology that is able to wirelessly connect different electronic devices, e.g. desktop computers, printers, mobile phones and laptops.

The Bluetooth Special Interest Group (Bluetooth SIG) was founded in 1998 by Ericsson, Nokia, IBM, Toshiba and Intel as a privately held, not-for-profit trade association, with the goal of constituting an industry standard and binding specifications. The first version of the Bluetooth specification was approved in 1999. As of 2011, over 1 billion devices are Bluetooth enabled, the core specification has reached version 4.0 and over 14,000 companies are members of the SIG ([Bluetooth SIG, 2008]).

The name Bluetooth stems from the Danish King Harald Blåtand, which translates to Harald Bluetooth. Under the reign of King Blåtand, several warring parties in parts of today's Norway, Sweden and Denmark were unified in the 10th century. Since the special interest group sought after a technology to unify different devices of competitive industries, they chose the name Bluetooth.

Bluetooth devices are classified according to their transmitting power and the resulting range of coverage:

- *Class 1*: 100 mW; range of coverage up to 100 meters
- *Class 2*: 2.5 mW; range of coverage between 10 and 20 meters
- *Class 3*: 1 mW; range of coverage up to 10 meters

Like WiFi, microwave ovens and cordless phones, Bluetooth operates in an unlicensed frequency band. For Europe and USA the range between 2,400 and 2,483.5 MHz is used, and in Japan the range lies between 2,471 and 2,497 MHz. In comparison to WiFi, Bluetooth is optimized for ad-hoc networks between Bluetooth devices, but provides a lower data rate of 3 MBits/s (Bluetooth Version 2.0). With Bluetooth Version 3.0+HS (High Speed), higher data rates up to 24 MBits/s can be achieved, but the actual data transmission is then realized via a WiFi link, while Bluetooth is only used for the initial handshake.

Up to eight Bluetooth devices can form a so-called piconet, where each device can again be part of several different piconets. Such a network, that contains Bluetooth devices that belong to different piconets, is called a scatternet. In each piconet, one device acts as the master device; all other devices act as slaves. As it was the case with WiFi networks, each Bluetooth device has a unique 48 bit wide Bluetooth address (sometimes called Bluetooth MAC or Bluetooth ID) and a so-called friendly Bluetooth name that can be freely chosen by the user. If a Bluetooth device wants to establish a connection, it performs an inquiry, which results in a list of all discoverable[6] Bluetooth devices in its vicinity as well as RSS indicators and a list of services for each detected device. According to the Bluetooth specification, such an inquiry requires around 20 seconds ([Bluetooth SIG, 2010]).

Positioning with Bluetooth can be accomplished by instrumenting an environment with fixed Bluetooth beacons or by using stationary Bluetooth enabled devices that are already in the environment, e.g. Desktop PCs, Bluetooth enabled input devices or Bluetooth enabled printers. The latter attributes for an opportunistic positioning system and a detailed example will be given in Section 4.3. For the former, the same principles as for WiFi positioning can be applied, but due to the low range of common Bluetooth devices, a proximity based approach is often used.

Eyeled: Indoor Navigation for the Visually Impaired The German company Eyeled GmbH[7] has developed an indoor navigation system for visually impaired people based on Bluetooth beacons with integrated speakers. Users can download the navigation system on their Bluetooth enabled Symbian OS mobile phone, either at home or on location via Bluetooth and will then be guided to their destination, for example inside a communal building. The Bluetooth beacons act as landmarks for the navigation system and can also output audible signals via the integrated speakers. According to a company spokesman, the Bluetooth beacons can also be integrated into the already existing signage inside the building[8].

[6] Whether or not a Bluetooth device is discoverable, depends on the user settings.

[7] http://www.eyeled.de

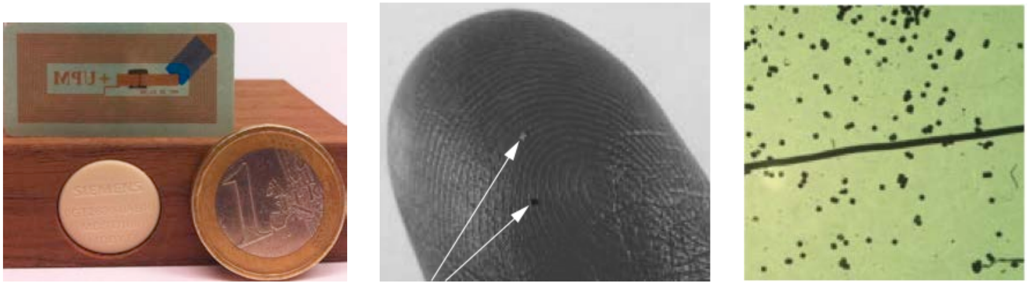
[8] Press release: http://www.eyeled.de/unternehmen/presse/index.php?File=005&news=IndoorNavigation

Chawate [Chawathe, 2009] describes an egocentric, onboard indoor-positioning system using Bluetooth beacons that addresses the important problem of long inquiry times. As already mentioned above, the inquiry process takes about 20 seconds, which is generally too slow for a user walking at normal speed. On the other hand, directly probing a Bluetooth device with known Bluetooth ID can be accomplished in at most 2.5 seconds. The author's idea is therefore, to use a beacon map on the user's Bluetooth device that contains the locations as well as the Bluetooth IDs of all available beacons in a building. With the use of this map and the knowledge about the previous position, the positioning system then directly probes for known Bluetooth beacons instead of starting a general inquiry. The needed beacon map is represented through a 4-tuple, which contains the locations of the beacons, edges between neighboring beacons, hyperedges that represent the range of a beacon as the set of locations from which it is detectable, and weights that represent the distance between two locations. The actual positioning is done through the hyperedges. Having observed a set of Bluetooth beacons, the system tries to find the location that best matches the observation. The author provides sophisticated algorithms on how to determine the set of beacons that have to be probed for in the next measurement. An evaluation of the system's accuracy is not given.

3.1.5 RFID Based

Radio Frequency IDentification (RFID) is a radio based technology and is based on modulated backscattered communication. First applications of the basic idea can be traced back to the second World War, where British airplanes modulated identification information onto incoming Radar signals. This modulated information could then be used to distinguish between friendly or attacking planes. The first research paper describing the basics of RFID was [Stockman, 1948], where the author predicted that 'considerable research and development work has to be done before the remaining basic problems in reflected-power communication are solved, and before the field of useful applications is explored'. The first commercial application of RFID started in the late 1960 with the introduction of Electronic Article Surveillance (EAP) equipment. These systems augmented merchandise with electronic circuits whose presence could be detected at a shop's cashing point or exit (cf. [Landt, 2005]).

In general, an RFID system consists of at least one reader and at least one so-called tag or responder. The tags contain at least some identification information (RFID ID), which will be revealed if an interrogation signal from a reader is received. Compared to the reader, a tag contains a very simple circuit, which means that they are usually very cheap to produce. In [Takaragi et al., 2002], the authors report about the so-called μ-chip manufactured by Hitachi, which is an RFID tag that is only 0.06 millimeters thick and 0.4 millimeters long on each side. In 2007, Hitachi announced



(a) Two standard RFID tags by UPM and Siemens in comparison to a 1 Euro coin

(b) μ-chips by Hitachi on a fingertip [Takaragi et al., 2002]

(c) Powder LSI chips by Hitachi in comparison to a human hair [Hornyak, 2008]

Figure 3.2: Comparison of passive RFID tag sizes, from standard tags (a) to μ-chips (b) to 'RFID Dust' (c).

even smaller tags, called Powder LSI (Large Scale Integration) chips, with a thickness of 0.005 millimeters and a side-length of 0.05 millimeters. However, standard RFID tags usually cover a much larger area, more in the range of centimeters than in sub-millimeters. Figure 3.2 shows a comparison of different RFID tag sizes: (a) shows a self-adhesive tag manufactured by UPM on the top and a Siemens tag, which is integrated into a wooded block; (b) shows two μ-chips on a fingertip and (c) shows numerous Powder LSI chips in comparison to a human hair.

RFID systems can be classified into passive and active. In passive RFID systems, the tags draw the needed operating power from the signal of the reader. Passive systems therefore only have a short communication range, typically up to 3 meters, but some reader/tag combinations can also reach up to 10 meters. In active RFID systems, tags are powered by batteries or other external power-sources, which generally leads to a higher reading range of up to 100 meters. The reading range in passive and active RFID systems depends on various factors, e.g. antenna design, reader power, tag power-consumption and the used frequency band. Like WiFi and Bluetooth, RFID systems use unlicensed frequency bands. In the case of RFID the following frequency bands are commonly used:

- **LF** (Low Frequency) 125 – 135 kHz
- **HF** (High Frequency) 13.56 MHz
- **UHF** (Ultra High Frequency) 868–928 MHz
- **Microwave** 2.45 GHz and 5.8 GHz

For passive RFID systems, the highest communication ranges are achieved in the UHF and Microwave bands. Active RFID systems usually operate only in the UHF and Microwave range (cf. [Dressen, 2004]). One of the problems of RFID is that its not truly standardized. As a matter of fact, there is a plethora of different standards available, where the most important ones are ISO 18000, which specifies protocols for different frequencies, ISO 14443 specifies so-called proximity RFID devices with a communication range up to 10 centimeters and ISO 15693 is for so-called vicinity RFID devices with a communication range up to 1 meter.

An important technology that is related to RFID, and in fact compatible to ISO 14443, is Near Field Communication (NFC). However, with NFC the strict distinction between reader and tag was repealed, i.e. each NFC-compliant device can act as reader or as tag (cf. [Juels, 2006]). NFC is expected to play an important role in making cashless payments, and thus NFC is already integrated in many smart phones. Through the compatibility to ISO 14443, NFC devices are also able to read passive RFID tags, which comply with the ISO norm.

Since RFID is a radio-based technology, the same principles for positioning can be used as with WiFi and Bluetooth. However, the small communication distance of passive RFID makes it especially suitable for proximity approaches. As a matter of fact, the already mentioned EAP systems are exocentric positioning-systems, with very coarse accuracy: Objects in a shop or warehouse are tagged with RFID chips and readers are placed at important points. If a reader detects the presence of an RFID tag, it can be derived that the tagged object is now changing from one area to another. Obviously, the accuracy depends on the number of readers and the reliability of the reading device. For person positioning, tags can be attached to, or integrated in, the clothes of users. RFID chips can also be implanted in animals as well as humans, however the latter is highly controversial ([Masters and Michael, 2005]).

A straightforward way to implement an egocentric positioning system is to deploy RFID tags into the environment and equip the agent with an RFID reader. The accuracy of such a system depends on the density of the RFID tags and the reliability of the reader. Manufacturers like Vorwerk[9] and Future-Shape[10] have developed carpets with integrated RFID tag-grids. However, the small communication distance of passive RFID means that the reader has to be close to the carpet, which limits this approach to robots, wheelchairs or shopping trolleys (see also Section 5.6), i.e. appliances where readers can be installed close to the floor.

Kiers et al.: ways4all In [Kiers et al., 2011] an indoor navigation system for visually impaired people, called ways4all, is described. The used positioning determina-

[9] http://corporate.vorwerk.com
[10] http://www.future-shape.com/

tion is based on passive RFID and is an onboard/egocentric system. Ways4all uses passive LF (134.2 kHz) RFID tags, which are placed at strategic spots in environments (entrances, intersections, barriers) that are already equipped with a so-called Tactile Guidance System (TGS). In principle, a TGS consists of specially formed tiles that include grooves or bumps, which can be felt through the use of a so-called long cane or white cane.

The IDs of all deployed RFID tags are stored into a database together with their coordinates. The RFID tags are organized into three different types: endpoint tags (ET) that mark possible destinations, intermediate tags (IT) that mark points along a route and virtual tags (VT) that mark possible destinations where no physical RFID tag is available. The RFID reader is either integrated into a long cane or white cane or attached via a clip to one shoe of the user. The reader is connected to the user's mobile device through Bluetooth. Since the TGS already provides 'rails' that help the users to follow specific routes, the ways4all system uses these rails to plan a route to a selected destination and uses read RFID tags as indicators if the user is correctly following the route and gives audible instructions at decision points.

The authors did not evaluate the position accuracy of their system, but tested the detection rate of the deployed RFID tags and the usability of the long-cane mounted reader versus the shoe mounted one. Although initial tests with slow walking speeds showed a detection rate of 80%, this rate dropped drastically to 33% when actual blind users where testing the system with their normal walking speed. The authors tried to improve the detection rate by forming tag arrays (TA) of up to four single tags, where each single tag in a TA is reporting for the same position. The TAs improved the detection rate to 40%. A user study was conducted at a public place in Vienna, with four blind men between 35 and 60 years old. All test persons could successfully install the system (including the RFID reader) and find their destinations. The authors conclude that they could improve the detection rate in their system by using or building passive RFID tags with a higher communication range.

Similar approaches, using passive RFID readers in long canes for visually impaired people, were reported in [Faria et al., 2010] and [E. D'Atri and et al., 2007].

Ni et al.: LANDMARC In ([Ni et al., 2004]) one of the first positioning systems using active RFID is described. The system, named LANDMARC, is offboard/exocentric and uses active readers and tags that operate at the uncommon frequency of 308 MHz. The communication range is specified with 150 feet (approximately 45 meters). The readers can operate in eight different communication ranges, where level 1 has the lowest range and 8 the highest. In a first test, the authors deployed nine readers in a test environment, where one reader was set to level 8 and placed in the middle. The other readers where placed on a circle around the middle reader, with lower levels. The idea was to locate an active RFID tag by checking which readers report its presence and finding the correct area, which is defined by the communication ranges and positions of each detecting reader. The attempt failed, because too many factors influenced the communication range of the RFID system. The actual LANDMARC approach was then to put additional reference tags into the environment, which should help to calibrate the system.

The positioning algorithm was in fact a fingerprinting approach, where the reference tags, whose coordinates are known, acted as reference points that could be calibrated at runtime, i.e. for each reference tag, vectors containing the reported signal strengths of all readers were continuously stored. The signal strengths were obtained by successive readings with different communication-distance settings. The same kinds of vectors were built for the tags with unknown positions and a k-nearest neighbor (see Section 2.5.3) method was used to determine the position of these tags.

The authors tested their approach with 4 readers and 16 reference tags. 8 tags where placed on known positions and measurements of this static setup were taken over a course of 48 hours. The collected data was then analyzed to determine the position accuracy of the system. It turned out that a 4-nearest neighbor provided the most stable results and that the influence of changing environmental conditions was low. With one reference tag per square meter, an average accuracy of 1-2 meters could be achieved. The authors also report that different tags provide different measurements, which they identify as the main problem of their approach.

3.1.6 Optical Positioning

Although light is also an electromagnetic radiation, it makes sense to distinguish optical systems from the radio-based methods above. First of all, the frequencies are much higher: infrared light (IR) is defined to be in the frequency range between 300 GHz and 400 Terra-Hertz (THz), visible light covers the range between 400 and 790 THz and ultra-violet (UV) from 790 THz to 30 Peta-Hertz (PHz). X-rays and gamma-rays range from 30 PHz to 30 Exa-Hertz (EHz) and 30 EHz to 300 EHz, respectively. Secondly, besides the use as a data-transmission medium, picture generating light sensitive sensors can be constructed, although UV light, X-ray and gamma-rays are usually not used for positioning systems, as they are highly energetic and are thus potentially dangerous.

3.1.6.1 Infrared Based

Infrared (IR) plays an intermediate role, as it can also be used for data transmission. Besides the already mentioned use in the IEEE 802.11 protocols (see Section 3.1.3), the IrDA (Infrared Data Association) protocol is a widely adopted standard. IrDA 1.0, also called SIR (Serial Infrared) was proposed in 1994 and allowed a data-rate up to 115.2 kBit/s. In its newest iteration, called VFIR (Very Fast Infrared), data-rates up to 16 MBit/s are possible ([Roth, 2005], page 109). Because infrared transceivers, sensors and senders are very cheap and where available in many mobile computing devices, early indoor positioning systems were based on this technology. As infrared light behaves like visible light, it is easily blocked by obstacles such as walls. This characteristic can be advantageous for indoor position systems, since room accuracy can be easily achieved. On the other hand, it can also be a disadvantage as an IR signal can unintentionally be blocked.

Want et al.: Active Badge The Active Badge system as described in [Want et al., 1992] was an early offboard/exocentric positioning system based on infrared light signals. A network of IR sensors is installed into a building. Users of the system have to wear an IR sender, called an active badge. This badge sends out an IR signal every 15 seconds with a 100-millisecond duration, which contains a unique identification (ID) code. IR sensors report sensed IDs to a centralized server, which in turn know the locations of each sensor and can thus position each badge. In [Want and Hopper, 1992] an extension to the original badge was introduced, called the Authenticated Badge. This badge can also receive information via infrared, which was used to implement a challenge-response method to prevent users from simply replaying recorded IR signals from other badges. Furthermore, the Authenticated Badge contains to two buttons, two LED indicators and a tone generator, which can be used for further user interaction. The Active and Authenticated Badge systems reached room-level accuracy.

Wahlster et al.: IRREAL IRREAL (InfraRed REsource Adaptive Localization) is an indoor positioning system, which was developed in the project REAL as part of the Collaborative Research Center 'Resource-adaptive Cognitive Processes' (SFB 378) and was funded by the German Research Foundation (DFG) ([Wahlster and Tack, 1997]). The used positioning method can be classified as onboard/egocentric, however the system is a clever combination between transmission of navigation information and positioning.

Special IrDA compliant IR senders were developed that could transmit information over a range of 20 meters. These senders are deployed in the environments, i.e. at the ceiling or walls, and are connected to a presentation server. Each sender not only transmits identification information about its own position, but also delivers presentation content that is viable in the current area, i.e. arrows that indicate the walking direction for a specific route or a time table for the nearest bus station. These presentations consist of different nodes, which can contain textual information or graphical representations. A specially designed protocol repeats important presentation nodes, i.e. nodes at the start of a presentation, more often than unimportant ones. In that way, the probability that the mobile device of a moving user has received the start of a presentation is increased. As the user stays longer in the instrumented environment, additional presentation information will eventually be received by the mobile device (cf. [Wahlster et al., 2001, Bartelmus, 2002, Baus, 2003]).

Thermal Based

Any object with a temperature above absolute zero emits electromagnetic waves. The wavelength and thus the frequency of that radiation depends on the temperature of the object. Objects within a temperature between 0°C to 70°C lie within the range of infrared light and can thus be picked up by infrared sensors. As indicated in Section 2.3.3, this effect can be used for passive positioning systems.

Hauschildt et al.: ThILo In [Hauschildt and Kirchhof, 2010, Kirchhof, 2011] such a Thermal Infrared Localization (ThILo) method is described. The authors use arrays of so-called thermopiles as sensors, which measure the difference between ambient radiation and object radiation. Each array has a resolution of 8 pixels and each pixel has a field of view of 6 degrees. A room is instrumented with two arrays at each corner. In theory, a person can be localized by using the direction of the pixels with the highest measured temperature difference in each corner and then performing a triangulation. In practice however, additional heat sources also influence the measurements.

The authors developed a semi-automatic system calibration, during which a human has to walk through the room and is advised by the system to stop at random points. The system takes repeated measurements during those stops and constructs a system of non-linear equations. The solution to the equation system is approximated with the Newton-Raphson method and results in position estimations for each stop.

In order to track multiple targets, a Probability Hypothesis Density (PHD) filter was implemented, which is basically an extension of particle filters (see Section 2.6.2). The system was evaluated in a 4.9 meters by 6.2 meters room without disturbing heat sources. One or two persons walked along predefined shapes (rectangles, diagonal) and the minimum and maximum distance to these tracks were measured. With one person in the room, the accuracy was between 9 centimeters and 26 centimeters. With two people it was between 12 and 68 centimeters. The authors conclude that background radiation and reflection are still issues to work on but the overall results so far are promising. Being a passive system, ThILo is an offboard/exocentric positioning system.

(a) An example QR-code

(b) A simple ARToolkit-marker

(c) An unobtrusive AR-marker by String Labs Ltd

(d) A 3D model of a shoe rendered over the detected marker

(e) Optical markers as used in motion capturing

Figure 3.3: Some examples of optical markers.

3.1.6.2 Camera Based

Cameras are used for surveillance purposes for decades. As they do not only provide positional information but also a plethora of other information, e.g. activity, gender, body language, they are perceived as a high breach in privacy. Although the automatic analysis of images is a very complex task, the sheer possibility to record images or video-clips for later manual or human-assisted evaluation is seen as being problematic. Nonetheless, surveillance cameras are often installed and accepted in public places, shops and high security areas, with the premise that recorded imagery is only analyzed with a reasonable ground for suspecting. However, the ability to automatically extract person identification, movement patterns and activities refuels the discussions about public surveillance cameras.

As indicated, this automatic data-extraction is a highly complex task and in positioning systems with cooperative users, this task can be simplified by the use of so-called optical markers. These markers are easy to detect and to identify by computers and can even store additional information. A common example is the QR code (Quick Response code), which basically is a two-dimensional barcode (see Figure 3.3a for an example). Such a QR code can contain up to 2,953 bytes plus error correction codes and can thus be used to store position information or URLs. Figure 3.3b shows a marker, which is used by the ARToolkit ([Kato and Billinghurst, 1999]). The ARToolkit can derive orientation and distance of such a marker in relation to a camera. Users with one or more markers attached to their clothing can thus be positioned very easily in an exocentric fashion.

If the size, position and orientation of such a marker in a room is known, the ARToolkit can be used to derive the distance and orientation of the camera relative to the marker. In [Piekarski et al., 2003] an egocentric positioning system is described, where a user equipped with rear, front and head cameras is positioned by using ARToolkit markers. They report a position accuracy between 10 and 20 centimeters, but the approach is hardly practical.

Figure 3.3c is also an optical marker, although it looks like a regular picture. To construct these types of markers, several methods exist. Machine-readable information can be incorporated into regular pictures in such a way that they are not attracting the attention of an observer (e.g. Mircosoft Tag[11] and DataGlyphs [Hecht, 1994]). Another method is to analyze pictures in preprocessing step to identify unique features that can be easily recognized (e.g. Bookmarkr [Henze and Boll, 2008] and MapSnapper [Hare et al., 2008]). The marker in Figure 3.3c is of the latter type and is tied to a specific iPhone application[12] that can identify the marker. By determining the marker's position and orientation relative to the phone-camera, the application renders a 3D model of a shoe at the markers position (see Figure 3.3d; the URL to the application is also stored in the QR code in Figure 3.3a).

For motion capturing, which is often applied to capture movements of an actor in order to simulate these movements with computer-generated graphics, simple light globes attached to a body-suit are often used as optical markers (see Figure 3.3e). In general, optical markers, which can be used as reference points, are called fiducials, fiducial points or fiducial markers.

Herranz et al. In [Herranz et al., 2011] an onboard/egocentric positioning system is described, which uses LED based markers deployed into the environment as senders and a camera worn by the user as sensor. The LED senders sequentially flash in the visual spectrum. The synchronization of this flashing is done via wireless connections and the positions of the LED senders are known to the positioning system. The worn camera detects these flashes and takes the brightest detected pixel as a starting point for a sub-pixel analysis process, which estimates pixel coordinates of each detected LED sender. The system tries to determine the position of the camera

[11] http://www.microsoft.com/tag/

[12] available at http://poweredbystring.com/showcase

through the use of a particle filter, where the system state represents the position, orientation and their derivatives, the prediction model contains a pinhole model of the camera and the pixel coordinates of detected flashes are used as measurements.

The authors evaluated their system using a circular motion, but instead of moving the camera, they rotated eight LED senders around the fixed camera. Although this is not the intended use of the system, the authors argue that this will provide them a better ground-truth and will reduce the effect of changing lighting conditions. The measurements were then evaluated in a simulation, to vary different parameters, like number of used senders, different camera frame-rates and amount of simulated measurement noise. The reported accuracy lies between 6 and 31 millimeters, where the best result was obtained with 300 frames-per-second and 8 senders. In a second experiment, the authors simulated movements in a room of 5×5 square-meters. The simulated accuracy was between 173 millimeters (with 10 senders) and 62 millimeters (with 30 senders). Although these results seem promising, real-life effects like obstruction of the senders and changing lighting conditions are not taken into account. Furthermore, the system was not tested on a mobile device.

Ruotsalainen et al. In [Ruotsalainen et al., 2011] an algorithm is described that uses the camera of a smart phone to deduce the heading direction of a user. The basic idea is to use standard algorithms for edge detection and to derive vanishing points from extracted lines that follow the detected edges. These vanishing points are used as features, whose camera coordinates can be determined. Through the use of a camera model, rotation of the camera relative to the vanishing points can be computed.

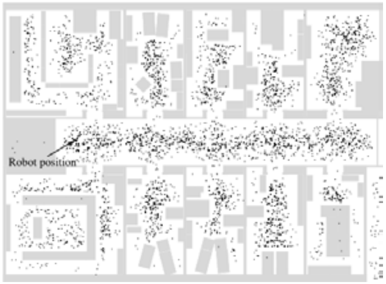
The authors tested the proposed algorithm by mounting a smart phone on a stationary platform that allowed orientation changes only on the x-axis. The phone was turned in 5-degree steps and pictures were taken at each step while completing a complete 360 degrees turn. The taken photographs were processed on a desktop PC using Matlab. The experiment was conducted in two different corridors. The measured mean error in the first corridor was 1.3 degrees and 1.8 degrees in the second. However, the authors admit that when the camera was turned into a scene with only a plane, a wall and an elevator door, the algorithm failed. The authors conclude that the proposed algorithm works well in corridors and outperforms the built-in compass, which showed a mean error of 18.1 degrees in the same test environment. The required computations could be performed with a 1 Hz rate on a desktop PC and still have to be adapted for smart phones.

Dettori In his diploma thesis [Dettori, 2008] developed an offboard/exocentric positioning system using a stereo-camera approach. Two off-the-shelf web-cams are deployed at a wall with a distance of about one meter (the exact distance can be specified in the system). The two resulting images are analyzed for corresponding parts, such that a correlation of both images can be computed. Through camera models, the camera-based coordinates of each pixel can be translated into a world-coordinate system and the distance from the camera to different object can be approximated by triangulation. Movements can be detected by an analysis of the optical flow, i.e. the observation of differences in successive images. The system assumes that only one person is in the room and matches the observed optical flow with a coarse body model, consisting of a torso and head. For the actual position determination, an extended Kalman filter is used, where the position state contains the world-coordinates of the body model and its rotation as well as its rotation velocity and movement velocity. The movement velocity can be derived from the optical flow. To approximate the rotation and rotation velocity of a body, which are hard to extract from images, the variance of the observed shoulder-length was used. Since this observation can lead to two solutions (if the observed shoulder length changes, it could be a rotation to the left or right), a second state vector was constructed, which acted as second hypothesis. The state vector with the highest mean probability was assumed to be the correct one. First tests showed an accuracy of 5 to 10 centimeters in a 5×6 square-meter room.

3.1.6.3 Laser-Range Positioning

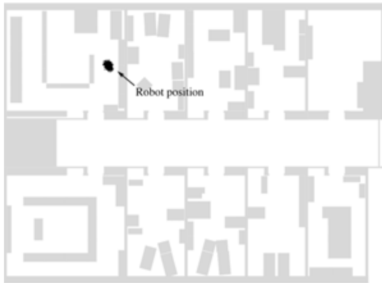
A laser-range scanner is an example for a sender/sensor combination as described in Section 2.3.2.3. Typically such a device sends out a short laser-pulse in a specific direction and measures the time of flight until the reflection of the pulse returns. A laser-range scanner thus can derive the distance to the next reflecting object in a specific direction. By using several lasers aimed at different angles or by rotating the laser, a laser-range scanner can determine a two- or even three-dimensional distance-map of its surroundings. Systems that follow this approach are called LIDAR, for LIght Detection And Ranging, or more specifically LADAR, for LAser Detection and Ranging.

In robotic applications, laser-range scanners are often used to detect obstacles or for determining the robot's position. The latter can be achieved with the help of a stored depth-map of a level in a building. A robot, which is equipped with a laser-range scanner, takes distance measurements from its current position and tries to find positions on the map that coincide with the measurements. With the help of a particle filter (see Section 2.6.2), these hypothetical positions are constantly tested as the robot moves along, eventually leading to one position. Figure 3.4 shows an example of such a laser-range based robot-positioning: (a) shows the initial state, where the particles are uniformly distributed over the depth map. (b) shows the state after a first



(a) Initial situation

(b) After first set of measurements



(c) After several sets of measurements

Figure 3.4: Robot positioning with a laser-range scanner ([Fox et al., 2001]).

set of distance measurements has been taken. Several hypothetical positions are left and two larger particle clouds indicate that the robot cannot yet decide which room it currently enters. (c) after several sets of measurements (while moving), only one particle cloud is left (cf. [Fox et al., 2001]).

The same principle is theoretically possible for personal positioning, however carrying a laser-range scanner is rather inconvenient and the non-steady height of the scanner while walking adds to the inaccuracies of the approach. As it was the case with passive RFID readers, laser-range scanners can be used to instrument appliances like wheelchairs, shopping trolleys or walking aids.

Röfer et al.: iWalker & Rolland At the Bremen Ambient Assisted Living Lab (BAALL), which is located at the German Research Center for Artificial Intelligence (DFKI) in Bremen, two appliances were instrumented with laser-range scanners: a walking aid, named iWalker, and an electric-powered wheelchair, named Rolland. The iWalker is based on an off-the-shelf walking aid (depicted in Figure 3.5a), which was upgraded with a laser-range scanner, electric brakes and wheel encoders ([Röfer et al., 2009a]). Rolland (shown in Figure 3.5b) is based on a commercially available, electric-powered wheelchair and was additionally equipped with

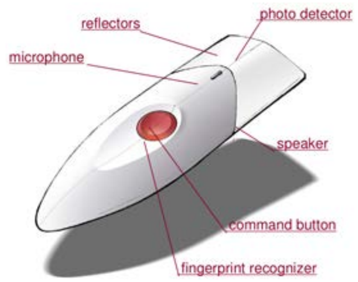
(a) iWalker (b) Rolland

Figure 3.5: Two appliances equipped with laser-range scanners where developed in BAALL at DFKI in Bremen.

two laser-range scanners: one at the front and one at the back of the chair. For both appliances, the laser-range scanners are mainly used to avoid obstacles on the way. This is achieved by building so-called local obstacle maps or occupancy grids. Such a map is quadratic array of cells, in which probabilities are stored that express the certainty of a possible obstacle. These maps are constantly updated by the laser-range scanners.

In the case of iWalker, which is manually pushed by a user, the detection of a nearby obstacle causes the walker to slightly brake one wheel. This braking automatically leads to a change of direction and helps the user to avoid the obstacle while giving them a tactile feedback at the same time. In the case of Rolland, the same principle is used, although the wheelchair is not manually powered but moved via an electric motor and can be steered by a joystick or, as described in [Röfer et al., 2009b], by a proportional head-joystick. Rolland assists the user in controlling the wheelchair by constantly monitoring for obstacles. If an obstacle is detected that lies on the predicted trajectory, Rolland adjusts the user given controls to help avoiding the obstacle. If the current control input indicates that the user is already trying to avoid the obstacle in a specific direction, Rolland will reinforce the current steering command.

Nakashima et al.: CoBIT The CyberAssist project, which was conducted at the Cyber Assist Research Center at the National Institute of Advanced Industrial Science and Technology (AIST) in Japan, focused on human-centered information systems ([Nakashima and Hasida, 2010]). During the project several iterations of a battery-less user device where developed, which should provide the user with location based information.



(a) Conceptual draft of MyButton.

(b) CoBIT

(c) Aimulet

Figure 3.6: In the CyberAssist project several user devices were developed to enable human-centered information systems ([Nakashima, 2007]).

Figure 3.6a shows a draft of such a device, called MyButton, which was equipped with reflectors, a photo detector, a speaker and a microphone as well as with a command button and a fingerprint recognizer. This device, and the devices that followed, should be worn as earpieces by the users. In order to provide information to the users and to determine their position (in an offboard/exocentric fashion), base stations called i-lidar were deployed in the environment. A first prototype of these base stations contained an infrared laser that could be steered by mirrors. When the laser was aimed at the reflector of the MyButton, the direction and distance to the base station and thus the position could be derived. The authors report, that they could determine the position within an accuracy of millimeters, but four seconds of scanning time were needed to find a MyButton.

In order to improve the time-resolution of the positioning, the i-lidar stations were upgraded with infrared cameras, with which the direction of a MyButton could be detected. The infrared laser could then directly aim at the reflector of a MyButton and determine the distance. With this approach a positioning accuracy within 1 centimeter was achieved ([Itoh et al., 2003]). Although the accuracy is extremely high, the authors admit that the needed laser costs ten million Yen (about 100,000 Euro) and not even mass-production will bring the costs below one million Yen ([Nakashima and Hasida, 2010]). The infrared cameras were not only used for determining the direction of a MyButton, they could also receive data from the device.

One of the reflectors of MyButton could be electronically switched between translucency and reflection. Through this mechanism, information could be transmitted by manipulating the reflections of an incoming infrared light-beam. Information was transmitted to a MyButton by sending out an infrared light-beam, onto which information was modulated by changing the intensity of the beam.

Figure 3.6b shows the first integrated version of the MyButton concept, which was called CoBIT, as abbreviation for Compact Battery-less Information Terminal. CoBIT contained a solar cell, which simultaneously provided the needed energy and was capable of decoding the intensity-modulations of an incoming light beam ([Nakamura et al., 2003]).

A low-cost and ecological version of CoBIT is shown in Figure 3.6c. It is called Aimulet and its housing is made out of bamboo. Aimulets were given away for free at the Expo 2005 in Japan, where the system was successfully demonstrated, although positioning was abandoned because of privacy considerations ([Nakashima, 2007]).

3.1.6.4 Optical Positioning in Gaming Consoles

Although cameras at home are perceived as a privacy violation, they seem to be accepted in conjunction with gaming consoles. For Sony's PlayStation 2[13] an accessory was available, which was called EyeToy. EyeToy was basically an USB webcam and games mostly used simple algorithms based on the difference of subsequential image frames to detect motion and coarse user positions. With the start of Nintendo's Wii[14] console in 2006, a change from the traditional gamepads to a motion-based interface took place. Nintendo's controller, called the WiiMote, includes an infrared camera that tracks blobs created by infrared LEDs, which are attached to the user's TV screen. Since the original distance of the LEDs between each other and their orientations is known, a coarse distance and orientation approximation of the user-held infrared camera can be accomplished. In addition the WiiMote also contains accelerometers and can be further equipped with a three-axis gyroscope, which allows for a more fine-grained estimation of the user's movements. Since the WiiMote does most of the calculations itself, it can be classified as an onboard/egocentric system, although its main purpose is not really positioning, but movement detection.

In 2009, Sony introduced its own motion control for the PlayStation 3, called PlayStation Move. The Move controller is equipped with a glowing orb at the top, which is illuminated by RGB LEDs. This globe acts as an optical marker, which is tracked by a camera, the PlayStation Eye, which is directly connected to the gaming console. With the help of the camera, the gaming console determines which colors

[13] http://us.playstation.com/ps2/
[14] http://wii.nintendo.com

are prominent in the current view and chooses a specific color for the glowing orb, which helps to detect and track the Move controller. Additionally, the Move controller is equipped with accelerometers, angular rate sensors and a magnetometer. The PlayStation Move can be classified as an offboard/exocentric system and as it is the case with the WiiMote, its main purpose is to determine movement.

Microsoft's solution for motion control for the XBox 360[15] differs from the other approaches in that it does not need any user-held devices. The system was released in 2010 and consists of a device, called Kinect camera, which is placed above or below the TV screen. The Kinect camera contains an RGB camera as well as an infrared-laser projector and a matching infrared-camera. The infrared-laser projects a regular pattern of dots into the environment, which are in turn detected by the infrared camera. Since the system has a model of the pattern, it can infer depth information by analyzing the pattern-distortions observed in the image of the infrared camera. The acquired depth information can be combined with the image of the RGB camera and further image processing can be used to derive body postures and positions. In addition, the Kinect device also has a microphone array and can be automatically tilted through electrical motors.

Because all three devices contain a broad array of sensors and are comparably cheap, they are often used in hobbyist and scientific projects.

3.1.7 Terrestrial Radio & TV Broadcast Based

Terrestrial Radio Broadcast infrastructures, as used for radio and television, can also be used for opportunistic positioning systems.

Rabinowitz et al. In [Rabinowitz and Spilker, 2005], a system is described that uses the embedded synchronization signals of digital television (DTV) broadcasts. The system was designed for DTV signals as specified by the American Television Standard Committee (ATSC), but the authors claim that other DTV standards such as Digital Video Broadcast (DVB) in Europe or Integrated Services Digital Broadcast (ISDB) in Japan could be used for accurate positioning. The proposed positioning system needs so-called Monitor Units at known positions, which monitor TV station timing-offsets. These timing-offsets are needed for the position determination. According to the authors, the positioning can either be onboard, where the Monitor Units provide the offset data to the mobile device, much like in DGPS, or offboard on the Monitor Units, where the mobile device has to send its measurements to the Monitor Units. A third alternative would be to alter the TV transmission protocols

[15]http://www.xbox.com/

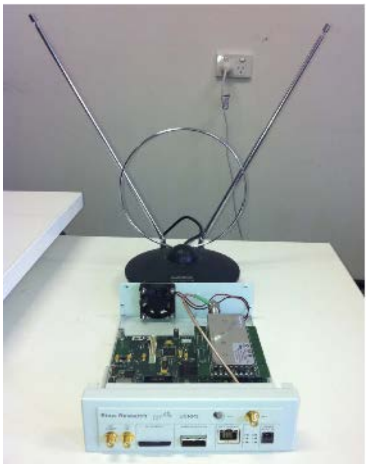


Figure 3.7: The sensor equipment used for FM-radio-based positioning ([Moghtadaiee et al., 2011]).

in such a way, that they incorporate the needed clock offsets. The positioning algorithm uses pseudo-range measurements to at least three TV transmitters with known positions and applies a trilateration approach. The authors tested the accuracy of their system in different environments: in an outdoor park a mean position-error of 3.2 meters was measured. In several indoor scenarios, the measured mean-error was between 10.3 meters and 23.3 meters. In a parking garage, the measured mean error was 12.3 meters.

Moghtadaiee et al. In [Moghtadaiee et al., 2011], a positioning approach is described, which uses signal-strength fingerprinting on FM (Frequency Modulated) radio-broadcast signals. In this system, fingerprints consist of a vector of signal strengths for a number of different FM radio channels. An evaluation was performed in the fourth floor of a multistory building, which consisted of seven rooms on 11×23 square-meters. A total of 150 reference points was taken during the offline phase, where 120 measurements within 12 seconds were taken for each reference point. The fingerprints consisted of 17 signal-strength values for 17 different FM radio channels, covering a frequency range between 88 MHz und 108 MHz. Figure 3.7 shows the used sensor equipment, consisting of a Linux based radio receiver and a ‘rabbit ear’ antenna. 28 fixed test-points where defined in the test bed, at which the accuracy of the onboard/egocentric positioning system was tested, using three different approaches: simple nearest neighbor (NN), k-nearest neighbor (kNN) and k-weighted nearest neighbor (kWNN). With NN a mean distance error of 3.29 meters was achieved. The kNN approach reduced the error to 3.09 meters and the kWNN performed best with 2.96 meters.

3.1.8 Magnetic Based

Magnetic compasses use the Earth's magnetic field to determine magnetic north. Although a compass does not directly determine its position, it can be seen as an opportunistic heading device. Modern smart phones are often equipped with magnetometers, which are usually used as an electronic compass to determine the orientation of the device or the current heading of its user. The magnetometer can also be used for position determination, e.g. by using the Earth's magnetic field or by instrumenting the environment with artificial magnetic fields.

Storms et al. An example of the first method is given in [Storms et al., 2010], which relies on the fact that the Earth's magnetic field varies depending on the current position. In essence, the same method is used as was described in Section 3.1.6.3 for robot positioning with laser-range scanners, but instead of a depth map, a map containing magnetic-field measurements is used, and instead of laser-range scanner, three-axis magnetic field sensors are used. A particle filter combines inertial measurements (see Section 3.2) with magnetic-field measurements.

The authors tested their approach in two connected, narrow corridors in their lab. For the creation of the magnetic-field map and the positioning experiment itself, the magnetic-field sensors were installed onto a non-ferrous vehicle, which was pushed trough the corridors. To determine the position accuracy, the vehicle was pushed along a predefined trajectory along the right side of both corridors. The authors report a maximum error distance of 60 centimeters and 'less than 0.2 meters for the majority of the trajectory'. Since the generation of a magnetic-field map is a complex task, the authors conducted a second, simulated experiment, in which a first vehicle drives along a random path while collecting magnetic-field data, and a second vehicle tries to follow the trajectory of the first one as closely as possible using the collected magnetic-field data. The reported accuracy lie is 0.3 meters, however the second vehicle seemed to have problems following the turn of the first vehicle, when driving from one corridor into the other. The authors conclude that positioning with magnetic-field sensors is promising, but admit that the sensors are highly sensitive to changing conditions, for example the way the connector cable is attached to the sensor, and that the stability of the magnetic field over time has to be further investigated.

Blankenbach et al. In [Blankenbach et al., 2011], an outline for an instrumented positioning system using artificial magnetic fields is given. The proposed system is called MILPS, which stands for Magnetic Indoor Local Positioning System. The basic idea is to deploy electrical coils in a building, which are activated sequentially. A mobile device can then derive its own position by determining the distance to three coils and performing trilateration. The authors conducted some initial tests with a direct current (DC) powered magnetic coil, with a diameter of 50 centimeters and 140 turns of wire. The coil was powered with 15 Ampere and measurements were taken with a three-axis magnetometer. To overcome interference problems, the coil's current-direction was reversed in defined time intervals, thus interferences with a lower frequency than the field-changing frequency could be filtered out. Measurements showed that the magnetic field of the coil could be detected at distances up to 16 meters, with up to four walls between the coil and the sensor. The distance between the coil and the sensor was approximated using the measured field strength. At distances under 6 meters, the accuracy of the approximated distance was between 4 and 7 centimeters. The authors conclude, that artificial magnetic fields provide excellent characteristics for penetrating objects, but further experiments and improvements have to be made to approximate distances greater than 10 meters away from a coil.

3.1.9 Ultra-Wideband (UWB) Based

Electromagnetic based communications protocols, as discussed so far, use a narrow frequency-bandwidth, usually in the range of a few Megahertz. In narrowband communication systems, a carrier-frequency is chosen onto which information is modulated. Ultra-WideBand (UWB) differs from this approach, in that communication is accomplished by sending carrierless, short-duration signal pulses, which last in the range of pico- or nanoseconds. The ratio of the time that a signal is present to the total transmission time is very low, thus the overall transmission power is very low in comparison to narrowband communication. By definition of the Federal Communications Commission (FCC), UWB signals must have a frequency-bandwidth of more than 500 MHz (in comparison, WiFi has a frequency-bandwidth of 22 MHz). Because of the high frequency-bandwidth, a very high data-rate can be achieved. However, the FCC has limited the allowed transmission power and thus only ranges of up to 10 meters can be achieved. The IEEE 802.15.3a working group tried to establish an UWB standard for Wireless Personal Area Networks (WPAN), but the group was dissolved in 2006, due to disagreements over two technology proposals. However, another working group has specified IEEE 802.15.4a[16], which specifies alternative physical layers for 802.15.4, one of which is based on UWB (cf. [Nekoogar, 2011], pp. 2–24).

For positioning applications, the high frequency-bandwidth is supposed to grant a high signal-penetration through different materials, and thus tackles the problem of

[16] http://www.ieee802.org/15/pub/TG4a.html

signal fading. As a generic electromagnetic wave based communication, the same positioning approaches can be used as with cell phones or WiFi, e.g. proximity, AOA, TOA, TDOA, signal strength and RSS fingerprinting.

Ubisense Ubisense[17] is a commercially available offboard/exocentric positioning system, which is based on UWB. Objects or persons are equipped with a proprietary tag, called the Ubitag, which contains a UWB sender. UWB sensors, called Ubisensors, are installed in the environment at known positions and are connected to each other via Ethernet. The Ubisensors are organized in cells of four to seven sensors, where one cell acts as master. Both, Ubitags and Ubisensors, are also capable of exchanging data via conventional narrow-band radio. The master sensors synchronize the measurements of several Ubitags, i.e. they provide a timing-schedule in which each Ubitag is assigned a specific time-slot. In these time-slots, Ubitags send their ID via narrow-band and emit UWB pulse sequence. The tags' positions are determined by a combination of TOA and TDOA measurements and thus only two Ubisensor measurements are needed to obtain a position. The manufacturer claims an accuracy of 15 centimeters for the system ([Steggles and Gschwind, 2005]).

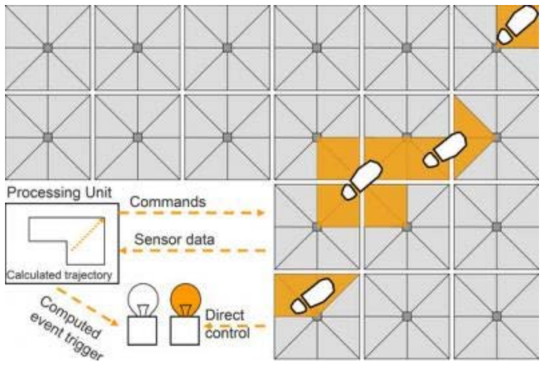
Stephan et al.: Real World Evaluation [Stephan et al., 2009] tested the accuracy of Ubisense at the SmartFactory^{KL}, which is a multi-vendor research, development and demonstration center for industrial information and communication technology located in Kaiserslautern, Germany. The SmartFactory^{KL} contains a complete production facility, which includes many metal structures, piping and glass vessels besides the heavy machinery. The authors tested the system in two different conditions: an optimal condition, in which no obstacles and no radio interferences were present, and a realistic scenario, which was conducted on the shop-floor of the factory. Reference points where determined with an accuracy of ± 2 millimeters in order to setup the system and to have references to test the positions determined by Ubisense. Under optimal conditions, Ubisense's position accuracy was in the range between 12.8 and 24.4 centimeters, and thus approximately in the range that was specified by the manufacturer. Under realistic conditions however, the position accuracy fluctuated between 35.1 centimeters and 124.3 centimeters.

3.1.10 Capacitance Based

Capacity based systems use an effect that is best known from a musical instrument called the Theremin (or aetherophone), which is played without touching the instru-

[17] http://www.ubisense.net
[18] http://kvraudio.com/

(a) Léon Theremin playing his instrument[18]



(b) SenseFloor can detect single footsteps[19].

Figure 3.8: Capacitance based interfaces as used in the beginning of the 20th century (a) and the beginning of the 21st century (b).

ment itself. The Theremin was invented by the Russian Professor Léon Theremin in the first half of the 20th century and basically consists of a variable capacitor, where an antenna is used as one plate of the capacitor while the other is provided by one hand of the player. Figure 3.8a shows Léon Theremin playing his own instrument. The proximity of the player's hand (or whole body), changes the capacitor's capacitance, which is the case of the Theremin is used to change the frequency of an oscillator circuit. The resulting frequency directly correlates to the distance of the user's nearest body part, i.e. the smaller the distance, the higher the frequency. However, the user has to be in close proximity (in the range of 1 meter) to the antenna.

Endres et al. [Endres et al., 2011] describe an initial study on how a capacity-based circuit can be used to detect and classify one-finger gestures in an automotive context. In this study a modified Theremin, called the Geremin, was used and the produced sounds where sampled with an analog-to-digital converter before further signal processing was applied, i.e. the determination of the frequency and the frequency-changes of subsequent measurements. Although only one antenna was used in the experiment, simple gestures like finger up, down, left, right and drawing rectangles into the air could be recognized with $\approx$70% accuracy.

SenseFloor The SenseFloor[19], manufactured by Future-Shape[10] (see also Section 3.1.5), uses the same principle for position determination. To cope with the

[19]http://www.future-shape.com/en/technologies/23/sensfloor

problem of the needed proximity to the antenna, they designed a textile underlay that contains 32 capacitance-based proximity sensors per square meter. Each sensor communicates its measurements to a control unit, which can then determine where people are standing or lying. Information about walking direction and moving velocity can be derived as well as how many people are currently present on the instrumented floor. The underlay can be installed beneath PVC, laminate or carpets and is mainly intended for the use in Ambient Assisted Living.

Touchscreens, such as used in most modern smart phones, can also be built using capacitance measurements, although here the direct touch of the user is required and usually also desired. The PointScreen, developed by Fraunhofer Institute for Intelligent Analysis and Information Systems[20] (IAIS), uses the Theremin principle and allows users to interact with a screen without touching it.

3.1.11 Wireless Sensor Networks (WSN)

Wireless Sensor Networks (WSN) consist of a number of self-powered nodes, where each node contains one or more sensors as well as data-processing and communication capabilities. The nodes can build ad-hoc networks over wireless communication and thus sensor information can be exchanged. Because of the ability of building ad-hoc networks, WSNs can be deployed on demand by scattering sensor nodes in an area of interest. This makes them valuable for military operations as well as for environmental and health applications ([Pottie and Kaiser, 2000], [Akyildiz et al., 2002]). A number of different standards and commercial sensor nodes are available. Among the most well known is the ZigBee[21] standard, which is based on IEEE 802.15.4 (see also Section 3.1.9). For wireless communication the ZigBee standard uses narrow-band transmission on the 2.4 GHz as well as the 915 MHz (Americas) and 868 MHz (Europe) frequency bands. Compared to other wireless communication protocols, the data rate is considerably low and ranges from 20 Kbits/s to 250 Kbits/s. The transmission distance between two nodes ranges from 10 meters to 1,600 meters and depends on the power output of the nodes as well as on environmental conditions.

Kuflik et al. In the PIL project (Personal experience with active cultural heritage IsraeL), an integrated framework for multimedia museum-guides was developed, which takes visitor positions as well as group interactions into account ([Kuflik et al., 2011b]). An offboard/egocentric positioning system was developed and deployed in the Hecht museum in Haifa, Israel. A WSN based approach was

[20] http://www.iais.fraunhofer.de/
[21] http://www.zigbee.org/

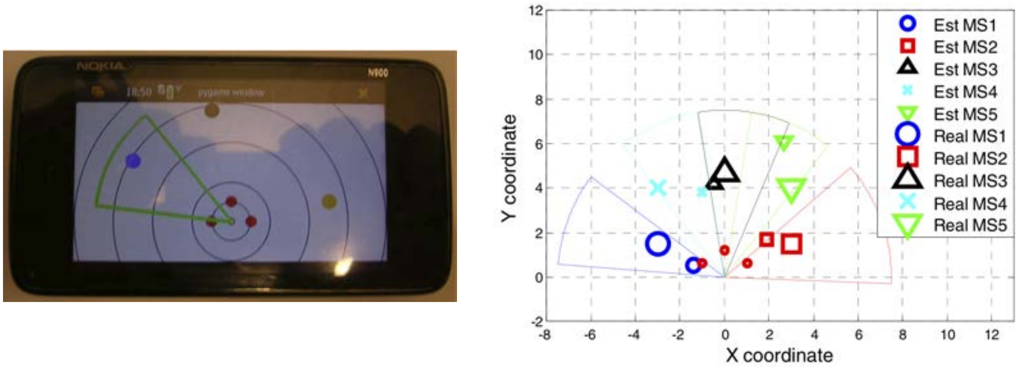
(a) A PIL blind (b) A PIL beacon (c) A PIL bridge node

Figure 3.9: The WSN in the Hecht museum consists of blinds (a), stationary beacons (b) and bridge nodes (c) ([Dim et al., 2011])

chosen, where the nodes consist of stationary beacon nodes and mobile nodes, which are called blinds and are worn by the visitors. A third kind of nodes is used to bridge the wireless communication of the nodes into a TCP network. The nodes are shown in Figure 3.9b. They were manufactured by the Italian company Tretec[22] and operate on the 2.4 GHz frequency band. The blinds can detect the presence of beacon nodes as well as the presence of other blinds, which is important to derive group interactions. Furthermore, the blinds contain accelerometers as well as a magnetometer, to determine a visitor's orientation, and can measure voice level and activity. Each blind sends its measurements via the bridge nodes to a centralized server, where the positioning estimation is accomplished by analyzing the sensed stationary beacons, which are placed at the entrance and exits as well as on selected exhibits. The sending range of each beacon was adjusted manually to minimize interference. The determined position (and group information) is sent back to the visitor's mobile device, where a multimedia museum-guide is running, which adapts its presentation according to the visitor's position. In locations where two exhibits were too close to each other to be discriminated by the system, visitors were presented with a choosing dialog on their mobile electronic museum guide. According to the authors, a positioning accuracy of 1.5 to 2 meters could be achieved, which in this context means that a visitor could be detected as being interested in an instrumented exhibit, if they are in the range of 1.5 to 2 meters of that exhibit ([Kuflik et al., 2011a], [Dim et al., 2011]).

Rosa et al. In [Rosa et al., 2011] a positioning method is proposed, in which WiFi enabled devices, such as mobile phones, laptops or desktop PCs, act as sensor nodes. The derived position is relative to all other sensed devices, i.e. the position is represented as a coordinate system in which the user is assumed to be at the origin and all other sensed devices are shown at their estimated position relative to the user. A screenshot of the running application is depicted in Figure 3.10a, here three devices

[22]http://www.3tec.it/



(a) A screenshot of the implementation on a Nokia N900

(b) A realtive-position map showing the real positions as well ([Rosa et al., 2011])

Figure 3.10: Relative positioning using WiFi ad-hoc connections ([Rosa et al., 2011]).

were detected an positioned. Figure 3.10b shows an evaluation for 5 devices, where also the real position for each device is depicted. The system works by taking signal strength measurements for each detected WiFi device. As the signal strength only gives an indication of the distance to the detected device, the application asks users to perform three scans at three different positions: one to their left, one in front and one to their right. In a calibration process, an empirical path-loss model was created, which is used to approximate the distance to each sensed device based on the received signal strength. The position of each device is then estimated by trilateration. The authors evaluated their system using 5 detectable devices in a classroom scenario. The position accuracy highly depended on the maximum distance to the measuring device. Results were best in a range up to 5 meters, where an accuracy of a few meters could be achieved (no exact numbers are given in the paper). The authors also report, that the orientation of the measuring device as well as the hand-grip of the user have a high impact on the measured signal strength and thus on the accuracy of their system.

3.1.12 Sound Based

Sound is a mechanical wave that is transmitted in a medium, like gas, liquid or a solid. Human perceptible sound lies in the range of 20 Hz to 20 kHz. Sound with frequencies higher than 20 kHz is called ultrasound, sound below 20 Hz is called infrasound. In general, the same principles as for electromagnetic waves can be applied, i.e. signal strength can be measured as sound pressure level, angle of arrival of sound waves can be determined and sound traversal times can be measured. The speed of sound through air is approximately 1,236 km/h, which is very low in comparison to the speed of light (exactly 299,792.458 km/h in vacuum). Due to this low speed, TOA measurements, even on low distances, can be made with lower accuracy clocks than in the case of radio signals. An easy way to determine the distance d between a sender and a receiver is by simultaneously sending a sound pulse and an electromagnetic pulse, and measuring the time difference Δt between receiving the electromagnetic pulse and receiving the sound pulse. The distance to the signal source can then be determined by

$$d = \frac{\Delta t \, c \, v_s}{c - v_s} \tag{3.1}$$

where c is the speed of light and v_s is the speed of sound. However, the speed of sound highly depends on the current temperature of air, which has to be taken into account for precise measurements. In practice, the calculation if often simplified by assuming that the electromagnetic pulse is received instantaneously and thus the measured time difference is treated as the direct TOA of the sound pulse. This method is also called the thunderstorm principle, as it can be applied to approximate the distance of an observer to a thunderstorm by counting the seconds between perceiving lightning and perceiving thunder.

3.1.12.1 Ultrasound Based

Machine-readable information can be transmitted using sound waves by the same methods that are used for electromagnetic waves, e.g. by modulating on a carrier frequency or sending short pulses. Examples of such sound based, machine readable transmissions in the audible frequency range are fax machine transmissions over phone lines or MOdulator/DEModulators (MoDem), which where commonly used before the introduction of ISDN or DSL to connect to computer networks. Because the sound of such a transmission is rather unpleasant, ultrasound frequencies are preferred when applicable.

Ward et al.: UltraBat One of the first ultrasound based positioning systems is described in [Ward et al., 1997, Ward, 1998, Harter et al., 1999, Addlesee et al., 2001] and ultimately led to an offboard/exocentric system called UltraBat. Users of the system have to wear tags, the so-called Bats, which can send out ultrasound signals and can receive radio signals. Ultrasound sensors are installed in a dense grid on the ceiling of an indoor environment, which can detect signals sent by an ultrasound tag. The sensors are wired to a server, which collects the sensor measurements and determines the position of each Bat via multilateration. The measurements are taken in rounds and each measurement process is started by the server, which synchronously sends a reset signal to the sensors and emits a radio signal. A Bat that receives the radio signal immediately sends a short ultrasound pulse. The sensors measure the time difference between the reset signal and the incoming of the ultrasound pulse. The server then calculates the distance of the Bat to each sensor using the thunderstorm principle, as the radio signal and the ultrasound pulse are not sent from the same location. According to the authors, UltraBat achieves an accuracy in the range of 10 centimeters, but a high number of wired sensors has to be installed to achieve this precision and the position of each sensor has to be accurately determined (cf. [Baus, 2003]).

Baunach et al.: SNOWBAT [Baunach et al., 2007] present a positioning system called SNOW BAT, which overcomes some of the high deployment efforts of the UltraBat system by using a WSN (see Section 3.1.11) instead of a wired node-network. The WSN consists of mobile nodes containing ultrasound senders and static nodes containing ultrasound sensors. Both kinds of nodes are also equipped with temperature sensors, which are used to approximate the correct speed of sound given the measured ambient temperature. The nodes communicate with each other through radio transmissions using a protocol called SmartNet.

As it was the case with UltraBat, the static nodes are deployed into the environment, knowing their own position. If a mobile node wants to find out about its own position, it broadcasts a radio message containing an ID and a time period Δt, which specifies a time delay after which the node will send a series of ultrasonic pulses. Static nodes that receive the radio message start a timer after the specified time delay Δt and wait for the incoming ultrasound pulses. If a node receives the ultrasonic pulses, it averages its measured TOA values over all incoming pulses and sends the computed value back to the mobile node. Static nodes that do not receive ultrasonic pulses stop measuring after a specified timeout-period.

The mobile node collects the incoming TOA measurements and determines its own position using multilateration. Since the position determination is calculated by the mobile node itself with the help of sensors in the environment, SNOW BAT classifies as an onboard/exocentric system. In order to calibrate the system, the static nodes are installed in the environment and a mobile node is brought to a number of reference points, which positions have to be exactly known. The static nodes then calculate their own positions, using the reversed approach that is later used for the positioning of mobile nodes. The authors claim a position accuracy of 15 millimeters, although they do not specify how they evaluated their system nor how many static nodes were used.

In [Runge et al., 2011], two calibration methods for SNOW BAT are presented. The first method is called the Explorer algorithm, which starts with three already calibrated static nodes, i.e. the exact position of each calibrated node is known and stored in the nodes. A mobile node is then brought into a position in which it can reach the three calibrated nodes and can thus determine its own position. The mobile node broadcasts its determined position and each uncalibrated node stores this position along with the distance measurement to that position. The mobile node is then moved along a predefined path, while keeping determining and broadcasting its own positions. Each uncalibrated node that has received at least three non-collinear positions can determine its own position, and once calibrated helps to determine the next position of the mobile nodes. The authors remark that with this approach, calibration errors sum up over time, as an initial error will lead to imprecise calibrations of new nodes, which will in turn influence the position determination for the next nodes. The explorer algorithm can thus only be used in small areas, in order to minimize the error propagation.

The second calibration method, called Distribute & Erase, can be applied on any number of nodes and in areas of any size. The area is roughly divided into large cells and in each static node the area in which it is located is stored. Again, three static nodes have to be exactly calibrated at the start of the algorithm. As in the Explorer algorithm, a mobile node uses these three calibrated nodes to determine and broadcast its own position but with the Distribute& Erase approach, the mobile node can be moved arbitrarily. Uncalibrated nodes use the received positions of the mobile node and their own distance measurements to iteratively adjust their own position. At the start, each uncalibrated node assumes to be in the center of its cell. With each subsequent measurement it adjusts its own position in order to minimize the distance error between the current measured distance and the reported position of the mobile node. If the distance error falls below a threshold, the node is considered calibrated. The moving node only takes calibrated nodes into consideration for its position determination. The authors tested their calibration approach in an area with 11×11 deployed static nodes (the size of the area is not given in the paper). With 1800 measurement steps, a calibration error less than one millimeter could be achieved, where an additional filter was used to reject unreliable distance measurements. However, taking these 1800 measurements takes 15 hours in which the moving node's position has to be changed every 10 seconds. The authors propose to use several moving nodes simultaneously to reduce that time. An evaluation of the position accuracy of the calibrated system is not given.

3.1.12.2 Speaker Positioning

Besides using ultrasound senders, audio based positioning can also be performed by using natural, audible sound signals, e.g. sounds produced by working machines, or spoken words. One way to accomplish this, is to use microphone arrays, which

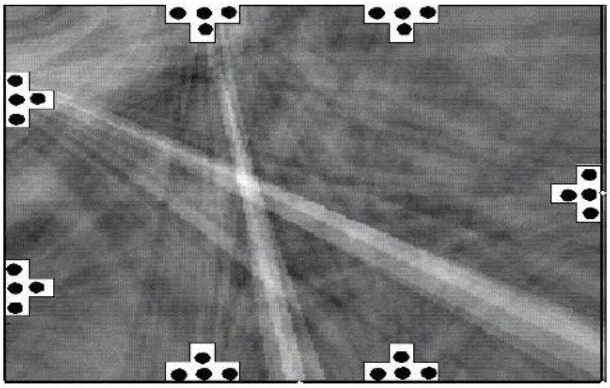


Figure 3.11: An example of a Global Coherence Field obtained by microphone arrays ([Brunelli et al., 2007]).

are deployed in the environment. Such microphone arrays can be used to derive the angle of arrival (AOA, see Section 2.5.2.1) of incoming sound signals, by analyzing the delay with which a sound signal arrives at each microphone of the array.

The reversed approach, i.e. applying different delays on the measured signals of each microphones before summing them, is called beamforming. Beamforming amplifies sounds coming from a specific direction, while attenuating sounds arriving from other directions. Using beamforming, a room can be 'scanned' for possible positions of speakers by subdividing the room in possible speaker positions through a grid and directing the beam at each grid-point. Note, that the directing of the beam is not mechanical, but purely computational, i.e. during a time-slice t the measurements of each microphone are stored and the result for each beam can then be computed from these measurements. The computed power of each beam's output signal for each point gives an indication of where a speaker (or other sound source) is positioned. The obtained power output measurements for each point of the grid can be represented trough a power field or through a Global Coherence Field (GCF). In order to construct a GCF, a coherence measure for each pair of microphones is computed for each point of the grid. The obtained coherence measures for each point are summed up and normalized by the number of used microphones (cf. [Omologo et al., 1998]). Figure 3.11 shows an example of such a CGF. The used microphone arrays are shown at the edge of the image. Single beams can be seen as bright lines and the brightest area in the picture represents the most plausible position of an active sound source. The image is taken from [Brunelli et al., 2007], which is described in more detail in Section 3.3.6.

Feld et al.: In-car positioning In [Feld et al., 2010], a method based on speech audio signals is described to position passengers inside a car, i.e. to determine which person sits on which seat. This 'in-car positioning' provides viable information for personalization services, e.g. to adjust the seat according to the occupant's preferences or to enable speech commands like 'open my window'. The proposed in-car positioning method uses the fact that the number of possible positions of passengers in a car is highly restricted. Since each possible position is equipped with a microphone, which is useful for other in-car applications as well, positioning can be done by directly monitoring the power of the output signal of each microphone, instead of using beamforming. The system uses directional microphones, which are installed in front of each seat. To protect the privacy of the user, an onboard/exocentric approach is used, in which the microphone signals – together with an indication of the seat at which it is installed – are sent to the mobile devices of the users. Each user's mobile device contains a user profile, which includes a voice-print of the user, i.e. a model describing characteristic features of the user's voice. The mobile device is thus able to compare incoming microphone signals with the voice-print of its owner and can determine the correct seat. The mobile device can use this information, for example to adapt the ringtone in case the user is sitting in the drivers seat. Additionally, users can specify whether they want to share their position information with the car. The system was evaluated in a Mercedes R-Class with four installed microphones (driver, front passenger, rear-passenger left and right). The system was trained with 10 adult speakers, using 30 minutes of speech for each speaker. For the evaluation, 76 minutes of speech were recorded in varying conditions, e.g. doors open, doors closed, overlapping dialog. The system tried to classify different lengths of speech. With 10 milliseconds of speech, the system already reached an accuracy of 62.7%. With about one second of speech, the accuracy raised to almost 100%.

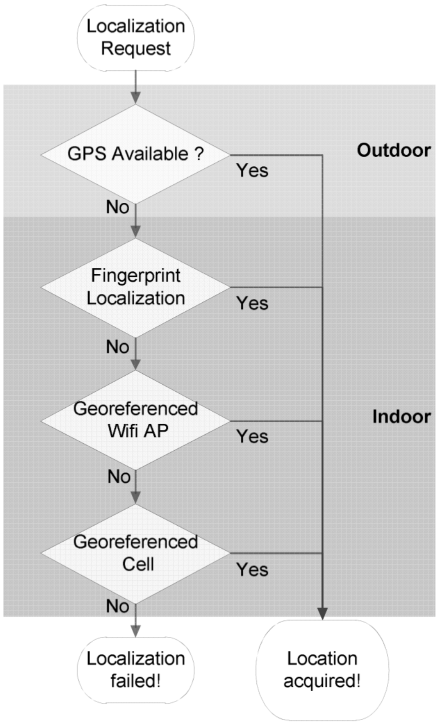
3.2 Inertial Positioning

Since inertial positioning relies on a number of proprioceptive sensors, it is in principle a positioning approach that combines different sensor technologies. The basic principle is dead reckoning, as already mentioned in Section 2.1.1.2: by measuring the current speed and direction of an agent, a new position is calculated, e.g. by applying Newton physics. In principle, inertial positioning does not need any instrumentation of the environment, however an exact start position is needed. Because every sensor is inaccurate to some extend, a position derived by dead reckoning contains errors. These errors add up with each subsequent position determination. In practice, a new position fix has to be gained from time to time to correct the accumulated errors. These correction positions are usually obtained through another positioning technology. Integrated devices containing sensors for inertial positioning are called Inertial Measurement Units (IMU) and are commercially available by several companies. Often they also contain a GPS receiver, to gain the needed start- and correction-positions. Most IMUs contain accelerometers to determine movements, gyroscopes to measure angle changes and magnetometers (compass) to gain direction information according to magnetic north. In robotic positioning, wheel encoders can additionally be used to count the number of wheel turns since the last position determination. By using knowledge about the circumference of the wheels and the number of rotations per wheel, the distance and direction of the movement can be computed. This method is called odometry. For humans the term odometry is often used to describe the counting of steps and sensors that deliver this kind of information are called odometers.

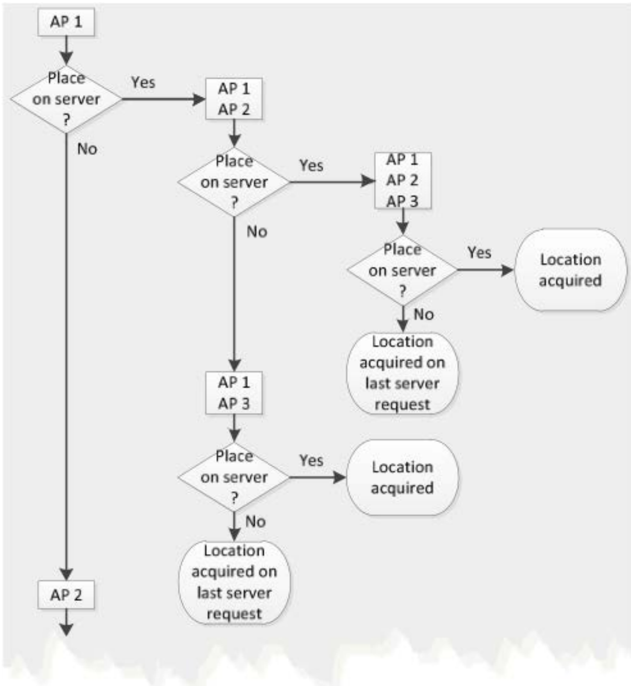
Köppe et al. In [Köppe et al., 2011] an IMU is described, which was specifically designed for safety and rescue applications. The device is called BodyGuard and consists of a GPS receiver, a 3-axis accelerometer, a gyroscope and a digital compass as well as an air-pressure sensor, to determine the altitude. Besides the IMU sensors, BodyGuard is also capable to measure temperature, humidity and the heart-rate of the user. The device can communicate through radio transmissions on the 868 MHz band and can store all measurements on an SD-card. Moreover, BodyGuard was designed for harsh environments and can operate in a temperature range between -25°C and 70°C. Although the IMU contains a processing unit, the authors used an external PC to do the position determination. The reported error accumulation of the system is 'less than ± 2 at a traveled distance of 100 meters'.

Link et al. [Link et al., 2011] present a navigation application called FootPath that relies solely on the built-in accelerometer and compass of a smart phone. The application uses maps from OpenStreetMap[23], which also contains rudimentary support for indoor maps. After a user has selected their current position and their destination on the map, a route is calculated. The application performs a step detection using the accelerometer measurements. Basically, the step detection works by detecting steep drops in the acceleration, which are produced by an up and down movement while walking. Positioning is done by assuming that a user is following the proposed route. When the user takes the first step, a normal stride length l is assumed and the next position is assumed to be at distance l on the route. In subsequent steps, the system compares the direction measured by the compass with the assumed direction derived from the route. If discrepancy is too big for a number k of subsequent steps, it is assumed that the position estimation is wrong and the system tries to find a new matching position on the path, by searching for a route-segment that best matches with the k misaligned steps. If such a segment is found, the algorithm tries to adjust

[23] http://www.openstreetmap.org/



(a) Position acquisition process



(b) Excerpt of fingerprinting positioning process

Figure 3.12: Flowcharts of LOCATEME's position acquisition and fingerprinting process ([Pereira et al., 2011a]).

the stride length l in such a way, that all steps taken so far lead to the found route-segment. The new user position is assumed to be at the end of that path-segment. The authors tested FootPath in an outdoor scenario on a predefined path using GPS as ground-truth. The reported average accuracy is 8.9 meters.

3.3 Positioning with Several Sensor Technologies

3.3.1 Pereira et al.: LocateMe

[Pereira et al., 2011a] implemented an application named LOCATEME, which they describe as a localization system to find friends wherever they are. The system runs on Android mobile phones and also provides a website, where users can register to manage their data and see current positions of their friends. The positioning for LOCATEME uses GPS outdoors and WiFi as well as cellular phone networks indoors. For indoor positioning, three different methods were implemented: a WiFi fingerprinting approach, a method based on geo-referenced WiFi access-points and one based on geo-referenced cells. Figure 3.12a shows the flow-chart of the position acquisition process: LOCATEME successively tries all implemented methods until one is successful or all failed.

The training for the WiFi fingerprinting method is initiated by the user, who can then indicate their position on a displayed map and provide an additional position description and floor number. LOCATEME scans for all available access points and sends the indicated position and the resulting scan data, including MAC address, SSID and signal strength values, to a server. In the actual positioning mode, the user's device will perform the same scanning process and send the results to the server for position calculation. The server recursively checks if the currently detected access points are available in its database and searches for the database entry, which contains most detected access points. Figure 3.12b shows an excerpt of this search process for three detected access points.

For the second method, a database containing geo-referenced WiFi access points is needed. If GPS is available, LOCATEME generates this database automatically by constantly scanning for WiFi access points and sending this data along with the current GPS position to a server. If GPS and the fingerprinting method fail while trying to determine a position, the server tries to estimate a position by calculating a weighted center mass, using the measured signal strength of detected access points as weights.

The last method is practically the same as the previous one, only that detected cells are used instead of WiFi access points.

Since LOCATEME provides its sensor measurements to a server for the actual position determination, it is an offboard egocentric system. Although several sensor-types are used, no sensorfusion is performed. Semantic descriptions of indoor locations can be provided, but no ontology or hierarchical location model is used. The authors do not provide an accuracy evaluation of their positioning system.

3.3.2 Gallagher et al.

[Gallagher et al., 2011] describe a system running on mobile phones that should help students and staff at University of New South Wales to find their way through the complex campus and gives them information about nearby POIs, like ATMs and bus stops. The emphasis of their system lies on the automatic switching between GPS for outdoor usage and a standard WiFi fingerprinting system for indoors. This switching should help to reduce the power consumption, because either the GPS or the WiFi scanning will be switched off if not needed. The system is server based, working with measurements sent by the mobile devices of the users. It is thus an offboard egocentric system. Users can also provide feedback about their current position to the system, which should help to increase the accuracy.

The switching between fingerprinting and GPS – or from indoor to outdoor – is accomplished through so-called ‘indoor transition zones’, i.e. rooms that contain or are nearby exits of a building. If the server detects a position inside such an indoor transition zone for a given number of times (default value is three times), it will tell the phone to switch on GPS and check if it is able to get a fix. If this is successful, the WiFi scanning will be switched off and the system changes to its outdoor mode. The change from outdoor to indoor is simply triggered by the loss of GPS signals.

The authors performed tests to evaluate the probabilities of correct switches and noticed that the switch from indoors to outdoors is always delayed due to the time GPS needs to provide a first fix. At low speeds (1 m/s), the correct switch was performed in 97% of all tests. This number decreases with increasing speed. At the highest tested speed (3 m/s) only ≈50% of changes were detected correctly.

3.3.3 Peng et al.

[Peng et al., 2011] developed a seamless outdoor/indoor positioning system for vehicle and pedestrian positioning using GPS and an active RFID system. The RFID system contains of a reader card and active RFID tags manufactured by Identec Solutions (see also Section 4.4.1.1). For pedestrians an additional inertial measurement module called the MinimaxX[24] was used. The MinimaxX contains a tri-axis accelerometer, three gyroscopes, a tri-axis magnetometer and has a built-in GPS receiver. The authors developed a new approach to integrate the measurements of all sensors, which is based on the Reduced Sigma Point Kalman Filter (RSPKF). The authors claim, that this new variation, dubbed Iterated Reduced Sigma Point Kalman Filter (IRSPKF) has less computational cost than the traditional RSPKF and leads to a higher accuracy.

The accuracy of the system was evaluated using two different sites: a test track for the vehicle, where an area with bad GPS reception was augmented with active RFID tags, and a test track for the pedestrian application, which consisted of an outdoor part and an indoor part. The indoor part lead through a house that was mainly constructed of timber; the outdoor part partially lead through canopy-covered areas as well as through open areas. Since the derived position measurements were tested against RTK GPS measurements (see Section 3.1.1.1), it can be assumed that the indoor environment still provided good GPS reception. For the vehicle test track, the

[24] http://www.catapultsports.com/products/minimax

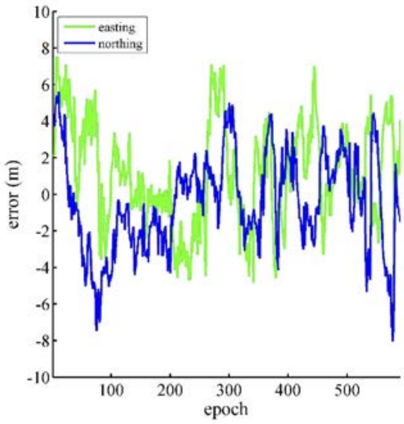


Figure 3.13: Positioning errors of the pedestrian positioning experiment as reported in [Peng et al., 2011]

.

accuracy could be improved from 2.923 meters with the RSPKF approach to 1.353 meters using the IRSPKF approach. For the indoor tests, the authors report a meter-level accuracy, but no exact numbers are provided. A graphical representation of the measured errors (see Figure 3.13) shows a derivation of ±8 meters.

3.3.4 Xiao et al.

[Xiao et al., 2011] propose an egocentric sensor-fusion approach based on WiFi tags, worn by the user, and inertial measurements. Their WiFi positioning is based on RSS fingerprinting, but to reduce the inaccuracies introduced by fluctuations of the RSS measurements, they use a region-based approach instead of single reference points, i.e. several single reference points are grouped into regions. The authors admit that this method will increase the time-effort for the system calibration, but argue that a simulated evaluation of the approach showed an accuracy improvement of 1.5 meters compared to the standard approach with single reference points.

For inertial measurements, the RAZOR IMU[25] by Sparkfun Electronics was used. The RAZOR provides a single-axis gyroscope, a two-axis gyroscope, a tri-axis magnetometer and a tri-axis accelerometer. A WiFi tag manufactured by G2 Microsystems was integrated with the IMU to provide a mobile platform and to send the inertial measurements to a centralized server. This server also receives the RSS measurements from all WiFi access points. The sensor fusion is accomplished on the server with the help of a Kalman filter that uses the WiFi position-estimates as measurements and the inertial sensing data as control inputs.

[25]http://www.sparkfun.com/products/9431

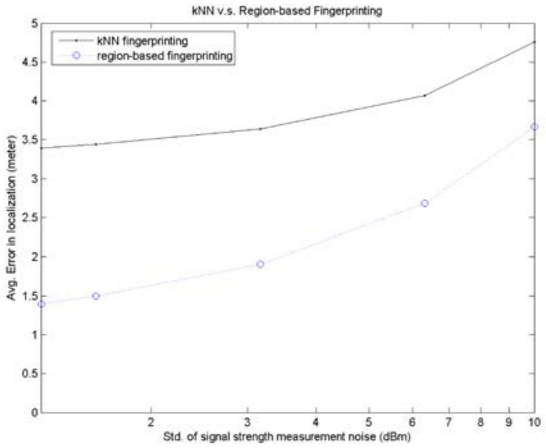


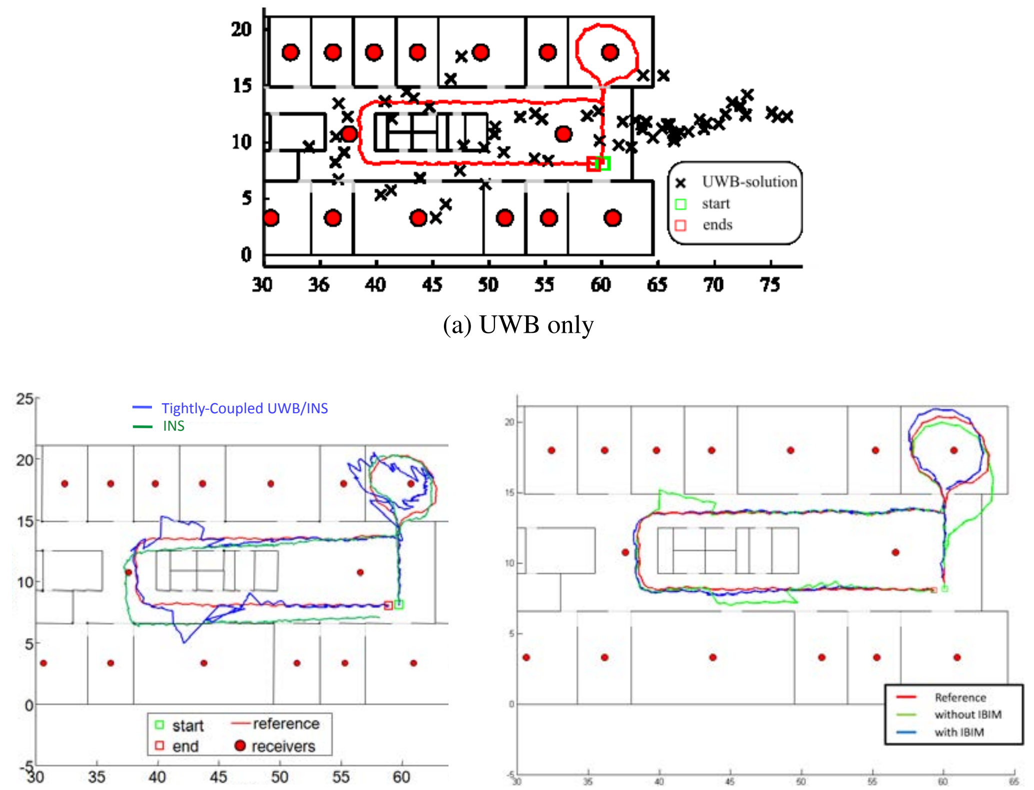
Figure 3.14: Average positioning error as shown in [Xiao et al., 2011].

An evaluation was performed, using a trolley carrying the mobile measuring platform. Four WiFi access points were set up and 24 single reference points where taken during the calibration phase. Varying numbers were tried for the grouping of these single reference points (3, 4 and 6 neighboring points). The authors report an increase of accuracy of 1.2 meter in average through the use of the Kalman filter with IMU provided control inputs. A provided diagram (see Figure 3.14) shows an average positioning error of ≈1.25 meters with regions consisting of three reference points and using a Kalman filter with IMU provided inputs. The accuracy gets worse when regions with more reference points are used.

3.3.5 Ascher et al.

[Ascher et al., 2011] describe initial studies on how to use an UWB based indoor positioning system to provide correction positions to an inertial positioning system. The authors argue that the problem of UWB based positioning systems lies in the fact that a high number of UWB senders or sensors have to be installed in the environment to guarantee a position determination. Inertial position systems on the other hand accumulate errors over time. With the combination of both systems, they want to reduce number of UWB nodes that have to be installed while maintaining a high position accuracy.

The authors did not test their approach in a real-world setting, but developed a simulation suite, which consists of a walk generator and a UWB simulation. With the walk generator, an arbitrary walking-path can be specified including different velocities, which will then be transformed into measurement data that a real inertial measurement unit (IMU, see Section 3.2) would produce. The walk generator first



(a) UWB only

(b) Inertial (INS) positioning vs. tightly coupled UWB & inertial

(c) Tightly coupled with and without filtering (IBIM)

Figure 3.15: Simulation results as reported by [Ascher et al., 2011]

.

produces ideal IMU data, which is then modified according to error statistics from real IMUs.

For the simulation of the UWB positioning system, a 3D model of an indoor environment can be specified, including different wall materials and objects with different electrical properties. With this model the radio propagation of UWB signals can be simulated. The envisioned UWB positioning system is egocentric, i.e. UWB senders are installed in the environment, which send out their own location. The respective sensor also contains the IMU. Whether the system is onboard or offboard is not specified. Since the UWB senders and the receiver are not synchronized, the authors propose to use TDOA measurements and multilateration for position estimation. The complete UWB positioning system can be simulated using the 3D model and appropriate radio propagation models.

The fusion of the UWB positioning system and inertial positioning is accomplished with a Kalman filter. The authors tested two different approaches: a loosely coupled

Kalman filter and a tightly coupled Kalman filter (see Section 2.6.1). The loosely coupled Kalman filter uses the calculated position of the UWB system as measurements and predicts the next position using the IMU measurements. This approach has the disadvantage, that it only works when the sensor can gain at least three TDOA measurements. The tightly coupled Kalman filter also uses the IMU measurements to predict the next position, but then calculates which TDOA measurements it would have to receive, if the predicted position were true. The difference between the predicted TDOA measurements and the real TDOA measurements are then used as measurement innovation. Thus, the tightly coupled Kalman filter works with any number of received TDOA measurements. Additionally the authors developed a mechanism to filter out faulty TDOA measurements, which basically monitors the measurement innovation and marks senders that repeatedly deliver far off TDOA measurements. These marked senders will then be omitted from position calculations. The authors call their filter approach Innovation Based Integrity Monitoring (IBIM).

The proposed methods were tested using the simulation suite. The authors modeled one floor of their office building and simulated 15 UWB senders as well as a walk trough one of the rooms and through the corridor. The authors do not give an average accuracy but provide a number of diagrams. Figure 3.15a shows the performance if only the UWB positioning system is used. As can be seen, most of the positions (black crosses) are far off the track (red line). This is due to the low number of UWB senders in the environment. Figure 3.15b is a comparison between inertial positioning alone (green line) and the combination of inertial and UWB positioning (blue line). As can be seen, the pure inertial position drifts away towards the end of the trace due to accumulated errors. The combined position seems to have some large deviations, when UWB senders from adjacent rooms are detected. Figure 3.15c shows the derived positions of the combined approach with and without enabled IBIM filtering. The large deviations seem to be corrected through the IBIM approach. The authors plan to deploy and test their system in a real-world scenario.

3.3.6 Brunelli et al.

[Brunelli et al., 2007] developed a system that can be used to position people in meetings or seminars. The system is offboard/exocentric and uses microphone arrays as well as cameras that are installed in meeting rooms. The audio based positioning uses the principle of beamforming to derive a global coherence field (GCF) as described in Section 3.1.12.2. The system scans the room with a spatial resolution of 5 centimeters. To deal with disturbances from coherent noise sources, a filter was implemented, which checks the distance of derived positions between successive measurements. Short noises that appear suddenly at a far distance from previous positions are thus ignored.

The camera based positioning is achieved by using a simplified model for a speaker in a meeting scenario. The authors assume that a speaker has a human-like shape and is standing upright most of the time. Furthermore, it is assumed that a possible target has a consistent color throughout a sequence of images. A possible target is identified in an image-sequence by analyzing the optical flow and trying to project a coarse 3D model of a human onto parts of the image that indicate a high optical flow (similar to the description in Section 3.1.6.2). A single camera obtains several hypothesis of where a person is currently standing. The integration of hypotheses from different cameras is done with a particle filter, where the position state contains position and velocity of a target, and the measurements of each camera are used to update the particles accordingly.

Since audio based positioning is only possible while a person is talking, the camera based positioning is more likely to constantly determine a position. To combine the camera based positions with the audio based ones, the particle filter for the camera based positioning was adapted. The basic principle is to derive a hypothesis for the audio based positioning from the current camera based position, i.e. if a person is standing at a particular position, the hypothetical position of a sound source can be calculated, by shifting the coordinates towards the head of the 3D model. The hypothesis can be checked, by forming beams around the area of that position. If a defined threshold is reached, which indicates that a person is really speaking, the computed GCF values are used to update particles in that region accordingly.

The authors tested the combined system as well as each single positioning method. For audio only, an average accuracy of 14.4 centimeters was reached, with 7 microphone arrays (as depicted in Figure 3.11). For multiple speaker positioning, the average accuracy dropped to 21.8 centimeters. Positioning based only on camera information reached an accuracy of 13.2 centimeters for single person tracking and four cameras.

The combination of both methods for a single person was tested in two conditions: only position a user when they are speaking and position a user for every possible point in time (regardless whether they are speaking or not). In the first condition, an average accuracy of 13.2 centimeters was reached. The second condition is reported with a slightly lower accuracy of 13.4 centimeters. The fusion of both positioning methods did thus not result in a higher average accuracy than using a single method. However, the authors report that the fusion of both methods performed better in some observed sequences.

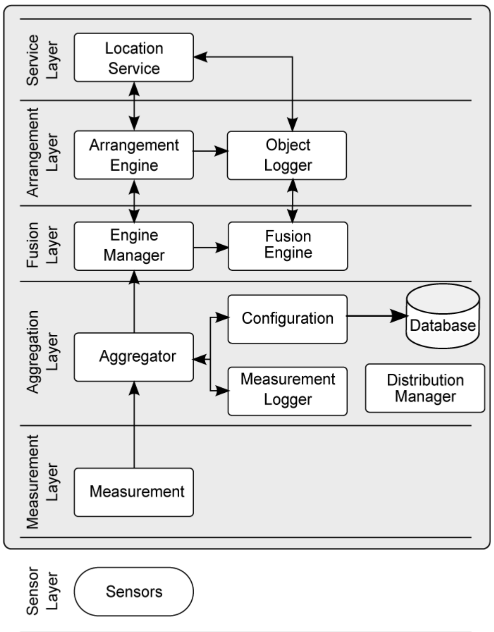


Figure 3.16: The architecture of the MapUme middleware ([Najib et al., 2011]).

3.3.7 Najib et al.: MapUme

In [Najib et al., 2011] a middleware called MapUme is presented, which is designed for offboard positioning. The middleware was implemented using Windows Communication Foundation (WCF), which is part of Microsoft .NET.

MapUme can run on a single server or in a distributed server-network, which should help to balance the load of the position computations. The architecture of MapUme is shown in Figure 3.16. The depicted sensor layer represents the actual sensor hardware. In the measurement layer, data structures and interfaces for each sensor have to be defined. The aggregation layer collects sensor data and is also responsible for the communication in a distributed server-network. Furthermore, this layer is responsible for the configuration (via XML files) of the middleware, for measurement logging and database access, through which maps, fingerprints, locations of base stations etc. can be stored. As the name fusion layer implies, it is responsible for the fusion of sensor data. The fusion engine uses an abstract factory pattern, which allows to implement different fusion algorithms that can be 'plugged' into the fusion layer. In the arrangement layer, tracked objects are represented with their current position and relations to an environment description, e.g. a map, can be derived. The service layer allows to implement location-based services, to which other applications can subscribe.

The authors tested MapUme, by implementing an offboard/exocentric WiFi positioning system, where the WiFi access points report their measured signal strengths directly to a MapUme server. To test the fusion engine, they added an IMU. A two-story building was equipped with eight WiFi access points on an area of 25×70 meters. The average position accuracy for WiFi based positioning is reported with 2.52 meters. The integration of the IMU was done with a particle filter and resulted in an average positioning accuracy of 2.27 meters. The exact evaluation methods are not disclosed.

The authors also tested how the middleware performs with a single server and in a distributed server-network and found out, that the distributed mode introduces a low network-latency of 3.436 milliseconds, which is negligible compared to the average computation time of 366 to 392 milliseconds per measurement.

3.3.8 Martínez et al.

[Martínez et al., 2011] propose a high-level interface to combine several positioning technologies and to provide location-based services. Although high-level descriptions are given, a concrete implementation does not seem to exist. The authors envision an architecture that is divided in two main parts: Location Event Providers (LEP) and Location Services (LS). The basic idea is to separate the technology dependent parts from the technology independent parts, i.e. the Location Event Providers are responsible for determining positions using specific hardware, while the Location Services use the provided positions and are thus independent from the hardware.

According to the authors, there should be a Location Event Provider for every supported position technology. A service discovery protocol is responsible for detecting available technologies, like WiFi, Bluetooth, UWB, and starts the according Location Event Providers. These providers can broadcast position events using a standardized protocol, which contains an event type description, a time-stamp and a description of the area where a user is in.

Location Services can receive these position events. If position events for the same person or object arrive, the authors propose to merge the reported areas through geometrical intersection. If no intersection is found, the Location Service chooses the position that was reported from the Location Event Provider that is known to have the highest accuracy. Using the obtained position, a Location Service can then implement additional services or forward the determined position.

System Description	Sensor Fusion	Seamless Outdoor/Indoor	Semantic Descriptions	Coordinates	Egocentric	Exocentric	Onboard	Designed Instrumentation	Opportunistic	Dynamic Evaluation	Static Evaluation	Accuracy	Technology
Ghinamo et al., 2011 (Indoor GPS)	✗	✓	✗	✓	✓	✗	✓	✓	✗	∼	∼	0.34 m - 1.04 m	GPS
Kohtake et al., 2011 (Pseudolites)	✗	✓	✗	✓	✓	✗	✓	✓	✗	∼	∼	n.a.	IMES Pseudolites
Sakamoto et al., 2011 (Pseudolites with robot)	✗	✓	✗	✓	✓	✗	✓	✓	✗	✗	✓	$\approx 0.17m$	IMES Pseudolites, rotating antennae
Pereira F. et al., 2011b (GSM at LHC)	✗	✗	✗	✓	✓	✗	✗	✗	✓	✗	✓	20-280 m	GSM over a leaky feeder
Dempsey et al., 2011 (Femtocells)	✓	∼	✓	✗	✗	✓	✗	✗	✓	∼	∼	Room level	Pico-, Femto-cells, Calendar
Bahl et al., 2000 (RADAR)	✗	✗	✗	✓	✗	✓	✗	✗	✓	✗	✓	2-3 m	WiFi
Ledlie et al., 2011 (Molé)	✗	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	Room level	WiFi
Chawathe, 2009	✗	✗	∼	∼	✓	✗	✓	✓	✗	✗	✗	n.a.	Bluetooth
Kiers et al., 2011 (ways4all)	✗	✗	✗	✓	✓	✗	✓	✓	✗	✗	✗	na	passive RFID
Ni et al., 2004 (LANDMARC)	✗	✗	✗	✓	✗	✓	✗	✓	✗	✗	✓	1-2 m	active RFID
Want et al., 1992 (Active Badge)	✗	✗	✗	✓	✗	✓	✗	✓	✗	✗	✓	Room level	Infrared
Hauschildt et al., 2010 (ThILo)	✗	✗	✗	✓	✗	✓	✗	✓	✗	✓^{pg}	✗	9-26 cm	Thermal Infrared
Herranz et al., 2011	✗	✗	✗	✓	✓	✗	∼	✓	✗	✓^{sim}	✗	3.1-17.3 cm	Visual LEDs, Camera
Ruotsalainen et al., 2011 (Heading direction)	✗	∼	✗	✗	✓	✗	∼	✗	✓	✗	✓	$1.3°$- $1.8°$	Camera
Dettori, 2008	✗	✗	✗	✓	✗	✓	✗	✓	✗	✗	✓	5-10 cm	Stereo Camera
Nakashima et al., 2003 (CoBIT)	✓	✗	∼	∼	✗	✓	✗	✓	✗	∼	∼	≈ 1 cm	Laser-range, Camera
Rabinowitz et al., 2005 (DTV)	✗	✓	✗	✓	✓	✗	✓	∼	✓	✗	✓	3.2 -23.3 m	TV receiver
Moghtadaiee et al., 2011 (FM Radio)	✗	∼	✗	✓	✓	✗	✓	✗	✓	✗	✓	2.96-3.29 m	FM Radio receiver
Storms et al., 2010	✗	∼	✗	✓	✓	✗	✓	✗	✓	✓^{pg}	✗	0.2-0.6 m	Magnetometer
Blankenbach et al., 2011 (MILPS)	✗	✗	✗	✓	✓	✗	✓	✓	✗	∼	✓	4-7 cm Range only	Magnetometer

Ubisense, 2005 Stephan et al., 2009	✗	✗	✗	✓	✗	✓	✗	✓	✗	✗	✓	12.8-124.3 cm	UWB
Kuflik et al., 2011 (PIL)	✗	✗	✗	✓	✓	✗	✗	✓	✗	✗	✓	1.5-2 m	WSN
Rosa et al., 2011 (Relative Positioning)	✗	✗	✗	✓	✓	✗	✓	✗	✓	✗	✓	meters	WSN of mobile devices
Ward et al., 1997 (UltraBat)	✗	✗	✗	✓	✗	✓	✗	✓	✗	✗	✓	$\approx$ 10 cm	Ultrasound
Baunach et al., 2007 (SNOW BAT)	✗	✗	✗	✓	✗	✓	✓	✓	✗	$\sim$	$\sim$	15 mm	Ultrasound WSN
Feld et al., 2010 (In-Car Positioning)	✗	✗	✓	✗	✗	✓	✓	✓	✗	✗	✓	Seat level	Microphone
Köppe et al., 2011	✓	✓	✗	✓	✓	✗	✗	✗	✓	$\sim$	$\sim$	$\pm$2m/100m	IMU
Link et al., 2011 (Footpath)	✓	✓	✗	✓	✓	✗	✓	✗	✓	✓^{route}	✗	8.9 m	Accelerometer, Compass
Pereira C. et al., 2011 (LocateMe)	✗	✓	✓	✓	✓	✗	✗	✗	✓	✗	✗	n.a.	GPS, WiFi
Ghallager et al., 2011	✗	✓	✓	✓	✓	✗	✗	✗	✓	✗	✗	n.a.	GPS, WiFi
Peng et al., 2011	✓	✓	✗	✓	✓	✗	$\sim$	✓	✗	✓^{rtk}	✗	meters	GPS, active RFID, IMU
Xiao et al., 2011	✓	✗	✗	✓	✗	✓	✗	✓	$\sim$	✓^{pg}	✗	$\approx$ 2 m	WiFi tag, IMU
Ascher et al., 2011	✓	✗	✗	✓	✓	✗	✓	✓	✗	✗	✗	n.a.	UWB, IMU
Brunelli et al., 2007 (audio & video)	✓	✗	✗	✓	✗	✓	✗	✓	✗	✓	✗	$\approx$ 13 cm	Cameras, Microphone arrays
Najib et al., 2011 (MapUme)	✓	$\sim$	✓	✓	✗	✓	✗	✓	✓	✗	✗	2.27 meters	Middleware, WiFi, IMU
Martínez et al., 2011	✓	$\sim$	✓	✓	$\sim$	$\sim$	$\sim$	✓	✓	✗	✗	n.a.	High level interface, no concrete implementation
LOCATO	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓^{nt}	✓	Room level, $\approx$ 1 m	Active RFID, IR, Bluetooth, Cells, WiFi, GPS

Table 3.2: Comparison of multiple-sensor systems and single-sensor based positioning systems with LOCATO.

^{pg} Predefined traces were used as groundtruth, derived from geometric primitives (straight lines, rectangles)
^{sim} A simulation was used for the evaluation
^{route} A calculated route was used as groundtruth
^{rtk} The positioning system was tested against RTK GPS measurements
^{nt} Natural traces were used as groundtruth, derived from observation and manually annotated

3.4 Summary and Discussion

This chapter provided an overview on the state of the art of positioning systems and techniques. For outdoor applications, GPS provides high position-accuracy if enough satellites are in line of sight. With GPS RTK, a position accuracy in the range of one centimeter is possible, but the applicability is limited due to higher hardware complexity. As AGPS is supported by most modern cell-phones, this can be seen as the standard for outdoor positioning. Galileo is not expected to surpass the accuracy of GPS, but hybrid GNSS receivers, which will be able to use GPS, Galileo and GLONASS satellites, will be able to benefit from the combined satellite-coverage. 4G cellphones will at least be able to provide coarse position-information in the range of 50 to 150 meters, even if no or not enough satellites are in the line of sight.

Regarding indoor positioning, GNSS pseudolites are an interesting alternative as they theoretically enable outdoor and indoor positioning by using only one receiver, but issues like the near-far problem still have to be solved. [Sakamoto et al., 2011] showed that an accuracy in the range of centimeters is possible, but the needed hardware can not be integrated into small mobile devices and thus the approach does not comply with the usability and applicability criteria for positioning systems.

A plethora of alternative solutions is available, where the large part of them is using electromagnetic signaling. The highest position accuracy can be reached by instrumenting the environment. Millimeter accuracy seems to be possible by using ultrasound ([Baunach et al., 2007]) and by using laser-scanners ([Nakashima and Hasida, 2010]). However, the former needs a dense network of ultrasound nodes and a high calibration effort, and the latter is rather expensive. Thermal infrared as proposed by [Hauschildt and Kirchhof, 2010] is an interesting idea with the potential of providing centimeter accuracies, but is still far from being a robust solution.

Opportunistic systems provide an accuracy in the range of meters or room-level and are an interesting alternative, as no additional infrastructure has to be deployed. [Storms et al., 2010] could show that the natural differences in the Earth's magnetic field could possibly be used for positioning, but the proposed approach is still highly sensitive to external influences. WiFi infrastructures are in widespread use and are thus good candidates for opportunistic positioning systems. The approach to enrich WiFi access points with positioning data, as proposed by [Gschwandtner and Schindhelm, 2011], could help to ease the process but would still need administration by the network operator and thus increases the cost of ownership.

The fingerprinting approach is the most promising for opportunistic systems, as it can also work with a small number of access points, as opposed to trilateration, multilateration or triangulation, which all need a minimum number of measurements. However, signal-strength fingerprints are subject to changing environmental-conditions, like air-humidity and the number of people in the environment. When a crowd-sourcing approach is used to gain the needed fingerprints for reference points, another problem comes into play: different mobile devices can have different reception or sending characteristics, leading to an incompatibility of fingerprints. The Localization Toolkit LOCATO, which is described in detail in Chapter III, tackles both problems.

Still, WiFi infrastructures are not available everywhere and depending on the application, a higher position accuracy might be needed. As indicated in Section 2.6, sensor fusion can be a solution to this problem. However, as the systems in Section 3.3 show, sensor fusion is mainly used to integrate IMU measurements into a position-giving system. Most systems use Kalman or particle filtering, where the filters are especially tailored to fit the respective system. [Najib et al., 2011] and [Martínez et al., 2011] proposed more general approaches, but the former is designed for offboard positioning and the latter uses the very simple approach of geometrical combination of reported positions. LOCATO tackles the problem of a more generic sensor-fusion as well.

The paper of [Stephan et al., 2009] showed that positioning systems can perform different than advertised in realistic scenarios. Most positioning systems are tested under optimal conditions, i.e. interfering factors are suppressed. Often the exact method of evaluation is not given, especially the origin of the ground truth. A static evaluation is often performed, i.e. reference points are used with which the derived position is compared. This method is well suited for positioning systems that want to derive a room-level or large-area accuracy. Positioning systems with higher accuracy should be tested with moving targets. Here, the ground truth is hard to obtain, and thus predefined paths are often used, which may not coincide with natural paths that are taken by users. This problem will be thoroughly discussed in Section 4.4.4.1.

Table 3.2 shows a comparison of all systems described in this chapter with LOCATO. The rows are ordered according to the sequence in which the systems appear in this chapter. ✓ indicates that a feature is present, ✗ that it is not present and $\sim$ indicates that the feature cannot be derived from the description of a system.

The column ‘System Description’ contains a reference to the paper where the according system is described as well as the system name, if one exists. A system is marked as being able to perform ‘Sensor Fusion’, if it fuses at least two different sensor technologies. Seamless Outdoor/Indoor describes if a system is capable of working outdoors and indoors without having the user to switch systems. A system that simply switches between a sensor technology for outdoor positioning and a sensor technology for indoor positioning does not classify as performing ‘Sensor Fusion’.

The column ‘Semantic Descriptions’ denotes systems that internally use semantic descriptions to represent positions. Likewise, ‘Coordinates’ denotes systems that use numerical coordinates for internal position representation. The columns ‘Egocentric’ and ‘Exocentric’ denote egocentric and exocentric systems. As described in Section 2.3.2.5, any onboard positioning system can be converted into an offboard positioning system. Thus only the column ‘Onboard’ is represented in the table. Offboard systems are marked as not being onboard. The column ‘Designed Instrumentation’ denotes whether a system needs a dedicated infrastructure, whereas the column ‘Opportunistic’ denotes systems that use an already existent infrastructure. Systems that were evaluated with moving position-targets are marked in the column ‘Dynamic Evaluation’. The column ‘Static Evaluation’ denotes systems that were evaluated by determining positions at known reference points, without moving the position targets. The accuracy of a system, if available, is given in the column ‘Accuracy’. In the column ‘Technology’, the used sensor technologies are listed.

Part III

LocaTo

4 LOCATO: LOCALIZATION TOOLKIT

With respect to the design criteria, which were specified in Section 1.1.5, it could be seen in the last chapter that there is no single positioning-system that provides a global optimum over all criteria. A building-owner or tenant, who wants to provide a positioning system, would thus choose a system that is tailored to their capabilities and needs. The main criteria will usually be the cost of infrastructure and the position accuracy, where the latter will depend on the applications that an operator wants to support. Regarding robustness of positioning systems, it could also be seen that trilateration, multilateration or triangulation tend to be less robust, due to their need of a line of sight to a specific number of sensors or senders. Furthermore, any positioning method that relies on signal-strength measurements is sensitive to environmental changes, like air-humidity or the number of people in the room.

The Localization Toolkit LOCATO was designed to facilitate the creation of positioning systems that are tailored to the needs of an operator while taking the Always Best Positioned paradigm into account, which addresses the needs of the users of positioning systems. In order to provide a high robustness, LOCATO provides newly developed positioning methods, which work without signal-strength measurements and can derive a position even if only one sender or sensor is within reach. Furthermore, the algorithms provide easy ways to perform sensor fusion and are easy to extend with more sensors. In addition, the toolkit provides methods to access a ubiquitous user-modeling cloud-service and a local blackboard-service.

4.1 Overview on the Localization Toolkit

LOCATO provides basic building blocks and additional tools to easily design and deploy positioning systems. It contains three core algorithms, each addressing different needs:

- **Proximity Detection:** This core algorithm can be used for low cost offboard-exocentric positioning. As the name already implies, it relies on simple proximity sensing and thus provides only coarse position accuracy.
- **Frequency Of Appearance (FOA):** This core algorithm is designed for onboard/egocentric *opportunistic* positioning. The FOA algorithm provides a novel, specially designed fingerprinting method that omits signal strength measurements and thus enables a more stable position determination and eases the process of sharing user-collected reference-fingerprints.
- **Geo Referenced Dynamic Bayesian Networks (geoDBN):** This core algorithm is designed for onboard/egocentric positioning systems with *designed instrumentations*. The geoDBN algorithm is another novelty and is designed to built high accuracy positioning systems that are easy to extend with new sensors.

The FOA and geoDBN algorithms follow the Always Best Positioned paradigm in that they are easily expandable with further sensors and that they work with any subset of provided sensors. Each core algorithm is available in Java, FOA and geoDBN are additionally available in C++. For each of the three core algorithms, example systems have been built, which are described in more detail in Section 4.2, Section 4.3 and Section 4.4.

Additionally, LOCATO provides support to the cloud service UBISWORLD (see Section 2.4.2.1), which allows to update user-profiles with position information and also provides access to the spatial ontology UBISEARTH. The connection to UBISWORLD is complemented with access methods to a local blackboard service, which allows users to connect to a local infrastructure, e.g. in a shop or airport, which in turn can provide additional context- or location-aware services, like automatic door opening, or navigation services. As FOA and geoDBN support the creation of onboard/egocentric positioning systems, it is up to the user to decide whether they want to connect to UBISWORLD or the blackboard service and how much information they are willing to share.

Figure 4.1 shows the components of LOCATO in a block-diagram. In the following, the three core algorithms as well as the external connections and tools will be described in more detail.

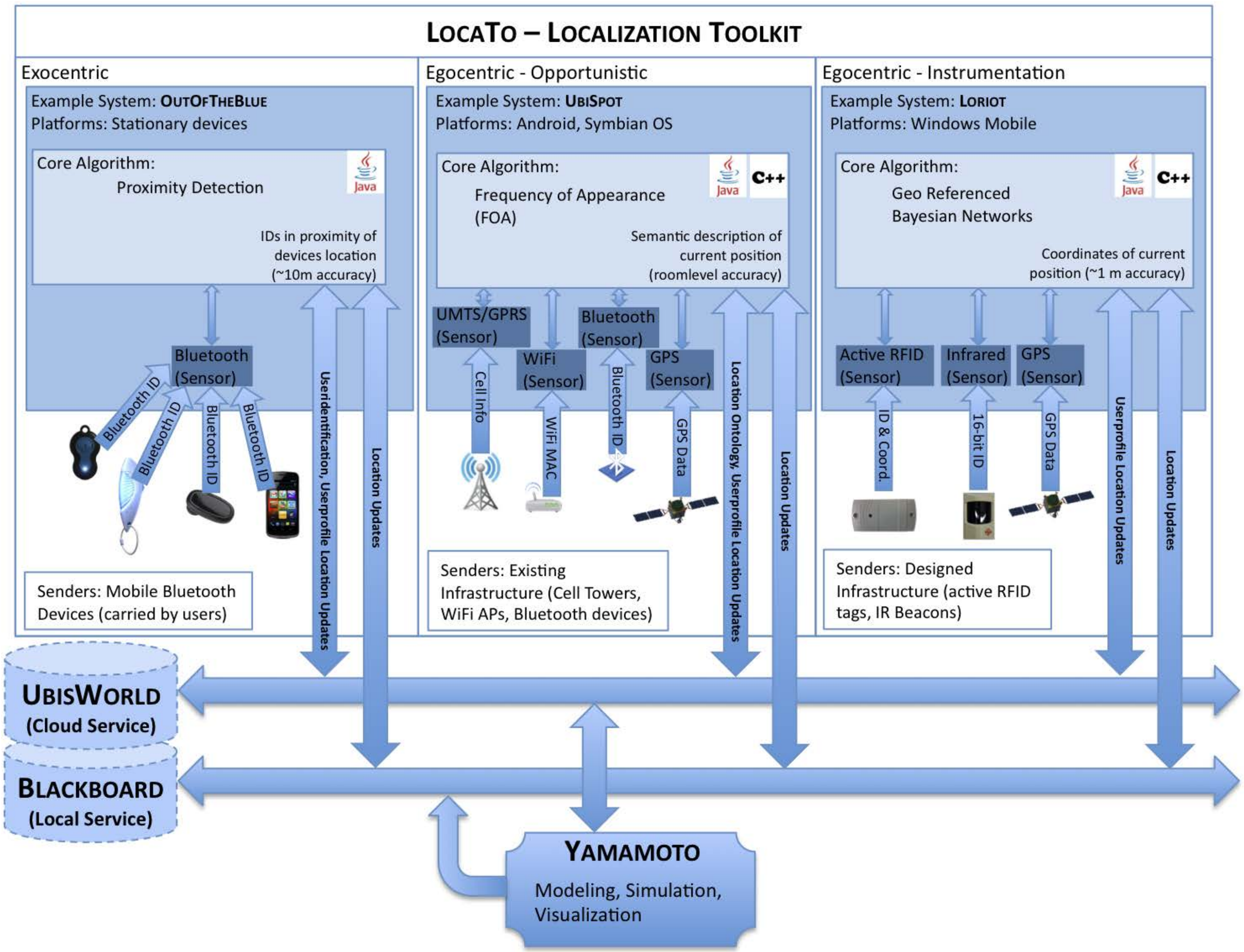


Figure 4.1: Overview on LocaTo – Localization Toolkit

4.1.1 Proximity Detection

The proximity detection algorithm is very simplistic and is tailored to the use of Bluetooth, although it can be easily extended to other sensor technologies. However, as it is a proximity detection, it should mainly be used with near-sensors, like NFC or passive RFID. As it is designed for offboard/exocentric positioning, sensors have to be installed in the environment, which must have a Java compliant computation device attached. The basic idea of the proximity detection algorithm is to periodically inquire the attached sensor and either use the data locally or to forward it to the blackboard service. When used locally, it can only be inferred that the sensed sender is in the proximity of the device. When the data is forwarded to the blackboard, sensor devices can exchange data, which allows a higher accuracy for positioning or the inference of a moving direction. Furthermore, algorithms running on other computing devices in the environment can subscribe to the data and can in turn provide location based services. Of course it is possible, to use different kinds of sensors, which all report their data to the blackboard. However, the core algorithm only provides the basic structure to periodically scan the sensor and to forward the data on the blackboard. All other inference has to be done by additional services.

A Bluetooth based example system called OUT OF THE BLUE is described in more detail in Section 4.2.

4.1.2 Frequency-Of-Appearance Fingerprinting (FOA)

The basic method of fingerprinting was described in Section 2.5.3. In Section 3 it could be seen that all fingerprinting systems incorporate the measured signal strength into their fingerprints. However, the signal strength is also very sensitive to environmental conditions, e.g. air humidity and people present in a room. Furthermore, reference fingerprints have to be collected, either by the operators of the building or by users themselves. With the latter method, a crowd-sourcing approach seems very reasonable, since this distributes the work onto many shoulders, and in the case of an onboard/egocentric opportunistic positioning system, no cooperation of the building owner is necessary to enable a working positioning system. In order to enable such a crowd-sourcing approach, the collected fingerprints have to be compatible to different devices. In this case, the incorporated signal strength poses another problem, since the measured signal strength also depends on the device itself, i.e. the antenna design, the case design including the used materials, the remaining battery strength and the used chipsets influence the signal strength measurements.

The developed Frequency-Of-Appearance (FOA) fingerprinting overcomes these problems, by replacing the signal strength measurements through observations on

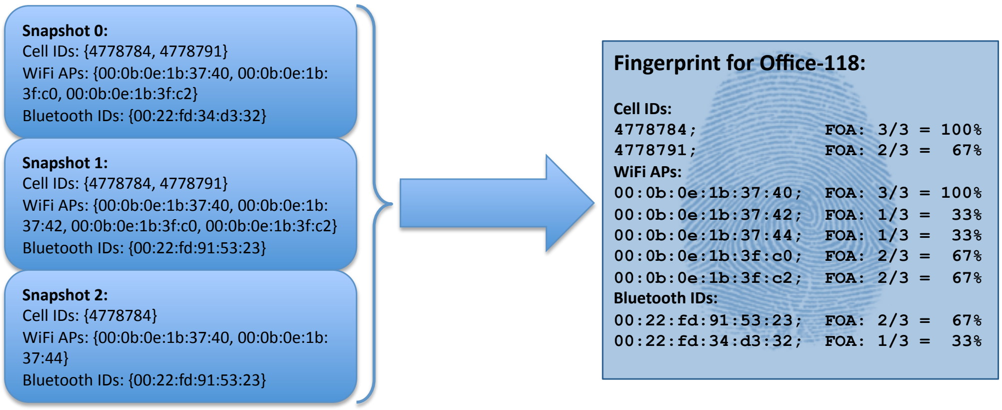


Figure 4.2: Example calculation of a Frequency-of-Appearance fingerprint with $m = 3$: The appearance of each ID in all three snapshots is counted out and the relative appearance is stored in the fingerprint along with a semantic description of the current position.

how often a particular sender was sensed over a period of time. Furthermore, FOA fingerprinting can be used to fuse an arbitrary number of sensors in the sense of the Always Best Positioned paradigm.

In order to acquire these FOA fingerprints, repeated measurements – so called snapshots – are taken. The duration of such a measurement ($SnTime$, short for Snapshot $Time$) depends on how fast the device can complete the scans for each sensor. Depending on the sensor type, increasing the $SnTime$ can result in detecting more senders.

Each snapshot contains a list of all sensed senders from any of the used sensors. Depending on the sensor type, or more specifically, on the data that each sensor delivers, an identifier has to be identified that uniquely describes a sensed sender. For example, the ID of a cell, the MAC of a WiFi enabled device or the MAC of Bluetooth enabled device. Each ID (since MAC addresses are just another form of identification, the term ID is from now on used to denominate MAC addresses as well as any other form of ID) can appear only once in one snapshot.

An FOA fingerprint is generated by taking a specified number m of snapshots and then counting how often each ID was seen in those m consecutive measurements. Since an ID has to be seen at least once to be part of the fingerprint, and because it can at most be seen in every snapshot, it follows that $1 \leq counter_{ID} \leq m$, where $counter_{ID}$ is the counter for a specific sender ID. An example calculation with $m = 3$ is given in Figure 4.2. This example is taken from UBISPOT, which is described in more detail in Section 4.3. The resulting reference-fingerprint is stored along with a representation of the position in which the measurements were taken.

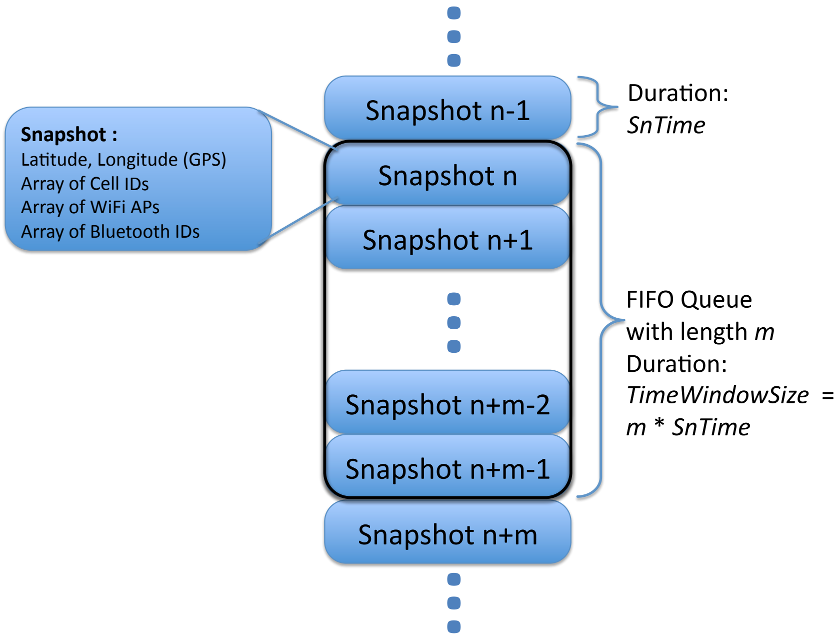


Figure 4.3: Calculation of Frequency-of-Appearance fingerprints with a FIFO queue of length *m* and resulting duration *TimeWindowSize*.

Efficient Calculation of FOA Fingerprints

In order to efficiently calculate FOA fingerprints, incoming snapshots are organized into a FIFO queue with an adjustable length. Figure 4.3 shows the general structure of such a queue with length m. Since the length of the FIFO largely contributes to the time that is needed to collect the data for one fingerprint, it is called $TimeWindowSize$.

When starting the FOA fingerprinting, each snapshot entering the FIFO is analyzed and for each detected ID new counters are initialized or existent counters are increased, depending on whether the ID has been seen before. The time that is needed to fill a queue with length m can be calculated by multiplying the queue length with the time needed to obtain one snapshot, i.e. $m * SnTime$. To obtain the first fingerprint, the counters are normalized by the length m of the FIFO, to get a value that is independent of the queue length. After this initial calculation, a new fingerprint can be generated every $SnTime$ seconds.

It follows that a subsequent fingerprint can only differ slightly from the direct predecessor, which on one hand is a desired effect, since the FIFO should prevent the system from toggling too fast between different locations. On the other hand, a high value for m will also prevent a fast recognition of an actual room change. Hence, the value m is subject to a trade-off between stability and response of the positioning system. This trade-off will be further analyzed and discussed in conjunction with UBISPOT in Section 4.3.4.

4.1.2.1 Matching Fingerprints to Locations

For position determination, a positioning system has to compare its currently measured fingerprint with the fingerprints stored in a database. This can be done by calculating the strength of a linear relationship between two sets of fingerprints, also known as the correlation coefficient of two random variables ([Clauss and Ebner, 1975], pp. 115–128). Consider two different fingerprints $A = \{a_0, ..., a_n\}$, $B = \{b_0, ..., b_n\}$. Each element in the set indicates different measured IDs of the same type of sender (e.g. WiFi access-points) with relative frequency of appearance a_i and b_i where $i \in \{1, ..., n\}$. The product-moment correlation coefficient $r_{A/B}$ is used to estimate the correlation of A and B:

$$r_{A/B} = \frac{\sum_{i=1}^{n} (a_i - \bar{a})(b_i - \bar{b})}{(n-1)s_a s_b} \tag{4.1}$$

Here, $\bar{a}$ and $\bar{b}$ are the means of A and B, and s_a and s_b are their standard deviations. According to [Cohen, 1988, pp. 109–139], the correlation coefficient between A and B has a significant impact, if the absolute value of r_{ab} lies in the range of $[0.50, 1.0]$.

Example Calculation

Given are two fingerprints in the database, one for location L_1 and one for location L_2, which are close together and thus contain the same access points. Let L_1 be $\{3, 2, 2, 4\}$ and $L_2 = \{1, 4, 0, 2\}$, meaning that the first access point was measured 3 times in location L_1 and once in location L_2. The second access point was measured 2 times in L_1, 4 times in L_2 and so on. Assumed that the current measured fingerprint F is $\{4, 2, 3, 4\}$, the computed correlation coefficients are $r_{F/L_1} \approx 0.8181$ and $r_{FL_2} \approx -0.5606$, meaning that location L_1 has a higher correlation to the current fingerprint than L_2 and thus L_1 is more likely to be the current position.

4.1.2.2 Efficient Calculation of the Correlation Coefficient

By closely examining Equation 4.1, it can be seen that $n + 2$ multiplications, $2n + n - 1 + 1 = 3n$ additions and one division have to be performed in order to obtain the correlation coefficient for two fingerprints, each containing n values, leading to $O(4n)$ when ignoring the possible speed differences between additions, multiplications and divisions. This number of operations can be fairly reduced by further examining the numerator in the fraction of Equation 4.1:

$$\sum_{i=1}^{n} (a_i - \bar{a})(b_i - \bar{b}) \tag{4.2}$$

The dividend alone contributes n multiplications and $3n - 1$ additions to the overall count of operations. By expanding, the sum above can be rewritten as:

$$\sum_{i=1}^{n}(a_i b_i - a_i \bar{b} - \bar{a} b_i + \bar{a}\bar{b}) \tag{4.3}$$

Since every summand contains the constant term $\bar{a}\bar{b}$, it can be taken out of the sum and quickly computed by $n\bar{a}\bar{b}$ and the remaining sum can be split into 3 sums. Thus the term can be rewritten as:

$$n\bar{a}\bar{b} + \sum_{i=1}^{n}(a_i b_i) - \sum_{i=1}^{n}(a_i \bar{b}) - \sum_{i=1}^{n}(\bar{a} b_i) \tag{4.4}$$

The two last sums contain the constants $\bar{b}$ and $\bar{a}$ respectively, meaning that these constants can be taken out of their sums:

$$n\bar{a}\bar{b} + \sum_{i=1}^{n}(a_i b_i) - \bar{b}\sum_{i=1}^{n}(a_i) - \bar{a}\sum_{i=1}^{n}(b_i) \tag{4.5}$$

Taking into account that the arithmetic mean $\bar{x}$ of a set of values x_i is defined as

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{4.6}$$

the values of $\sum_{i=1}^{n}(a_i)$ and $\sum_{i=1}^{n}(b_i)$ can be efficiently calculated by $n\bar{a}$ and $n\bar{b}$, eliminating the need of the last two sums and leading to:

$$n\bar{a}\bar{b} - \bar{b}n\bar{a} - \bar{a}n\bar{b} + \sum_{i=1}^{n}(a_i b_i) = -n\bar{a}\bar{b} + \sum_{i=1}^{n}(a_i b_i) \tag{4.7}$$

Equation 4.1 can thus be rewritten as:

$$r_{ab} = \frac{\left(\sum_{i=1}^{n}(a_i b_i)\right) - n\bar{a}\bar{b}}{(n-1)s_a s_b} \tag{4.8}$$

reducing the computation to $n + 4$ multiplications, $n - 1 + 2 = n + 1$ additions and one division, or $O(2n)$. Compared to the original runtime of $O(4n)$, this is twice as fast for large values of n.

4.1.2.3 Ranking System

As already indicated above, correlation coefficients should at best be applied to measurements of the same class, i.e. only WiFi access points, only cells or only Bluetooth device IDs. In theory, it would be possible to calculate the correlation coefficient over a mixed set of measurements, but taking the different characteristics of different senders and sensors into account, e.g. different ranges and stability of the signals, this seems not a good idea. Instead, the fingerprints are separated into sub-fingerprints, i.e. one sub-fingerprint for each sensor type. This also opens up the possibility to assign different weights to each sensor type.

The correlation coefficient for each sub-fingerprint is calculated separately. To combine these results, a ranking system is used, i.e. score points (abbreviated as *Sc*) are assigned to indicate a level of matching-accuracy. The computed correlation coefficients are used as a basis for these score points. As already mentioned above, a good correlation is given if $|r_{A/B}| \in [0.5, 1.0]$. For example, in UBISPOT only those locations are considered as matching candidates, whose coefficient lies in the interval of $[0.6, 1.0]$. To reduce the computational effort, it is good practice to choose one sensor type as a filter to be able to reduce the candidates for the current position. For example, in UBISPOT, the sensed cell ID is used to preselect only those reference-fingerprints that contain the sensed cell ID. The computation of the score-points is shown in Algorithm 1.

Algorithm 1 FOA Ranking Computation

Let F denote the current fingerprint consisting of m sub-fingerprints for m sensors: $F = \{SF_0, \ldots, SF_m\}$
Let $L = \{L_0, \ldots, L_n\}$ be the set of n candidate-fingerprints, where each L_i consists of m sub-fingerprints SL_{ij} ($0 < i < n$ and $0 < j < m$).

1. For each candidate-fingerprint L_i of L:
   (a) initialize the score-point counter for candidate L_i: $Sc(L_i) = 0$
   (b) For each sub-fingerprint L_{ij} of L_i:
      i. update the score-point counter using the correlation coefficient and a weighting factor w_j
      $Sc(L_i) = Sc(L_i) + w_j r_{SF_j/L_{ij}}$
2. Select the candidate L_k with the highest achieved score-point value $Sc(L_k)$ as the current position
3. Done.

4.1.2.4 Summary

FOA fingerprinting is especially designed for opportunistic systems. By eliminating signal strength information from the content of the fingerprints and replacing it through the frequency of appearance, a more stable positioning determination is enabled. Furthermore, created reference-fingerprints can be easily exchanged between different devices, which eases the process of sharing collected fingerprints in a Web 2.0 fashion (crowd-sourcing). UBISPOT is a practical example of an opportunistic onboard/egocentric positioning system using FOA fingerprinting and enabling crowd-sourcing of reference fingerprints. UBISPOT is thoroughly described in Section 4.3, where also an evaluation is given.

Although originally designed for semantic position representation, FOA fingerprints can also contain numerical coordinates. In that case, Algorithm 1 can be extended in the penultimate step, by calculating the center of mass of all position candidates, using the normalized score-points as weights.

The FOA method can be easily extended with more sensors, by simply adding identification information for sensed entities to the fingerprint vector and specifying an according weight. As a matter of fact, LOCATO allows to specify a JAVA-method for each sensor-type, in which more elaborated score functions can be implemented. Furthermore, the algorithms are designed such that they follow the Always Best Positioned paradigm, i.e. they work with any subset of supported sensors.

4.1.3 Geo-Referenced Dynamic Bayesian Networks (geoDBN)

Geo-referenced dynamic Bayesian Networks (geoDBNs) are based on dynamic Bayesian networks, as described in Section 2.6.3. As a matter of fact, the examples given in Sections 2.6.3 and 2.6.3.1 were simplified versions of geoDBNs.

The basic idea of geoDBNs is to use the concept of Bayesian Networks to create a more general model for the behavior of Kalman filters and particle filters. This is accomplished by creating several instances of a generic geoDBN, which describes the characteristics of the used sensors and the according senders, at each possible location and collecting evidences in subsequent time-slices. Instead of creating one huge network that contains all senders installed in the environment, geoDBNs assume that all senders of the same type have the same reliability. This greatly reduces the demand on computational power and memory requirements, which is an important aspect for any system that runs on resource-limited hardware, like a mobile device. Figure 4.4 shows an example of such a generic geo-referenced Bayesian network (the dynamic part will be discussed further below) and its conditional probability tables. This example network uses two different sensors: Sen-

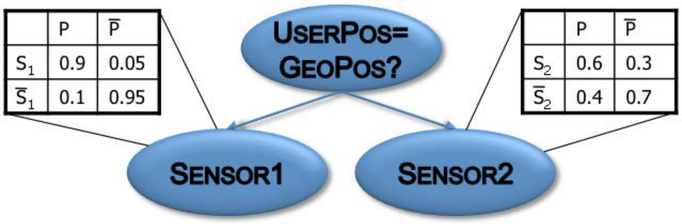


Figure 4.4: Example of a geo-referenced Bayesian network and corresponding conditional probability tables.

sor1 and Sensor2. The network consists of three nodes: the top node, labeled *UserPos*=*GeoPos*, contains two states – *yes* and *no* – indicating whether a user is at position *UserPos* or not. The two lower nodes are sensor nodes, each containing two states as well: *detected* and *not detected*. As stated above, several instances of this network are created at runtime, where each instance represents a possible location of the user. The term geo-referenced stems from the fact, that the top node of the network represents the belief that a user is standing at a specific position, and thus the whole network is geo-referenced to that position (see also [Schwartz et al., 2010a, Schwartz et al., 2005, Brandherm and Schwartz, 2005]).

The basic interpretation of the geoDBN in Figure 4.4 is as follows: if the current user position *UserPos* is the same as the position *GeoPos* of the currently considered geoDBN, then there are certain probabilities that the user's device will sense senders that signal this position. These probabilities, which can be estimated based on tests, are coded in the CPTs of the sensor nodes. The example values in Figure 4.4 are taken from LORIOT. The two states of each sensor node – *detected* and *not detected* – represent the probability whether a sender for Sensor1 or Sensor2 is detected or not. Thus, four basic cases have to be considered:

case P The user is standing at *GeoPos*

case $\bar{P}$ The user is not standing at *GeoPos*

case $S_{1/2}$ The device is detecting the signaling sender for *GeoPos*

case $\bar{S}_{1/2}$ The device is not detecting the signaling sender for *GeoPos*

The CPTs contain probability values for each combination of $a_{1/2}$ and $b_{1/2}$:

Sensor1. A probability of 90% (P, S_1) is assumed that a present sender will be detected by Sensor1, and a 10% (P, $\bar{S}_1$) chance is assumed that a present sender is not detected if the user is at *GeoPos*. The probability that a user who is not at position

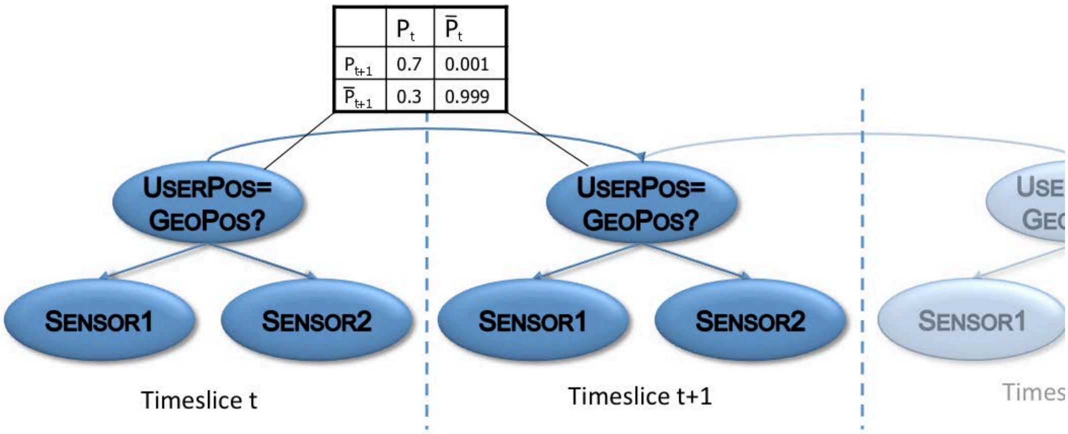


Figure 4.5: Time-slices of a geoDBN and the corresponding conditional probability tables for the transition edge between two time-slices.

GeoPos will nonetheless detect the sender is considered 5% ($\bar{P}$, S_1). A 95% ($\bar{P}$, $\bar{S}_1$) probability is given to the event that the user is not at *GeoPos* and will not detect the sender.

Sensor2. The CPT of the Sensor2 node is interpreted in the same way. For Sensor2 a larger sensing range and higher probability for overreach is assumed. Thus, the probability distribution is more even: 60% (P, S_2) probability to detect an appropriate sender if the user is at *GeoPos* and 40% (P, $\bar{S}_2$) to not detect it. 30% ($\bar{P}$, S_2) probability to detect the sender even if not at *GeoPos* and 70% ($\bar{P}$, $\bar{S}_2$) to not detect it.

The static part of geoDBNs, which was discussed so far, only covers one measurement. If geoDBNs were only static Bayesian networks, previously measured sensor data would have no effect on the calculation of the current position. Through the use of *dynamic* Bayesian networks, the previous position can be taken into account by introducing an edge leading from one time-slice to the subsequent one. Figure 4.5 shows several time-slices of a geoDBN and the CPT assigned to the inter-time-slice edge. This edge is taken into account when the roll-up of the current time-slice is calculated (see Section 2.6.3) and thus influences the calculations of the next time-slice. This CPT models the movement of the user, and thus mimics the prediction stage in a Kalman or particle filter.

According to [Weidmann, 1993] normal walking speed lies in the interval of 0.5 to 2.2 m/s with an average of 1.34 m/s, meaning that a user covers a maximum distance of 2.2 meters in one second. Depending on the sending range of the used senders, not much difference is expected between two subsequent time-slices. These considerations should be taken into account when modeling the probabilities of the CPT. Again four cases have to be considered, taking the current time-slice t and the next time-slice $t+1$ into account:

case $\boldsymbol{P}_\mathbf{t}$ The user is currently standing at *GeoPos* (*UserPos* = *GeoPos* in time-slice t)

case $\bar{\boldsymbol{P}}_\mathbf{t}$ The user is currently not standing at *GeoPos* (*UserPos* $\neq$ *GeoPos* in time-slice t)

case $\boldsymbol{P}_\mathbf{t+1}$ The user will stand at *GeoPos* in the next time-slice (*UserPos* = *GeoPos* in time-slice $t+1$)

case $\bar{\boldsymbol{P}}_\mathbf{t+1}$ The user will not stand at *GeoPos* in the next time-slice (*UserPos* $\neq$ *GeoPos* in time-slice $t+1$)

As in the case of the sensor nodes, the CPT contains all combinations of $P_t/\bar{P}_t$ and $P_{t+1}/\bar{P}_{t+1}$. If a user is at *GeoPos* in the current time-slice, a probability of 70% (P_t, P_{t+1}) is assumed that they will be at the same position in the next time-slice. The probability that they will not be at the same position in the next time-slice is set to 30% (P_t, $\bar{P}_{t+1}$). If a user is not at *GeoPos* in the current time slice, the probability that they will be at *GeoPos* in the next time-slice is set to 0.1% ($\bar{P}_t$, P_{t+1}). The probability that they still will not be at *GeoPos* in the next time-slice is set to 99.9% ($\bar{P}_t$, $\bar{P}_{t+1}$).

Example Calculation

In order to estimate the current user position, the evidences for the sensor nodes of the geoDBN are set according to sensor measurements. The belief of the top-node (*UserPos* = *GeoPos*) is then calculated using standard Bayes inference algorithms.

As an example, assume that a system with two sensors (Sensor1 and Sensor2) that can detect senders of type Sender1 with Sensor1 and of type Sender2 with Sensor2. If the system just detected a previously unseen Sender2 with a specific coordinate, it will instantiate a new geoDBN and the state *detected* will be set to 100% and the state *not detected* will be set to 0% in the Sensor2 node.

If no senders of type Sender1 signaling for the same coordinate were detected in the current and previous time-slides, both states in the Sensor1 node will be left at their a-priori probability since the system cannot decide whether there is a Sender1 present that was just not detected or whether such a Sender1 does not exist . Using the given CPTs, the inference algorithm will result with a belief of 9.52% for the event that *UserPos* = *GeoPos*.

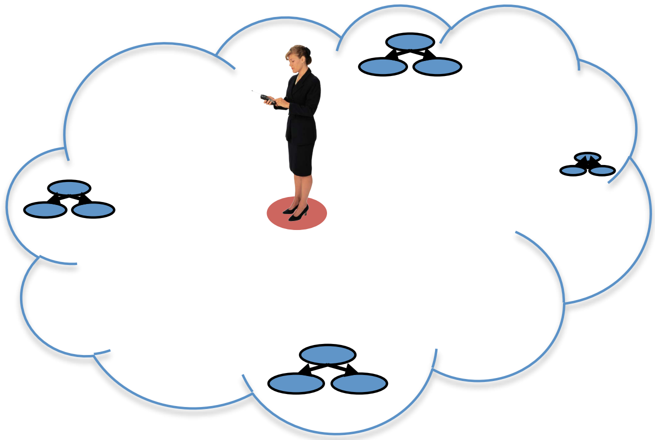


Figure 4.6: A cloud of geoDBNs arises around the user. Each geoDBN has a weight (indicated by the size of each geoDBN in this picture), determined by its belief that the user is standing at its position. The user's position is estimated by calculating the center of mass of the cloud.

If the same Sender2 is measured again in the next time-slice, the states of the Sensor2 nodes will again be set accordingly and the inferred probability for *UserPos* = *GeoPos* rises to 12.66%. If the Sender2 will not be measured again in the third measurement, the state *not detected* will be set to 100% and *detected* to 0%, since the Sender2 has been seen before and thus the system can infer that such a tag exists but was not detected. With the states set accordingly, the belief will drop to 5.32%. On the other hand, if the Sender2 and a Sender1 for the same coordinate is detected, the belief will rise to 77.96%.

In short, if senders are measured repeatedly in subsequent time-slices, the computed probability of the *UserPos*=*GeoPos* node will rise depending on the reliability of the detected senders. It will fall if a sender is not measured again. This resembles the Frequency Of Appearance method of LOCATO, since repeated detections of a sender are indirectly taken as a measure for the distance to the user. Furthermore, the geoDBNs help to smooth out false positives, e.g. overreach of senders, as well as false negatives, e.g. receiving-errors by a sensor.

4.1.3.1 Position Estimation

As already indicated, new geoDBNs are instantiated for each newly detected sender and existing geoDBNs are updated for previously detected ones. Repeated measurements lead to a number of geoDBNs, each giving a probability – or rather a belief – that the user is at the position of the geoDBN. Graphically speaking, a cloud of geoDBNs arises around the user where each geoDBN is a particle of that cloud (as illustrated in Figure 4.6). The weight of each particle is determined by the belief of the respective geoDBN. The estimation of the current user position is calculated as the weighted sum of the coordinates of these particles:

$$\text{UserPos}(t) = \alpha \sum_{i=1}^{n} w(\text{GeoDBN}[i])\ \text{GeoPos}(\text{GeoDBN}[i]) \tag{4.9}$$

Here, n is the number of existing geoDBNs at time t ($n \geq$ the number of received senders at time t), GeoPos(GeoDBN[i]) is the coordinate and w(GeoDBN[i]) the weight of the ith geoDBN. α is a normalization factor that ensures that the sum of all weights multiplied with α is one.

$$\alpha = \frac{1}{\sum_{i=1}^{n} w(\text{GeoDBN}[i])} \tag{4.10}$$

A new estimation of the current position can be calculated after each new measurement. The schematic approach looks like this:

Algorithm 2 Basic Algorithm for the Position Calculation with geoDBNs

1. Perform a new measurement by inquiring all sensors.
2. Obtain the coordinates of each detected sender.
3. Extend every existing geoDBN with a new time slice and cut off the old time slice.
4. Insert the new evidences of the sensors:
    (a) If there is not already a geoDBN at a received coordinate, create a new geoDBN and insert the evidence.
    (b) If there is a geoDBN at a received coordinate, insert the evidence in the current time slice.
5. Go through all geoDBNs and calculate the estimation that the user is at the associated coordinate.
6. Sort the geoDBNs in descending order of their belief.
7. Mark geoDBNs as unused that provide an estimation that is lower than $threshold_{use}$.
8. Calculate the user position by considering only those geoDBNs that provide an estimation above $threshold_{consider}$.

4.1.3.2 Efficient Calculation

Since geoDBNs were designed to run on mobile devices, calculation cost and memory usage are crucial. To reduce both, the number of instantiated geoDBNs must be as low as possible. To achieve this goal, geoDBNs with a weight lower than $threshold_{use}$ are marked as unused (see step 7 in Algorithm 2).

To keep the overhead for memory management low, these unused geoDBNs can be 'recycled' by resetting them to initial values and new coordinates. Furthermore, a maximum number of possible geoDBNs can be specified. If this number is exceeded, those geoDBNs that provide the least estimation will be deleted.

The dynamic Bayesian networks themselves were implemented using a tool called JAVADBN (see [Brandherm, 2006]). This tool provides a graphical user interface to model dynamic Bayesian networks of n_{th} order and automatically generates Java or C++ code for the modeled networks. The resulting code is already optimized regarding computational complexity as well as memory usage and contains inference as well as roll-up algorithms.

4.1.3.3 Summary

GeoDBNs are designed for high accuracy onboard/egocentric positioning systems in instrumented environments following the Always Best Positioned paradigm. As it is the case with FOA fingerprinting, signal strength values can be omitted to provide a higher stability. However, they can also easily be integrated, e.g. by using evidence values for the sensor nodes that are proportional to the measured signal strength.

GeoDBNs resemble particle filters in that they provide hypotheses at different positions, collecting evidence over time. Other than particle filters, the number of hypotheses (particles) is not fixed, but rather depends on the number of sensed senders. Thus, the number of hypotheses automatically adjusts to the environment and through the thresholding in Algorithm 2, hypotheses that are too far away, or left behind, are automatically removed over time.

Besides using numerical coordinates, also semantic descriptions can be used to reference the geoDBNs. In that case, each geoDBN can be seen as a vote for a specific location, where each vote has a weight proportional to the calculated belief. A straightforward way to determine a position is to choose the geoDBN with the highest vote. If a hierarchical location model is used, votes on lower layers can be added on higher layers of the hierarchy. By using a defined threshold, the layer exceeding the threshold can be chosen as the current location.

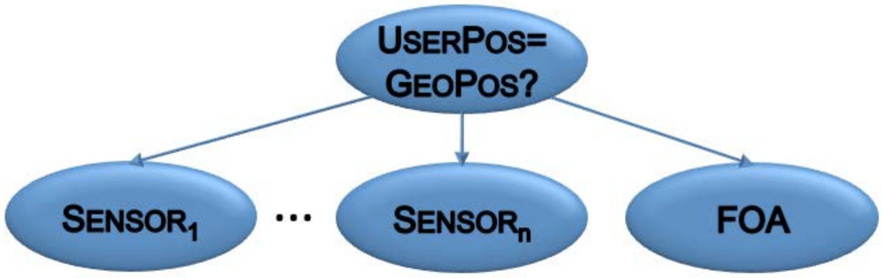


Figure 4.7: FOA systems can be integrated into geoDBN systems by adding a node representing the FOA system.

New sensors can be easily added by inserting a new sensor node and specifying the CPT of that new node. IMU sensors (see Section 3.2) can be easily integrated, by using them to adapt the CPT entries of the inter-time-slice edge between succeeding time-slices.

Any other positioning system can be integrated as a subsystem, also by adding a new sensor node, which represents the subsystem, and specifying the reliability of the subsystem in the CPT of that node. With this method GPS can be easily integrated, but also any other system that is able to derive a compatible position representation. However, systems using trilateration, multilateration or triangulation rely on having a line of sight to enough senders. In situations in which not enough senders can be sensed, such a subsystem will not contribute to the position determination. To overcome this restriction, the sensors of the subsystem themselves can be included in addition, again as sensor nodes. Using this method, even in the case that not enough senders are present, a coarse position-estimation is possible.

With the same approach it is possible to integrate an FOA based positioning system into a geoDBN system. The prerequisite for this is that both systems use the same position representation, i.e. numerical coordinates or semantic descriptions, or that the position representation of the FOA system can be translated into that of the geoDBN system. Figure 4.7 shows an example with n sensor nodes and one FOA node.

LORIOT, which is described in Section 4.4, is an example of an onboard/egocentric positioning system that is built with geoDBNs. LORIOT uses active RFID tags and infrared beacons as senders in indoor environments and GPS for outdoor positioning. The system was also rigorously evaluated, using step-accurate traces as ground-truth (see Section 4.4.4).

4.1.4 External Connections and Tools

4.1.4.1 Blackboard: iROS Event Heap

The name Blackboard-Service is very descriptive. As a matter of fact, such a service works like a blackboard at a public place: people write messages onto it and other people can read them. In conjunction with LOCATO, the iROS Event Heap was used, which was developed at Stanford University by [Johanson et al., 2002, Johanson and Fox, 2002]. The Event Heap is a client-server architecture implemented in Java. The Event Heap server stores and organizes messages as tuples, called events. Clients can connect to the server in order to send and receive events. Single events contain several named standard fields, where some are mandatory and others are optional. The mandatory fields are *EventType*, which is a freely chosen but unique String that describes the type of the event, *SourceID*, which uniquely identifies the sender of the event, and *TimeToLive*, which states a number of milliseconds, after which the event will be deleted. Furthermore, fields can be defined, which contain the actual content of the message. For example, the OUT OF THE BLUE system sends events of the *EventType* "RAWBLUETOOTH", which contains detected Bluetooth addresses in a field called "BTADDRx", and a semantic description of the location of the client in a field called "LOCATION".

Clients can subscribe to events, where filters can be specified to subscribe for specific events, e.g. all events of specific *EventType*, or only events that were sent by a specific sender. Events are stored on the server and distributed as long as the specified *TimeToLive* value was not reached. With this mechanism, clients that connect to the server can still receive events that were sent before the connection. The iROS Event Heap was chosen because of its open architecture, which allows to send arbitrary messages.

4.1.4.2 UBISWORLD

UBISWORLD and its subsystem UBISEARTH were already described in Section 2.4.2.1. LOCATO provides access methods to both services, where the connection to UBISEARTH is used to gain access to the spatial ontology in order to query the ontology, to download parts of it or to modify and extend it. The connection to UBISWORLD is mainly used to update user profiles with the current position of the user. As UBISWORLD also stores old values for each user profile to some extend, a special view was implemented in UBISWORLD, which allows a user to visualize a history of their positions in a so-called film metaphor. Figure 4.8 shows an example of such a position history in the film metaphor.

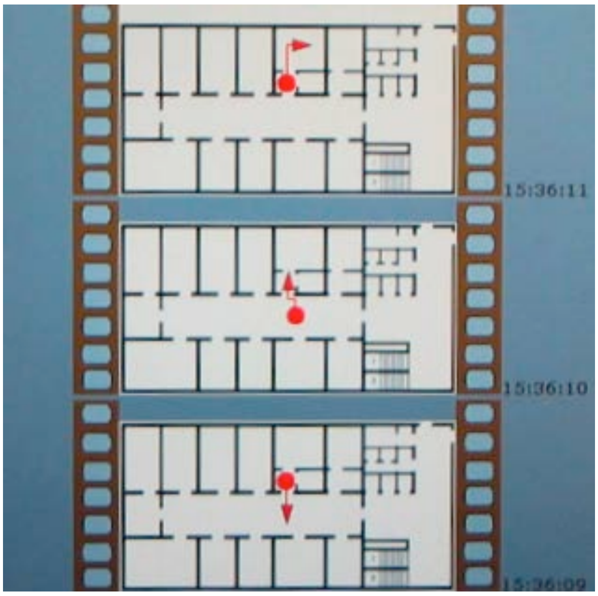


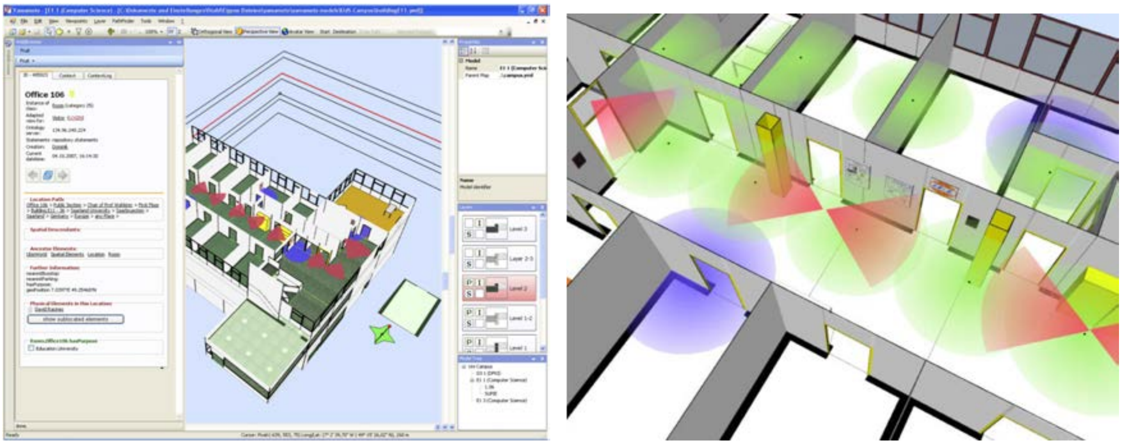
Figure 4.8: UBISWORLD provides a film-metaphor view of the history of positions of a user.

4.1.4.3 YAMAMOTO

In order to use the geoDBN core algorithm of LOCATO, the environment has to be instrumented with senders. The coordinates of these senders have to be determined and stored somewhere. For example, in the case of LORIOT, the coordinates are stored directly onto active RFID tags. However, the deployment of senders has to be planned according to the environment a positioning system should be used in. Having a detailed model of the environment provides great help to plan the needed infrastructure.

In [Stahl and Haupert, 2006], YAMAMOTO (Yet Another MAp MOdeling TOolkit) was introduced as a toolkit to quickly and efficiently create such detailed models of multi-story buildings. To model a building, an architectural floor plan is used as backdrop image and the outlines of rooms and corridors are manually traced, leading to a 2D representation of each story represented by vertices and edges.

By marking edges as being doors, windows or walls the 2D model can be extended to a so-called 2.5D model, which allows vertical arranging of multiple stories. Associating semantic attributes, like *not passable*, *passable for pedestrians* or *passable for wheelchairs* to edges, allows for user-adapted route finding and planning. Figure 4.9a shows a screen shot of YAMAMOTO during the modeling process of building E11 at Saarland University.



(a) Screenshot of YAMAMOTO

(b) IR beacons and active RFID tags modeled with YAMAMOTO

Figure 4.9: YAMAMOTO can be used to model a building and plan the positioning infrastructure. Shown are IR beacons (red), active RFID tags (green) and Bluetooth beacons (blue).

YAMAMOTO was extended to represent different senders, like infrared beacons, RFID tags, WiFi access points or Bluetooth beacons ([Stahl and Schwartz, 2010]).

Three basic primitives are used to model various senders: *Point (x, y)* for senders that should be modeled without taking their range into account, like WiFi access points with unknown sending range, *Circle(x, y, radius)* for senders with radial sending characteristics, like active RFID tags, and *Section(x, y, radius, beam angle, orientation)* for directional senders, like IR beacons.

Each primitive can also be associated with a symbolic name and their sending ID. Figure 4.9b shows the model of the lab of Prof. Wahlster at Saarland University, including the LORIOT instrumentation.

A model derived with YAMAMOTO can also be geo-referenced to known points or areal photographs. In the latter case, the outline of the YAMAMOTO model can be manually scaled and aligned to fit into the respective area of the building on an areal photograph with known geo-references (as outlined in Figure 4.10).

YAMAMOTO automatically derives the needed scaling and rotation matrices to convert its internal coordinates into the coordinate system of the areal photograph. WGS84 compliant coordinates can thus be derived for every point inside the modeled building. If an instrumentation was planned with YAMAMOTO, the resulting list of senders can be exported to an XML format, called YML for Yamamoto Modeling Language, which includes the coordinates of each sender as well as their ID.

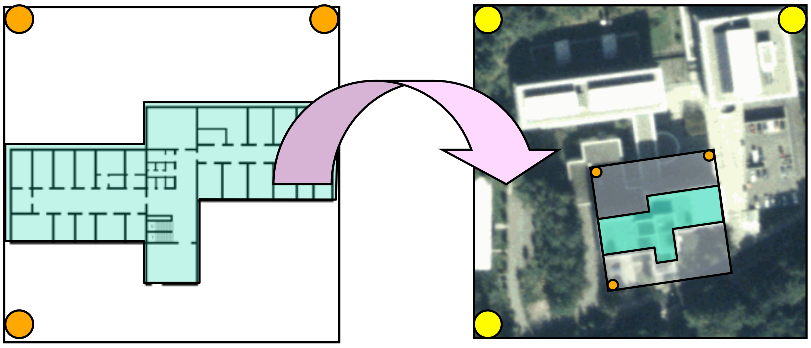


Figure 4.10: YAMAMOTO allows to geo-reference a model by rescaling, resizing and placing it into an already geo-referenced areal photography.

Simulation and Visualization with Yamamoto

As YAMAMOTO also has an interface to the blackboard service, it can also be used to simulate the proximity detection core algorithm. An avatar can be freely moved inside the modeled building, either in a birds-eye's view or in an egocentric perspective, as known from 3D computer games. If the avatar's coordinates are inside of the range of a modeled sensor with radial characteristics, the same events are sent to the blackboard that would be sent in a real world deployment by the actual sensor. With this simulation, services can be tested prior to deploying the sensors in the real environment.

Furthermore, a TCP/IP socket connection is provided, which realizes the so-called Yamamoto Control Interface (YCI). The YCI is bidirectional, i.e. positions can be received and sent out. A positioning system can connect to the socket and provide determined coordinates of a user. The avatar will then automatically be placed at the determined coordinates. This communication direction can be used to visualize determined positions. The other direction, i.e. sending out the current coordinates of the user controlled avatar, can again be used to pretest location based services, prior to the deployment of the infrastructure. More features and practical applications of YAMAMOTO, like activity modeling or route finding, are thoroughly described in [Stahl, 2009].

4.1.5 Summary

LOCATO provides three core algorithms. One that can be used to design offboard-/exocentric positioning systems, and two newly developed core algorithms to design onboard/egocentric positioning systems that follow the Always Best Positioned paradigm. Of these two core algorithms, FOA is designed for opportunistic positioning systems, while geoDBN is designed for positioning systems in designed instrumentations. The algorithms are optimized for low computational complexity and can thus be executed on mobile devices with restricted resources. As explained in Section 2.3.2.5, onboard positioning systems can be easily converted into offboard positioning systems. Thus, these two core algorithms can also be used to design offboard/egocentric systems.

Furthermore, LOCATO provides access methods to the cloud service UBISWORLD, including its subsystem UBISEARTH, and access methods to a local infrastructure blackboard service, the iROS Event Heap. Both services can be used provide additional services.

The external modeling toolkit YAMAMOTO can be used to design instrumentations as well as to test the instrumentations prior to deployment, and to visualize the output of a deployed positioning system.

The next three sections present example positioning-systems that were implemented using LOCATO.

4.2 OUT OF THE BLUE: A Bluetooth-based Offboard/Exocentric Positioning System

OUT OF THE BLUE was designed as a very simplistic but also very coarse-grained *exocentric* positioning system. It uses Bluetooth technology and its main advantage lies in the fact that users do not have to purchase any new device as long as they already own a Bluetooth enabled nomadic device. On the environment side, any stationary Bluetooth-enabled device that is capable of running Java and provides access to the Bluetooth stack via Java can be used to detect the presence of users. OUT OF THE BLUE is the only offboard/exocentric positioning system developed in this thesis.

4.2.1 Hardware

4.2.1.1 Nomadic Device

OUT OF THE BLUE was designed to work with any Bluetooth enabled mobile device. The sheer Bluetooth capability is enough, since no additional software has to be

[1] http://www.bluenio.com
[2] http://www.gearfuse.com/bluebird-keeps-an-eye-on-your-luggage/

(a) Wireless headset (b) nio™ Bluetooth tag[1] (c) Bluebird luggage tag[2] (d) Sony Ericsson T660

Figure 4.11: OUT OF THE BLUE is not limited to smartphones on the user's side. Any Bluetooth enabled device is usable as positioning tag, including so-called 'dumbphones'.

installed on the user's device. In that sense, a minimalistic Bluetooth circuit that is only capable of receiving standard scanning requests and sending out an appropriate answer could be used as positioning tag. Thus, the range of devices is not limited to smartphones and the like, but also includes low-cost wireless headsets or special Bluetooth tags as shown in Figure 4.11.

4.2.1.2 Senders and Sensors

OUT OF THE BLUE uses Bluetooth for the instrumentation of the environment as well as on the user-device side. Since Bluetooth relies on bidirectional communication, the basic technology is the same on both sides. There is however a difference in the needed computing power.

As stated above, a minimalistic Bluetooth circuit is sufficient on the user side. On the instrumentation side, at least the ability to process the information gained from periodically scans is needed. To take full advantage of the system, there should also be a means to exchange this information with other devices on the instrumentation side.

In an office setting, the office workers' desktop computers can be used to host the OUT OF THE BLUE software client. Usually, public displays also provide data connectivity as well as computational power and thus are perfect devices on the instrumentation side. In both cases – desktop computers or public displays – the needed Bluetooth capability can be retrofitted using USB dongles.

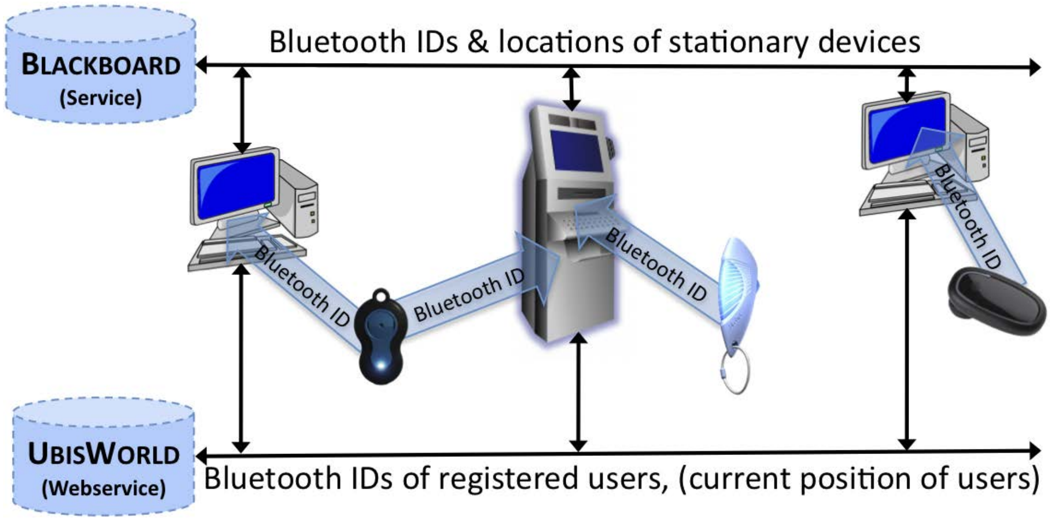


Figure 4.12: OUT OF THE BLUE consists of stationary devices, like desktop PCs, public displays or kiosk systems that scan their environment for mobile Bluetooth devices, like cell phones, wireless headsets or Bluetooth tags. Information can be shared via a blackboard service and registered users can store their Bluetooth IDs in UBISWORLD.

4.2.2 Methods

4.2.2.1 Proximity Detection

As described in Section 3.1.4, every Bluetooth enabled device has a unique numeric address. Two Bluetooth devices that want to exchange data need to know each other's addresses. Since Bluetooth was also designed for ad-hoc communication, a Bluetooth device can scan its surroundings for other Bluetooth devices or services. As a result it gets a list of addresses of all Bluetooth devices that are willing to share that information and that are in a close enough range.

The main idea behind OUT OF THE BLUE is that this mechanism provides a simple means to implement proximity detection: Stationary Bluetooth enabled devices, such as desktop PCs or panel PCs used as public displays, can periodically scan their environment for other Bluetooth devices. If such a stationary device has knowledge about the range of its own Bluetooth transceiver, it can derive which other Bluetooth devices are in that range. Given additional information, i.e. the Bluetooth address of the mobile device of a certain user, it can infer whether that user is currently in its vicinity.

Figure 4.12 shows an example setup of OUT OF THE BLUE. In such a setting, a public kiosk system at the entrance of a building could provide users the option to register with the system. For ease of use, users can place their Bluetooth enabled device in a shielded box at the kiosk system, so that the Bluetooth ID of that particular device can be determined. If the user already has an UBISWORLD account, they can give permission to certain parts of their user profile or they can register for a specific service, e.g. route guidance to their destination. Along the way of the user, public displays detect their Bluetooth ID and can then adapt their presentation to the user's profile or their requested service.

OUT OF THE BLUE runs on the aforementioned stationary devices and uses the proximity detection core-algorithm of LOCATO. It runs in the background and continually scans the environment in a freely adjustable time interval. Such an OUT OF THE BLUE sensor-node can operate in two modes: *isolated* or *sharing*. In *isolated* mode, a node keeps all information gained through the scanning process by itself, which is usually suitable for a public display to adapt its presentation to the number of users in its vicinity or to show user specific information. In *sharing* mode, the resulting list of mobile devices is sent to the blackboard system (see Section 4.1.4.1), via the interfaces provided by LOCATO. Each stationary device can therefore sense the presence of registered users in their direct vicinity, but in the *sharing* mode they can also gain knowledge about users further away and can try to reason over this, e.g. to estimate the walking direction.

The accuracy of the position depends on the range of the stationary Bluetooth devices and on the range of the user's mobile device. Most Bluetooth dongles available for PCs are Class 1 or Class 2, resulting in a range between 10 and 100 meters if no obstacles attenuate the signals. This range is usually decreased through walls, doors and furniture indoors.

Since most modern mobile phones are Bluetooth enabled, this system is readily available to a broad public. The missing feedback to the user about their own position however limits its application to adequate services, like public displays adapting their content to the nearby users or user sensitive self-opening doors. An example service that was realized with OUT OF THE BLUE is given in Section 5.5.

4.2.3 Summary

OUT OF THE BLUE is an offboard/exocentric positioning system using a single sensor technology. With regard to the design criteria for positioning systems, it was designed to minimize the cost of ownership for the users. As most mobile phones are Bluetooth capable and simple Bluetooth devices – like wireless headsets – are available at low cost, Bluetooth was chosen as sensor technology. As the system was designed for use with public displays, the cost of ownership for the positioning system is low compared to the costs of the public displays themselves. The low costs of ownership is accompanied by a low accuracy, which is in the range of tens of meters, a low robustness, which is due to long inquiry times of the Bluetooth sensors, and a low privacy protection, due to the offboard/exocentric approach. In terms of usability and applicability, OUT OF THE BLUE is lightweight and small in size as it can be used with any Bluetooth capable mobile phone. Due to the repeated Bluetooth inquiries, which have to be answered by the user's mobile device, the battery consumption is increased. However, users who do not want to use the system can either switch off Bluetooth or set their device into non-discovery mode to prevent higher battery consumption.

4.3 UBISPOT: An Opportunistic Onboard/Egocentric Positioning System

UBISPOT is an onboard/egocentric *opportunistic* positioning system, as described in Section 2.3.3. The system uses standard, built-in sensors of modern mobile phones to detect cells, WiFi access points and Bluetooth devices. For outdoor positioning, GPS is also taken into account.

UBISPOT can reach room level accuracy inside buildings and uses semantic descriptions based on the spatial ontology of UBISEARTH rather than numerical coordinates. Furthermore, UBISPOT does not only tackle the problem of determining the current location of a mobile device, but also how a database containing locations and measurements can be established. This database can be privately created by each user to contain only their locations of interests or by sharing these entries in a Web 2.0 fashion via UBISWORLD, of which UBISEARTH is a part ((see Section 2.4.2.1)).

UBISPOT was designed for context aware applications and services that do not rely on a meter or sub-meter level accuracy, but rather on the current area. One such application is directly integrated into the system: automatic ring tone switching when entering or leaving specified locations. Through the use of the spatial ontology, also the category of rooms in which the phone should be muted can be given instead, e.g. conference rooms or lecture halls.

The position determination is accomplished using the FOA fingerprinting algorithms of LOCATO. UBISPOT was rigorously evaluated in an environment with small rooms but high-density infrastructure of WiFi access points and Bluetooth devices, to determine the highest possible accuracy. The infrastructure was used as is, i.e. no additional WiFi access points or Bluetooth devices were introduced into the environment.

Figure 4.13: Sensors and senders of UBISPOT: GSM/UMTS cells, WiFi access points and Bluetooth devices act as senders. For outdoor positioning, GPS is used in addition.

4.3.1 Hardware

4.3.1.1 Senders and Sensors

UBISPOT uses the most common built-in data transmission transceivers of modern cell phones:

- **GSM/UMTS** commonly used for telephony and data transmission
- **WiFi commonly** used for Internet access
- **Bluetooth** commonly used for short-range voice/data transmission, e.g. wireless headset or exchange of contact information

Although these technologies are able to communicate in both directions, for the use in positioning UBISPOT merely uses them as sensors. Figure 4.13 shows the respective senders:

- **Cell Towers** broadcast a unique ID
- **WiFi Access Points** broadcast a unique MAC address
- **Bluetooth Devices** identify themselves with a unique Bluetooth ID

As described in Section 2.3.3, these senders are not installed into the environment with the purpose of positioning – they are already there, for either communication (cell towers, WiFi Access Points, Bluetooth) or to replace wires between devices and additional hardware (Bluetooth).

4.3.1.2 Mobile Devices

UBISPOT was implemented for two mobile operating systems: Symbian OS v9.x Platform S60 and Google Android. As testing platforms three mobile phones were used: a Nokia E60, a Google/HTC Nexus One and a Google/Samsung Nexus S. The Nokia E60 provides GSM Tri-Band and UMTS Single-Band capabilities and has integrated WiFi and Bluetooth transceivers. The mobile phone does not have a built-in GPS device, so a wireless Bluetooth GPS-receiver (Holux GPSlim 236) was attached for outdoor positioning. The E60's ARM processor runs with only 220 MHz, so it is the optimal platform to test the computational efficiency of the FOA algorithm.

In addition to the built-in WiFi and Bluetooth transceivers, the Nexus One and Nexus S phones have integrated GPS receivers and both provide GSM Tri-Band and UMTS Tri-Band capabilities. The larger displays and the touch-screen functionality of both devices allow a better user-interface and usability. Furthermore, both Android devices run with a clock speed of 1 GHz. Figure 4.13 shows the Nexus One with running UBISPOT client as well as the senders that are used for positioning. UBISPOT was tested on Android Version 2.3 (Gingerbread) and Version 4.0 (Ice Cream Sandwich).

4.3.2 Methods

4.3.2.1 Frequency-Of-Appearance Fingerprinting

UBISPOT is based on the Frequency-Of-Appearance fingerprinting algorithm of LOCATO, as described in Section 4.1.2. UBISPOT fingerprints contain a list of all sensed cell data, WiFi access point MAC (Media Access Control) addresses, Bluetooth device IDs and the measured latitude and longitude of the GPS receiver, if available. Table 4.1 shows which data are captured for each sensor. Cells provide the most data, besides a unique cell ID, they also provide IDs for the local area, the country and the network. As mentioned in Section 4.1.2.3, UBISPOT uses the cell data to provide a pre-filtered list of candidates for the position determination through ranking. As cells are available mostly everywhere and cover a large range, they are the optimal selection for this task.

Fingerprints are stored along with the semantic description of the position in which the measurements were taken in an XML format.

GPS	GSM	WiFi Access Point	Bluetooth Device
Longitude	Cell ID	MAC Address	Bluetooth ID
Latitude	Local Area ID		
	Country ID		
	Network ID		

Table 4.1: The collected data for each sensor in UbiSpot. Each Snapshot can contain several instances of GSM, WiFi Access Point and Bluetooth Device.

4.3.2.2 Building and Sharing the Database

UBISPOT constantly calculates FOA fingerprints with the optimized FOA method described in Section 4.1.2. In order for users to train their system on a specific location, they indicate this by choosing the appropriate menu entry. UBISPOT uses the hierarchical location model of UBISEARTH to denominate the determined position (see also Section 2.4.2.1). The topmost hierarchy consists of Continent → Country → Region → City → Building. Moreover, users can share fingerprints via UBISWORLD. If a user enters a new area for which UBISPOT does not already provide fingerprints, it tries to download them via UBISWORLD. In order to choose the correct subtree of the spatial ontology in UBISEARTH, the currently sensed cell data is used. If no fingerprint data is available, users can train the system themselves. Moreover, users can add new nodes to represent different floors and specific areas or rooms of a building.

In order to do so, users can browse to their current location and refine the model to their needs. Figure 4.14 shows how a user browses to the 'Chair of Prof. Wahlster' and adds a new location named 'Office 118'. This new entry is then stored on the mobile phone together with the current FOA fingerprint. As can be seen in Figure 4.14e, each entry is also marked with a symbol: the orange dot denotes that an entry is either a parent node or an untrained child node. A green star shows that a child node's fingerprints are up to date (at most seven days old), a yellow star shows that it is between seven and fourteen days old and a red star denotes that the fingerprint is

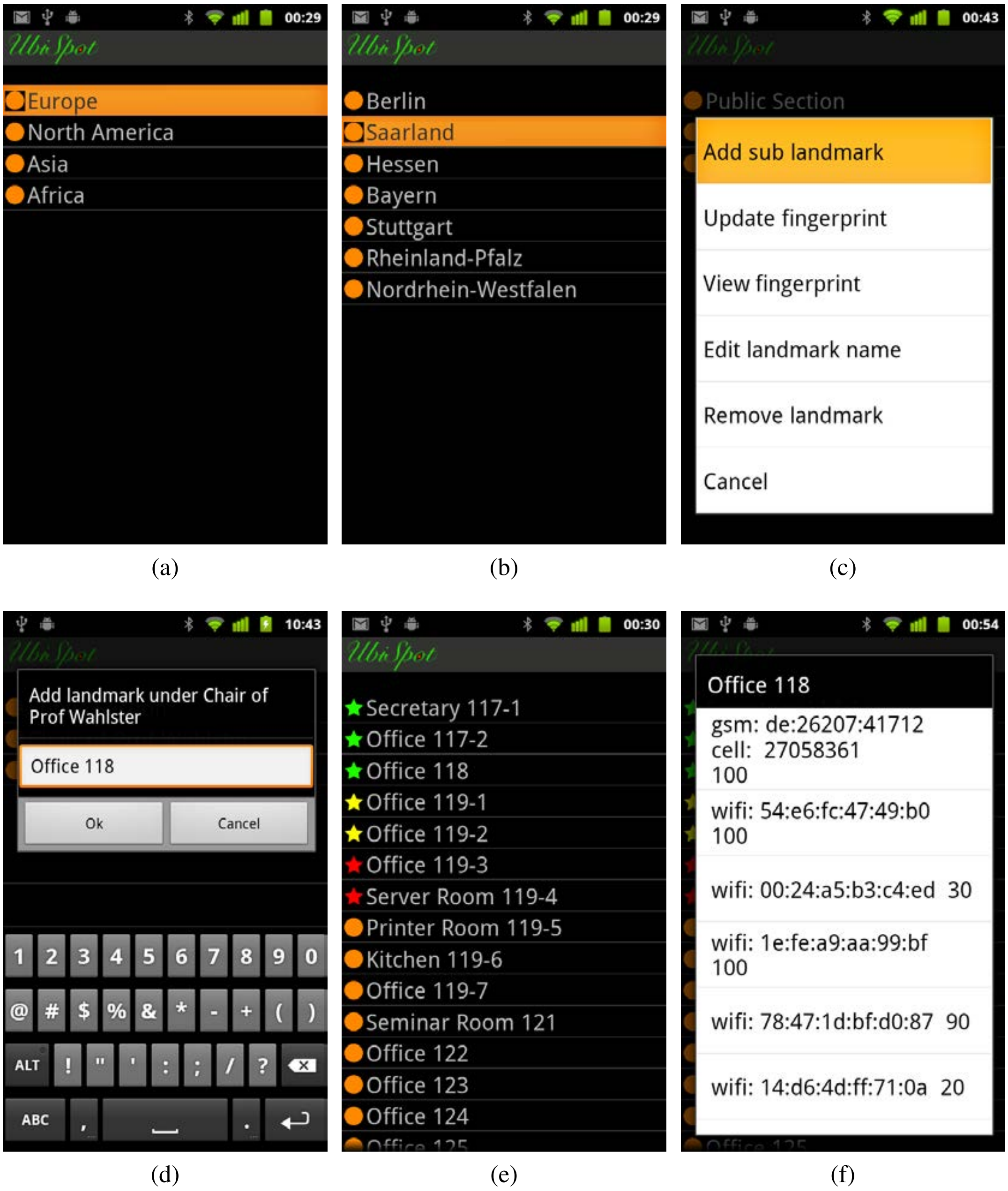
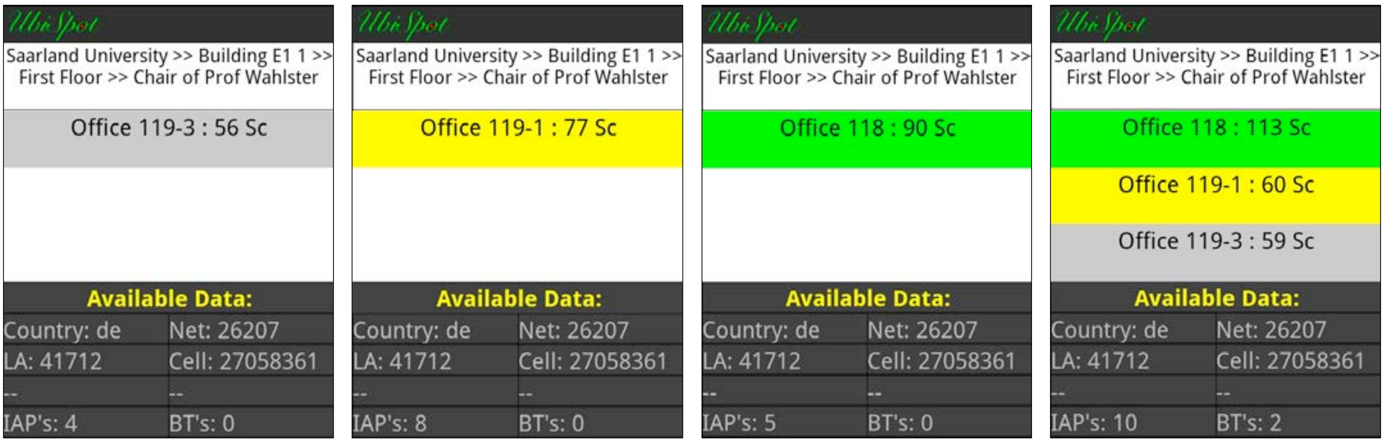


Figure 4.14: Training of a new location in UBISPOT: A user browses through the imported UBISEARTH location model and refines it by a new location. This location is stored with the current fingerprint.



(a) (b) (c) (d)

Figure 4.15: Output of the UI of UBISPOT: The confidence level of the system is color-coded: green for highest confidence, yellow for middle and gray for lowest confidence.

older than fourteen days. If UBISPOT finds a data-connection, it automatically tries to update old fingerprints for the current area via UBISEARTH. In case that no new fingerprints are available, a user can choose to update them while being at location and upload them to UBISEARTH. Figure 4.14e shows the details of a fingerprint that was taken in Office 118. The type of each sensed sender is shown (Cell, WiFi, Bluetooth) as well as the obtained ID. The number after the ID shows the percentage of how often the according sender was observed.

4.3.3 Output to the User

After UBISPOT has determined the score points for the current measurement, it presents the user its current position estimation. To depict the level of confidence – based on the derived score-points – a color coding is used:

1. Green, if the derived score-points for the location are above 85. This indicates that the system is certain that its estimation is correct.

2. Yellow, if the derived score-points for the location are between 75 and 84.

3. Gray, if the derived score-points for the location are between 45 and 74.

Examples for this color-coded output can be seen in Figure 4.15. Besides the name of the current room, the overlying parts of the location hierarchy are also shown.

The direct parent and its predecessor node are shown directly above the room name (in these examples the Chair of Prof. Wahlster and the floor number). The other ancestor nodes are shown as a scrolling text (Germany >> Saarland >> Saarbrücken >> Saarland University >> Building E1 1). A statistical overview on the current fingerprint is shown at the bottom of the screen. It contains the county ID (Country), network ID (Net), land area ID (LA) and number (Cell) of the strongest available cell as well as the number of currently sensed WiFi access points (IAP's) and the number of detected Bluetooth devices (BT's). If a good GPS reception is given, the reported latitude and longitude will also be shown.

4.3.4 Evaluation

From the design of UBISPOT, a list of extrinsic and intrinsic factors that obviously influence the accuracy of the position estimation can be derived:

- available instrumentation (extrinsic)
  - number of cell towers
  - number of WiFi hotspots
  - number of Bluetooth devices
- size of rooms or density of trained landmarks (extrinsic)
- number of trained landmarks in the system (intrinsic)
- $TimeWindowSize$ used for the FOA fingerprinting (intrinsic)

The influence of single senders on UBISPOT can also be derived from the system's design: if only a single cell tower is available, then UBISPOT will only be able to derive a rough location, e.g. Saarland University Campus. With only a single WiFi hotspot, the derived area will be smaller, e.g. Building E11. The same holds for a single detected Bluetooth device, only that the detection range will usually be much smaller. Regarding the extrinsic factors, the most interesting question is therefore if UBISPOT can achieve room level accuracy in a highly instrumented environment but with a high density of trained landmarks, i.e. small rooms. Since the lab of Prof. Wahlster in the computer science building of Saarland University provides such an environment, it was chosen as a test field for the evaluation.

The most interesting intrinsic factor is the used $TimeWindowSize$, since this parameter can easily be modified. Since a larger $TimeWindowSize$ contains more information, it is expected that the accuracy rises with larger values for this parameter. On the other hand, a large $TimeWindowSize$ also delays the detection of a position change.

With respect to these considerations the evaluation should answer the following questions:

1. How does $TimeWindowSize$ influence the accuracy of UBISPOT?
2. Which $TimeWindowSize$ gives a good trade-off between accuracy of position estimation and delay of a position change?
3. How close are false recognitions to the true location?
4. Do detected Bluetooth devices decrease or increase the accuracy?
5. How does the accuracy change according to the number of trained landmarks?

Since UBISPOT gives out location names rather than coordinates, accuracy here means how often the determined location name is equal to the real location name.

4.3.4.1 Evaluation Design

As indicated above, the lab of Prof. Wahlster provides an ideal testbed for evaluating UBISPOT, since university campus is well equipped with WiFi hotspots and Bluetooth devices are scattered over the offices in form of Bluetooth mice and Bluetooth enabled cell-phones. The floor plan of the lab is shown in Figure 4.16, the dots indicate learned landmarks. 19 landmarks were learned in this environment, including two corridors and stairways. Four additional landmarks were learned in the attached building E13, so that a total of 23 landmarks were stored in the system. The number of trained landmarks has a direct influence on the a-priori probability of guessing the right room, i.e. the probability that a randomly chosen room is the right one. In the used test environment with 23 learned landmarks, this a-priori probability is $\frac{1}{23} \approx 4.35\%$.

No instrumentation was added to the already existing WiFi hotspots and Bluetooth devices. The evaluation was conducted using three Nokia E60 cell phones.

Regarding the variation of the $TimeWindowSize$ parameter, the naïve approach would be to bring the mobile phone in different locations and have it logging its determined positions with varied $TimeWindowSize$. However, this method would consume a high amount of time: polling all required sensory data of the Nokia E60 to calculate one snapshot takes 8 seconds. To create a fingerprint with

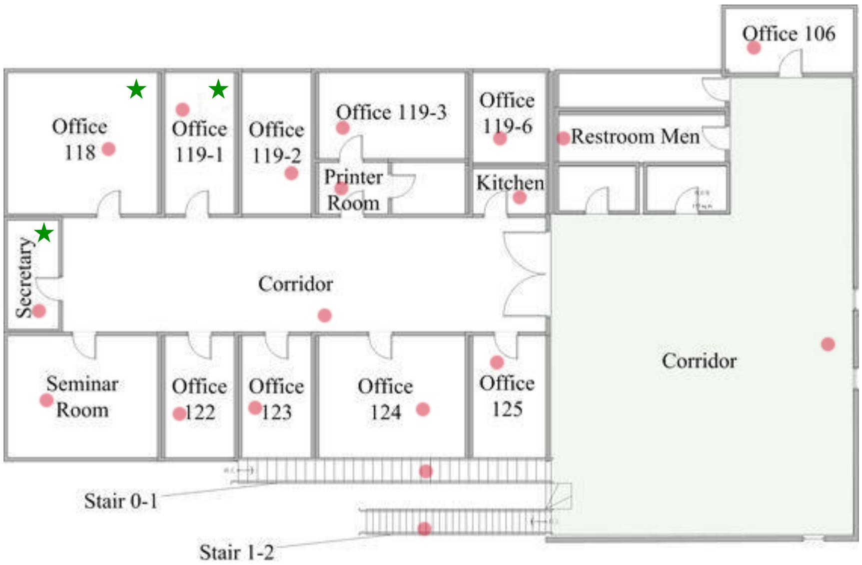


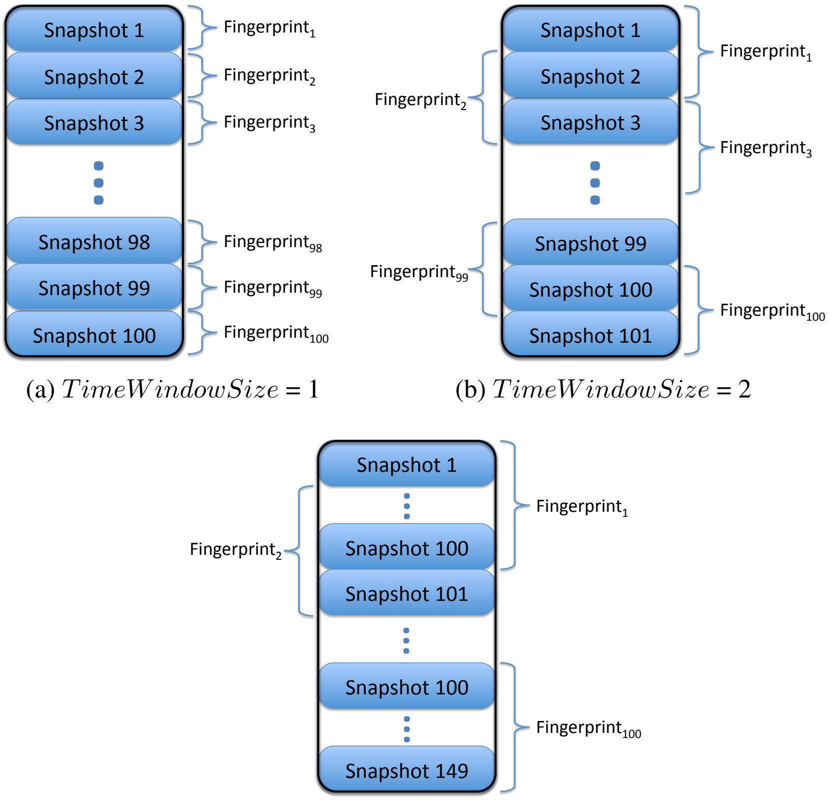
Figure 4.16: Floor plan of the evaluation environment, located in the first floor of computer science building $E1_1$ of Saarland University, Germany. Red dots indicate rooms that were trained as landmarks, green stars indicate the tested rooms.

$TimeWindowSize = tws$, a number of tws snapshots has to be taken. Since one determined location would not be enough to deduce a meaningful accuracy statistic, this fingerprinting has to be repeated for the desired number of samples, e.g. 100 times. Because each fingerprint is derived by using a FIFO, a total of $n - 1 + tws$ snapshots is needed to calculate n fingerprints (see Figure 4.17). It follows that varying the $TimeWindowSize$ from 1 to 50 at a single location and taking 100 samples for each value of $TimeWindowSize$ would need

$$\sum_{tws=1}^{50} (99 + tws) * 8s = 49800s = 13 \text{ hours } 50 \text{ minutes} \tag{4.11}$$

Besides the timely effort to collect the data, this approach could also compromise the evaluation. It is important to keep in mind that for the evaluation the question 'What would the system's output be, if it would use a different $TimeWindowSize$ in the otherwise exact same situation?' has to be answered. If the infrastructure changes throughout these approximately 14 hours, e.g. some WiFi hotspots or Bluetooth devices get switched off during the measurement of some higher $TimeWindowSize$, it would give the impression that a lower $TimeWindowSize$ performs better.

Fortunately, it is sufficient to collect 149 snapshots at one location – which takes $149 * 8\text{s} = 19$ minutes 52 seconds – and then recalculate the FIFO outputs for all varying $TimeWindowSizes$ as depicted in Figure 4.17. This method minimizes the



(a) $TimeWindowSize = 1$ (b) $TimeWindowSize = 2$

(c) $TimeWindowSize = 50$

Figure 4.17: The calculation of 100 FOA fingerprints with different $TimeWindowSize$. Since the fingerprints are calculated by using a FIFO, $99 + TimeWindowSize$ snapshots are needed.

probability of getting different base measurements for the evaluation and thus allows to reproduce as exactly as possible which position would be derived by UBISPOT with a specific $TimeWindowSize$. A similar argumentation holds for collecting the measurements in different locations: The question to be answered here is 'What would the system's output be, if the measurements were taken at a different location?'. If data for different locations are collected in succession, there is a high risk that the infrastructure has changed in the meantime. Therefore, the measurements were taken simultaneously in three adjacent rooms (marked with stars in Figure 4.16) using three Nokia E60 devices.

Of course in a real world setting the infrastructure *will* change over time, especially the availability of Bluetooth devices. This effect is still taken care of in the evaluation, because the training of the system was done two weeks prior to taking the evaluation measurements. Moreover, the system's training data were collected with only one mobile phone and thus during different times of the day and over the course of two days.

To perform the actual analysis, the core positioning-algorithm of UbiSpot was re-implemented in Java and complemented with an algorithm to calculate fingerprints with different $TimeWindowSize$ using exact real life measurements collected with the Nokia E60 devices. This software – named UbiSpot Sim [Ji, 2011] – runs on a standard desktop PC and automatically calculates hit and miss statistics.

To summarize, the evaluation was done in three phases:

Phase 1: Train UbiSpot for 23 landmarks. This phase started two weeks before phase two on different times of day and over the course of two days.

Phase 2: Collect ≈ 150 real life measurements in three adjacent rooms using three Nokia E60 devices. For training and measuring, the devices were placed in the geometrical middle of each room. This step took ≈ 20 minutes.

Phase 3: Perform the evaluation off-line using the real life measurements collected in phase two and the trained landmark database from phase one.

4.3.4.2 Results

Figure 4.18 shows the measured accuracy plotted against the $TimeWindowSize$. A position estimation was count as hit, iff the derived room coincided with the room where the measurements were taken in and as miss otherwise. Therefore, the plotted accuracy represents the number of correct estimations out of the total number of estimations for the given $TimeWindowSize$ according to the formula

$$\frac{\text{number of hits}}{\text{total number of estimations}}.$$

Dashed lines in the graph represent position estimations without taking Bluetooth into account, solid lines depict estimations including Bluetooth.

How does the TimeWindowSize influence the accuracy of UbiSpot? In respect of the first question the graph shows that the accuracy rises with increasing $TimeWindowSize$ for most of the tested rooms and conditions (Bluetooth on or off), except for the estimations derived for room 119-1 without Bluetooth. Room 119-1 without Bluetooth is obviously an under-performer in comparison to the rest and has to be examined closer.

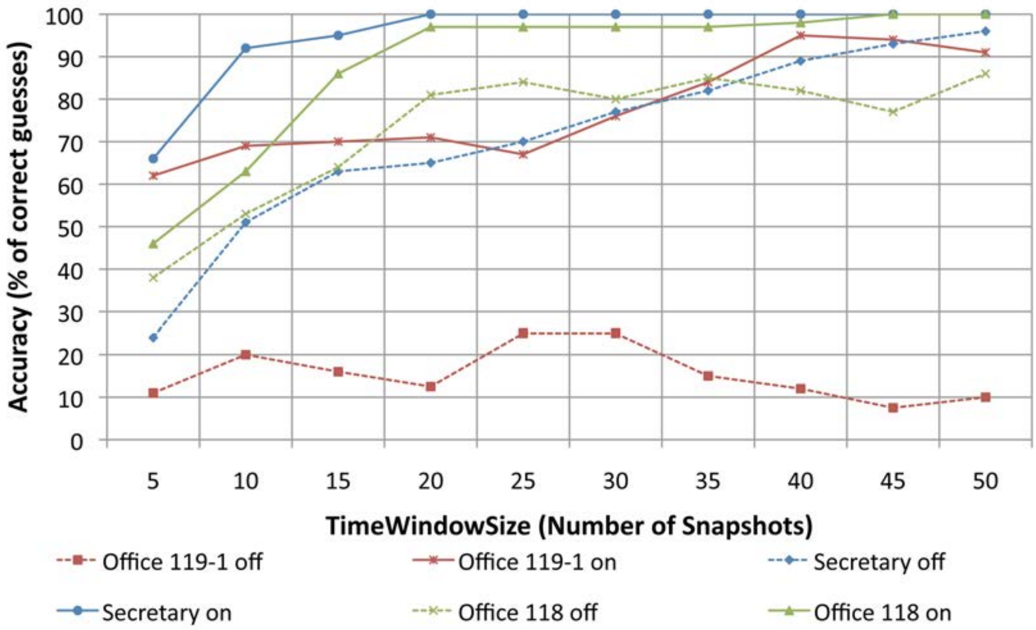


Figure 4.18: Recognition accuracy versus $TimeWindowSize$ for three adjacent rooms. Solid lines indicate results including all available sensor data, dashed lines indicate the results when Bluetooth is not considered for the position calculation.

Do detected Bluetooth devices decrease or increase the accuracy? The graph shows that those position estimations that are considering Bluetooth devices are in all cases better than those without Bluetooth. This result is surprising, since most of the detected Bluetooth devices were mobile phones and as such can easily change their position. The fact that they still contribute in a positive way to the position estimation maybe due to the owners of the phones being in their respective offices most of the time. Most obviously, including Bluetooth helps to detect room 119-1, bringing the accuracy from 7.5% in the worst case ($TimeWindowSize$ of 45) up to 94% in the best case ($TimeWindowSize$ of 40).

Which TimeWindowSize gives a good trade-off between accuracy of position estimation and delay of a position change? A higher $TimeWindowSize$ means a longer time delay until a new position estimation has stabilized. Considering the graph in Figure 4.18, a $TimeWindowSize$ of 25 seems to provide a good trade-off since three of the six locations do not gain in accuracy when rising the $TimeWindowSize$ further. A value of 25 means a 200 second delay until a new position has stabilized after changing the location.

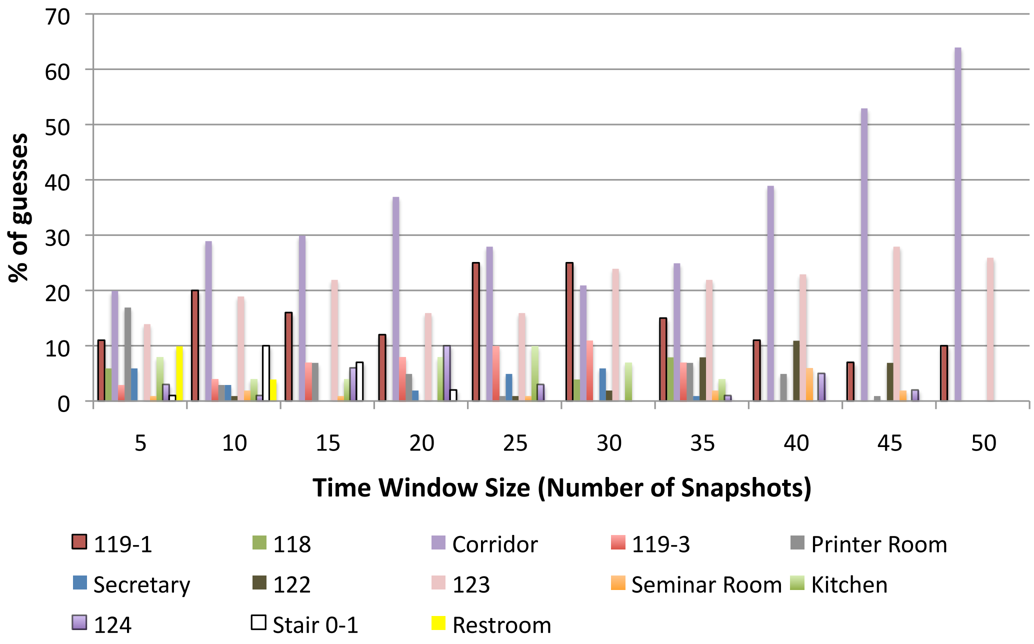


Figure 4.19: Hits and misses for office 119-1 not considering Bluetooth devices. Most of the time UBISPOT mistakes the office 119-1 for the corridor.

How close are incorrectly classified positions to their true position? Figure 4.19 shows how room 119-1 was classified for each $TimeWindowSize$ without taking Bluetooth devices into account. The vertical axis shows the percentage of guesses for each classified room, e.g. at $TimeWindowSize$ 20, room 119-1 was classified as corridor in 37% of all measurements, and as room 123 in 16%. The bars are ordered in distance to the actual location. Locations that do not appear in the graph were never classified as being the current location. It can be seen that most of the time room 119-1 was mistaken for the lab's corridor, which is still close to the actual location. Far away locations, like the restroom, only appear with low $TimeWindowSize$. None of the additional four landmarks of Building E1 3 ever appeared as classification for room 119-1. From this graph, it is also apparent that the number of confusions drastically reduces with higher $TimeWindowSize$, e.g. only three locations are considered by the system with a $TimeWindowSize$ of 50.

Figure 4.20 shows the same type of graph, this time with Bluetooth devices included in the position estimation. Apparently, the room is correctly classified most of the time and the number of confused locations is reduced from 12 without Bluetooth to 8 including Bluetooth. As in the previous case, this number gets smaller with rising $TimeWindowSize$. The misclassification with the highest distance to room 119-1 is the kitchen, which is only present in $TimeWindowSize$ 5 and 10. With $TimeWindowSize$ 50, only one misclassification is left, which is the corridor.

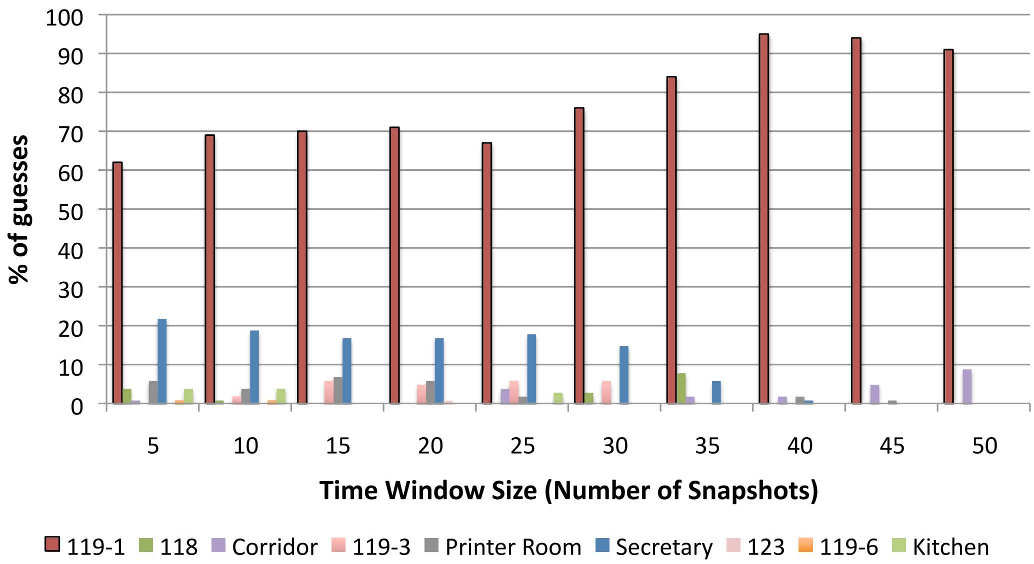


Figure 4.20: Hits and misses for office 119-1 including Bluetooth devices. Office 119-1 is correctly classified most of the time, the number of incorrect classifications is drastically reduced.

Figure 4.21: Accuracy plotted against $TimeWindowSize$ when only three landmarks are stored in the trained database.

How does the accuracy change according to the number of trained landmarks? To answer this question all but the tested three rooms were deleted from the trained landmarks database. The reason behind this question is that UBISPOT was designed to spot the user's personal landmarks and so it is reasonable that a user will not train each and every room in a building, but only important ones, like the own office, the kitchen, a meeting room. Figure 4.21 shows the accuracy statistics for the reduced set of landmarks. As expected, the accuracy is higher for all measured rooms. Even at $TimeWindowSize = 5$ each room gained a higher accuracy than in the first test and the accuracy ranges from 43% to 74% as compared to 11% – 66%. Moreover, with a $TimeWindowSize \geq 45$, each room could be identified with 100% probability.

4.3.5 Summary

UBISPOT is an opportunistic onboard/egocentric positioning system following the Always Best Positioned paradigm by combining cell info, WiFi access point MACs and Bluetooth addresses. It was implemented using the FOA core algorithm of LOCATO and was rigorously evaluated in a dense environment with small rooms close to each other. According to this evaluation, UBISPOT is capable to achieve room-level with a 68% accuracy (worst case) when all sensors are used.

The system is robust against environmental influences, like air humidity or the number of people in the room, due to omitting the signal strength in the fingerprints. It is however sensible to changes in the infrastructure, e.g. changing WiFi access points. This sensibility can be overcome by updating the reference fingerprints, either per user or through sharing fingerprints via UBISWORLD.

Regarding the cost of ownership, UBISPOT is a low-cost system for the operator, due to the opportunistic nature of the system, as well as for the user, as it runs on Android or Symbian smart phones with no additional hardware, which is also beneficial for the usability and applicability of the system. The algorithms are optimized for low computational resources, however the repeated inquiries of WiFi and Bluetooth sensors have an impact on the battery consumption of the mobile device. The power consumption can be improved by choosing longer delays between successive measurements, which delays the position determination. As the system is onboard-/egocentric, it provides a high privacy protection.

Although UBISPOT is designed as an opportunistic positioning system, an operator can choose to increase the accuracy of the system in their building by deploying additional Bluetooth beacons. This will increase the cost of ownership for the operator while maintaining the cost for the users.

4.4 LORIOT: A High Accuracy Onboard/Egocentric Positioning System

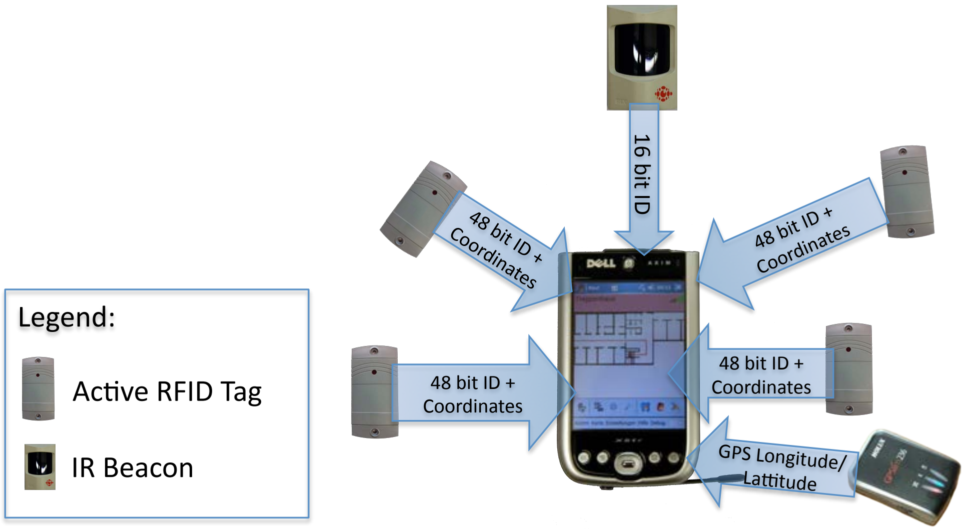


Figure 4.22: LORIOT uses active RFID tags and infrared beacons, which are distributed into the environment, to estimate the current position of a Windows Mobile PDA.

LORIOT is the acronym for Location and Orientation in Indoor and Outdoor Environments. The system aims at high precision positioning using a dedicated instrumentation of the environment, which consists of active RFID tags and infrared beacons. LORIOT follows the Always Best Positioned paradigm, by being able to work with either RFID or infrared alone, or by combining both if available. In outdoor scenarios, the system switches to a GPS receiver for obtaining positioning data. LORIOT can also deliver WGS84 coordinates indoors and is thus compatible with GPS based applications.

In comparison to UBISPOT, which has a delay of about 200 seconds until a new location can be derived, LORIOT's positioning is instantaneous and the system is thus capable of providing real-time positioning information of a moving user. Since all positioning calculations are performed on the mobile device of the user, their privacy is protected. Nonetheless, users can decide to share their position information with trusted services or persons.

The positioning accuracy of LORIOT was rigorously evaluated and in contrast to most positioning systems found in the literature, it was tested with moving users and compared to highly accurate natural ground truth traces.

4.4.1 Hardware

4.4.1.1 Senders and Sensors

Figure 4.22 shows the senders as well as the mobile device that LORIOT uses to estimate its own position. The required sensors are either built-in or attached to the device. In detail, the technologies for indoor positioning by LORIOT are:

Infrared LORIOT uses infrared beacons (IR beacons), manufactured by eyeled GmbH[3]. These beacons are powered by three AA batteries and send out a 16-bit wide identification code that can individually be adjusted for each beacon through DIP (dual in-line package) switches (blue boxes to the left and right of the infrared LEDs in Figure 4.23b).

The emitted infrared beam has a range of about 2 meters and has, due to the physical attributes of light, conical sending characteristics. The price for such a beacon is about 80 Euro. The required infrared sensor is often already integrated in mobile devices for data exchanging purposes. Due to restrictions of the infrared protocol, only one IR beacon can be detected at a time.

IR beacons are usually attached to walls or ceilings, pointing downwards to 'illuminate' a small spot on the floor, as depicted in Figure 4.23d. The position and dimensions of the infrared light cone can be adjusted by the mounting angles of the beacon itself as well as by bending the LEDs inside the beacon to widen or narrow the gauge of the light beam in the horizontal direction.

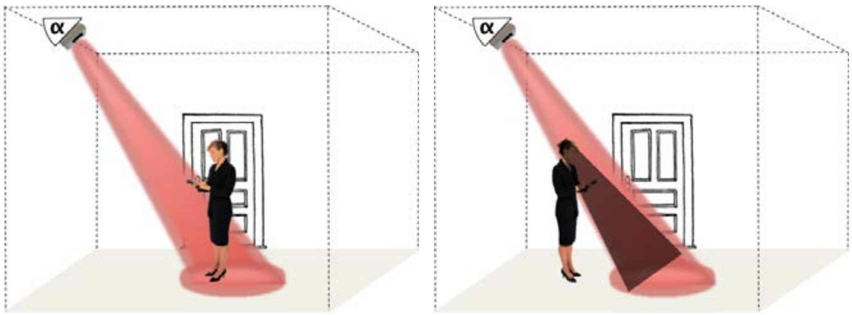
If the user's mobile device detects such a light beam, it can infer that the user is standing somewhere inside the illuminated spot. If the device knows the direction of the light beam, it can also derive direction information about the user.

On the downside, a free line of sight to the beacon is needed to detect it, meaning that the signal can be easily blocked by other persons or by users themselves, e.g when walking in the same direction as the light beam, as shown in Figure 4.23e.

To overcome this problem, several IR beacons pointing in opposing directions are often installed at one location. Because of these properties – short sending range and included directional information – IR beacons are mainly used for signaling points of interests, like exhibits in a museum or particular shelves in a shop, or for signaling decision points, like crossing corridors or doors on opposing walls.

[3]http://www.eyeled.de

(a) IR beacon (b) IR beacon circuit (c) Built-in IR sensor



(d) Installation of an IR beacon (e) Blocking an IR beacon

Figure 4.23: IR beacon as manufactured by Eyeled (a). Each beacon sends out a 16-bit wide ID, which can be configured by DIP switches. The circuit is powered by 3 AA batteries (b). The beacons can be detected and identified by standard IR sensors, which are often already integrated into mobile devices (c). IR beacons are perfect to signal points of interests (d).

Active RFID The active RFID tags used by LORIOT are manufactured by Identec Solutions AG[4]. These tags have a reading range of up to 6 meters and are powered by coin cell batteries, as can be seen in Figure 4.24c. According to the manufacturer, the batteries last 6 years when accessed up to 600 times each day.

Each tag has a memory of 64 bytes, out of which 56 bytes can be used to store application specific data. 6 bytes (48 bits) are used to store a unique ID for each tag, leading to over 281 trillion possible IDs. The IDs are hard-coded by the manufacturer and cannot be changed. The tags operate in the UHF band, or more specifically, at a frequency of 868 MHz for the European market and 915 MHz for the North American market. The tags also provide an LED, which can be triggered by the reading device.

[4]http://www.identecsolutions.com

(a) Active RFID tag front (b) Active RFID tag back (c) Active RFID tag circuit (d) Active RFID reader card

Figure 4.24: Active RFID tag by Identec Solutions as used in the LORIOT system. Each tag has a unique ID number that is also printed on the back of the housing (b). The tags are powered by coin cell batteries and the circuit contains an LED that can be used to indicate activity of each tag (c).

Due to the physical attributes of radio waves, the RFID's sending characteristics is radial. One active RFID tag costs about 20 Euro. Reading devices for these active RFID tags are also available from Identec Solutions. In conjunction with the mobile device, the i-CARD III PCMCIA reader card (shown in Figure 4.24d) is used, which costs about 1500 Euro. (The high costs mainly arise from the fact that these readers are manufactured in very low quantities.) The i-CARD III can detect 100 tags per second and through the use of a randomized anti-collision algorithm it can reliably identify up to 2000 tags in its reading range.

Active RFID tags overcome the restrictions of IR beacons by their high sending range and radial sending characteristics at the cost of lower precision and no immediate direction information. For the use with LORIOT, the RFID tags can be installed at the ceiling or the floor of a building and are usually ordered in a grid, so that multiple tags can be detected in one measurement. Although the reader card provides signal strength information for each detected tag, LORIOT does not directly rely on these measurements, but uses a similar approach as UBISPOT by taking the frequency of appearance into account.

4.4.1.2 Mobile Device

LORIOT was implemented in C++ for Windows Mobile devices. It was tested and evaluated on a Dell Axim X51v PDA. The Dell Axim uses Windows Mobile 5.0 but is unofficially upgradeable to Windows Mobile 6.0. A port of Android called Ax-

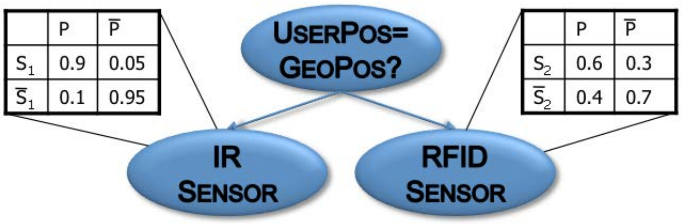


Figure 4.25: Geo referenced dynamic Bayesian network and the corresponding conditional probability tables as used by LORIOT.

Droid is also available[5]. The Axim has an Intel PXA270 processor running at 624 MHz and comes with 64 MB on-board RAM and 128 MB flash ROM. It features a CompactFlash (CF) Type II as well as a Secure Digital (SD) expansion slot and a long-range Infrared Data Association (IrDA) interface. Drivers to read the IR beacons through the IrDA interface were provided by eyled GmbH. The active RFID tag reader card was attached via a PCMCIA-to-CF slot adapter. An API to the RFID reader card is part of the development package of Identec Solutions. Since the PDA itself does not have an internal GPS, a Bluetooth GPS receiver was used for outdoor purposes.

4.4.2 Methods

LORIOT overcomes the disadvantages of IR beacons and active RFID tags by combining both their advantages. In order to fuse the sensory data of these two sender types, LORIOT uses the geoDBN core-algorithm of LOCATO. Figure 4.25 shows the used Bayesian network and its conditional probability tables. The CPT entries for the IR sensor are chosen to represent the high reliability of the IR technology: when standing in the range of an IR beacon, the probability to sense it is very high. For the RFID sensor, the values are chosen lower, because of the higher range of the active RFID tags and the high probability of overreach.

4.4.2.1 Obtaining Tag and Beacon Positions

Since LORIOT needs the coordinates of detected beacons or tags, a way had to be found how to communicate this information to the system. In an early version the tag and beacon IDs and their coordinates were simply stored as an XML-File on the mobile device, which was parsed when starting LORIOT. This approach is however impractical in real world situations where hundreds or thousands of buildings could be equipped with theses senders. Either the list would have to contain all tags and

[5] http://axdroid.blogspot.com

beacons of all buildings or the system would have to download this information accordingly when detecting an unknown ID. Current installations of LORIOT use the internal memory of the active RFID tags to store the coordinates they are signaling for. These coordinates can be of any kind, but in order to keep compatibility with existing location-based applications, it is best to use WGS84 coordinates. Additionally, IDs and coordinates of nearby IR beacons can also be stored in the tags. Thus, if a user is walking into a new building LORIOT can obtain all necessary information out of the current environment without the need of an additional data connection like WiFi or UMTS.

4.4.2.2 Data Caching

The measurement step of the geoDBN algorithm (Step 1 in Algorithm 2) results in a list of active RFID and IR beacon IDs. To read the memory content of the detected RFID tags, a special read-memory task has to be issued for each of the detected RFID IDs. In a real world setting however, it often happens that the ID of an RFID tag can be obtained, but the attempt to read the memory content fails. To overcome this problem and to reduce the amount of extra time that is needed to read the memory of each tag, a caching strategy is used, so that the memory-reading step only needs to be performed once for each previously unseen RFID ID.

It is however noteworthy that the effect of not being able to read the memory contents of an RFID tag could have a beneficial effect on the overall accuracy of the system, since it could act as a natural filter on tags that are too far away. The effect of the caching algorithm on the accuracy will thus be further discussed in the evaluation (see Section 4.4.4). The caching algorithm is straight forward and is executed in step 2 of the main geoDBN algorithm:

Algorithm 3 Caching Algorithm for tag and beacon coordinates in LORIOT

---

1. Iterate through the list of received RFID IDs from the newest measurement
   (a) If the current ID is already present in the database, retrieve the coordinates and proceed with the next ID.
   (b) If the current ID is not present in the database:
      i. Issue a read-memory command.
      ii. If not successful, proceed with the next ID.
      iii. If successful, parse the memory to extract the RFID coordinates and optional IR beacon IDs and their coordinates.
      iv. Store the new gained information in the database.

4.4.3 Output to the User

LORIOT is implemented as a background process and is not meant to provide direct output to the user. Once it is started, it checks for available sensors and immediately starts collecting measurements and performing position calculations. The estimated position coordinates are sent to an internal socket, to which other applications running on the mobile device can connect. There is however a simple user interface, that allows the user to configure certain aspects, like enabling/disabling caching or giving permission to send positioning data to a web service, e.g. UBISWORLD. Various applications that make use of LORIOT are described in Section 5.

4.4.4 Evaluation

First informal tests of LORIOT were conducted at the lab of Prof. Wahlster, where an accuracy of approximately 1 to 1.5 meters could be observed. Although the lab provided an ideal test field for UBISPOT (Section 4.3.4), this does not apply for LORIOT, which is supposed to deliver more accurate, sub-room level positions: the small size of the rooms in the lab does not allow for a large field of RFID tags and thus one of the main error sources – overreach of far-away RFID tags – could not be thoroughly tested. Moreover, since LORIOT is designed to position moving persons, appropriate movement traces are needed as ground truth for the evaluation. A rigorous evaluation was planned with the following requirements:

- The instrumented area should be large and include obstacles but should be without attenuating walls to maximize the probability of overreach.
- As ground truth, moving traces should be used, which should be as natural as possible to avert that users consciously or unconsciously adapt their movements to possible restrictions of the positioning system.

The evaluation should answer the following questions:

1. How accurate is LORIOT on average?
2. How is the accuracy influenced if
   (a) only IR beacons are considered in the position estimation?
   (b) only RFID tags are considered in the position estimation?
   (c) RFID tags and IR beacons, are considered in the position estimation?
   (d) the caching algorithm is enabled or disabled?
3. What is the influence of walking speed on the position accuracy?

Since LORIOT outputs numerical coordinates, the error of an estimated position can be expressed as the distance to the position of the user in the ground truth trace. The overall accuracy can be expressed via statistical analysis of the measured distances.

4.4.4.1 Evaluation Design

In order to address the questions above, the evaluation was done in three phases:

Phase 1: Natural walking traces were recorded with a video camera and manually transcribed into coordinates for each observed footstep.

Phase 2: Ground truth traces were marked on the floor and then followed again while carrying a mobile device with activated LORIOT. Each trace was followed twice: once with the original walking speed and once with a very slow walking speed. The calculated positions as well as all raw sensor data were logged for each trace.

Phase 3: Using the raw sensor data, all positions were recalculated using LORIOT's positioning engine to obtain four different conditions:

1. using only IR data
2. using only RFID data, without caching
3. using only RFID data, with caching
4. using RFID and IR data, without caching
5. using RFID and IR data, with caching

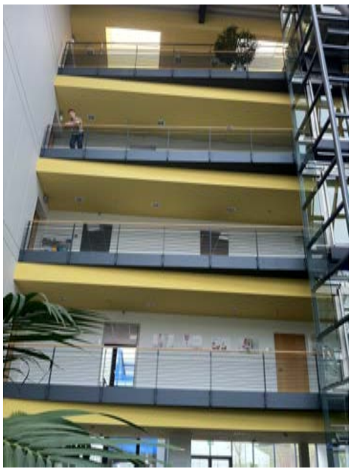
Each step will be explained in detail in the next sections.

4.4.4.2 Ground Truth Acquisition

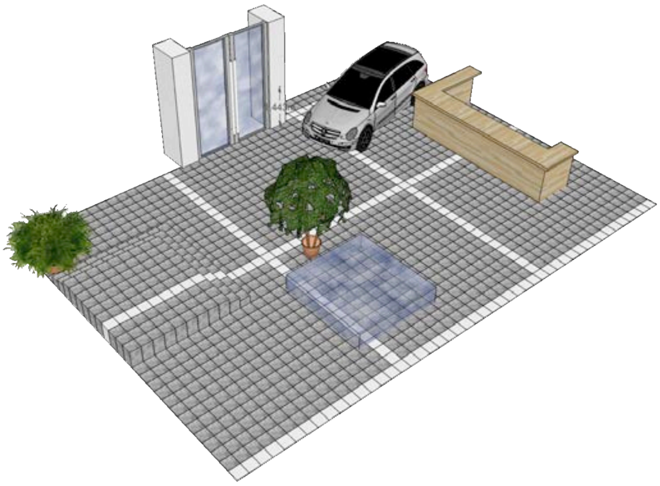
The main foyer of DFKI building in Saarbrücken was chosen as testfield for the evaluation because it provides a large area without attenuating walls (as can be seen in Figure 4.26a) and because it was built with an open architecture (Figure 4.26c), allowing to observe a large part of the area from the top level of the building (see Figure 4.26b). Moreover, the tile seams on the floor provide a visual coordinate system that can be used to acquire the needed ground truth traces. An accurate 3D model of the foyer was created that also represents each tile, as can be seen in Figure 4.26d.

(a) DFKI foyer

(b) DFKI foyer top view



(c) DFKI foyer open architecture



(d) 3D model of the DFKI foyer

Figure 4.26: The foyer of DFKI Saarbrücken was used a testfield, since it provides a large area and a visual coordinate system through the tiles.

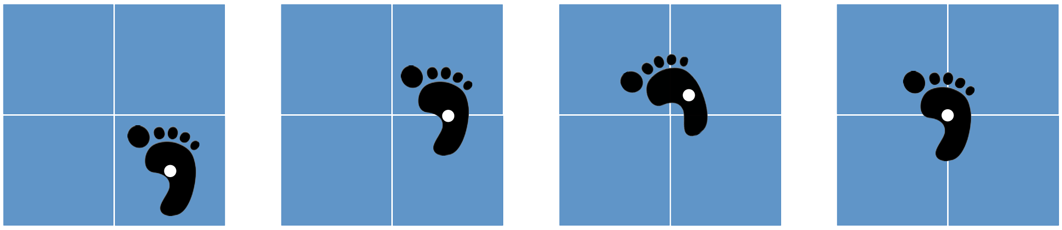
To obtain the required natural walking traces, an HD camera was installed at the top level such that a large part of the foyer could be observed (see Figure 4.27a). With permission of DFKI's workers' council, videos of walking people in the foyer were recorded over the course of three days. On each day, about 0.5 hours of video was recorded around lunchtime, which ensured that many people were crossing the foyer. To enhance the visibility of the tile seams, white adhesive tape was applied at selected spots (see Figure 4.27a). The tile seams and marked spots were used to overlay a grid on the videos, to further enhance the visibility of each tile. The grid also contained a unique ID for each tile.

To derive numerical coordinates for each single footstep of the recorded persons, the enhanced videos were manually analyzed. The quality of the videos was high

(a) Adhesive tape was used to enhance the visibility of the tile seams.

(b) Video-overlay representing the coordinate system.

Figure 4.27: A grid overlay was used to annotate each step of a person with according coordinates.



(a) Foot rests on one tile

(b) Foot rests on two tiles

(c) Foot rests on three tiles

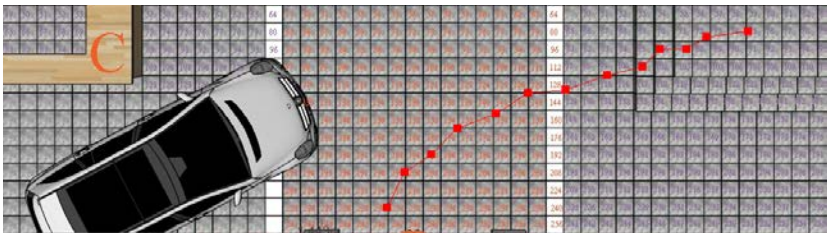
(d) Foot rests on four tiles

Figure 4.28: Four basic cases were considered for obtaining coordinates of each step of a person.

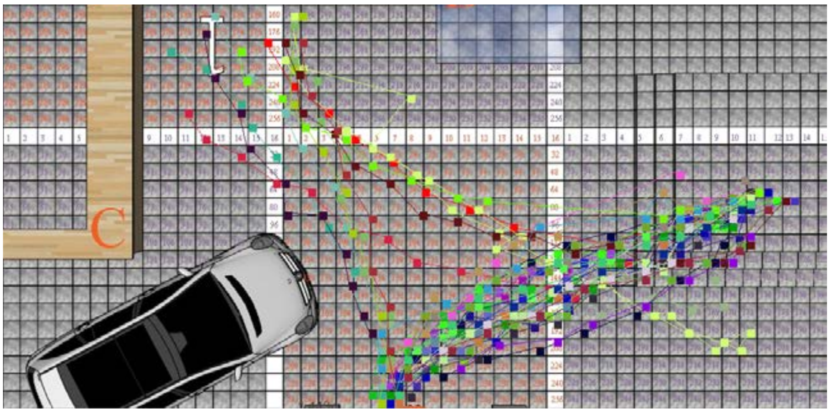
enough to discriminate four basic cases for each step, depending on how many tiles a person's foot is resting on. Figure 4.28 shows these four different cases. The actual coordinates were then derived by using the coordinates of the middle point of each covered tile and calculating the geometric middle according to the formula:

$$x = \frac{1}{n}\sum_{i=1}^{n} x_{\mathrm{id}_i},\ y = \frac{1}{n}\sum_{i=1}^{n} y_{\mathrm{id}_i} \tag{4.12}$$

where n is the number of tiles covered and x_{id_i} and y_{id_i} are the x and y coordinates of the middle point of a tile with identification id_i. The white dots in Figure 4.28 indicate the resulting coordinates for each case.



(a) Visualization of a single trace. Each square represents a single step.



(b) Visualization of all traces obtained in one day.

Figure 4.29: Example visualizations of extracted ground truth traces.

Using this method, [Saliba, 2011] extracted the coordinates and time-stamps for every single footstep of a recorded person. This lead to 119 highly accurate ground truth traces. A tool was implemented to visualize the recorded traces and to perform evaluation calculations. Figure 4.29 shows the visualization of a single trace as well as all traces obtained in one day.

4.4.4.3 Obtaining System Traces

To keep the ground truth traces as natural as possible, none of the recorded persons wore a mobile device. Thus, the acquisition of the system traces, i.e. LORIOT's estimated positions, had to be done in a separated step. In this step 58 active RFID tags were placed on the floor of DFKI foyer, with a distance of 105 centimeters between two adjacent tags. Figure 4.30 shows the distribution of the tags. Coordinates of each tag were stored on their internal memories using the same coordinate system as in the ground truth acquisition process. In addition, 10 IR beacons were placed in the environment using microphone stands.

From the 119 available traces, 16 were randomly chosen. These traces were laid out one after the other, according to the coordinates obtained in the ground truth acquisition process.

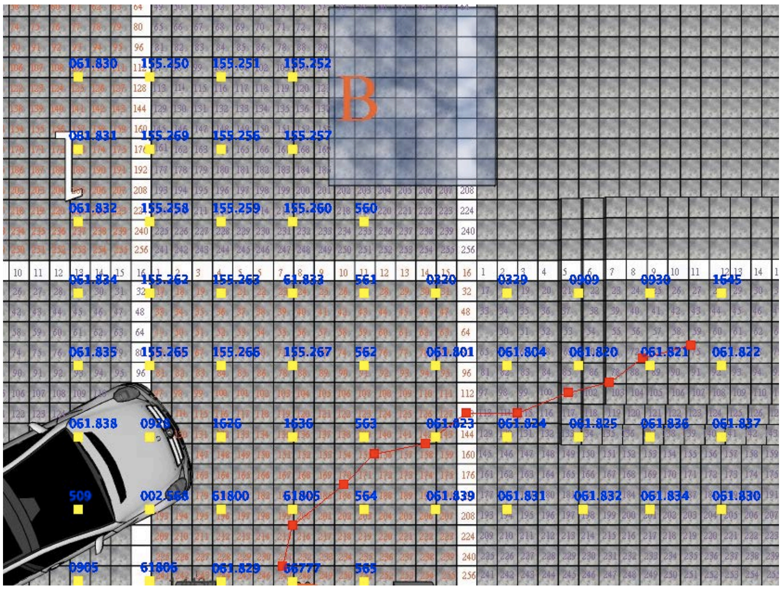


Figure 4.30: 58 active RFID tags were distributed in the DFKI foyer.

Figure 4.31 shows one such laid out trace. Each trace was then followed step by step while carrying a PDA with LORIOT running. Each trace was followed two times with two different speeds:

1. Original speed of the recorded trace. This was accomplished by playing back beeps according to the original time-stamps of the trace.

2. In a very slow speed, where after each step a pause of approximately one second was made.

The LORIOT system was modified to log all calculated positions, their time-stamps and raw sensor data into text files. This process led to 32 log files including derived positions and all measured raw sensor data.

From each log file, five system traces were derived by using LORIOT's positioning algorithm in varying conditions: considering only IR beacons, considering only RFID tags without caching, considering RFID tags & IR beacons without caching, considering only RFID tags including caching and considering RFID tags & IR beacons including caching. This led to 160 system traces that were compared to their respective ground truth.

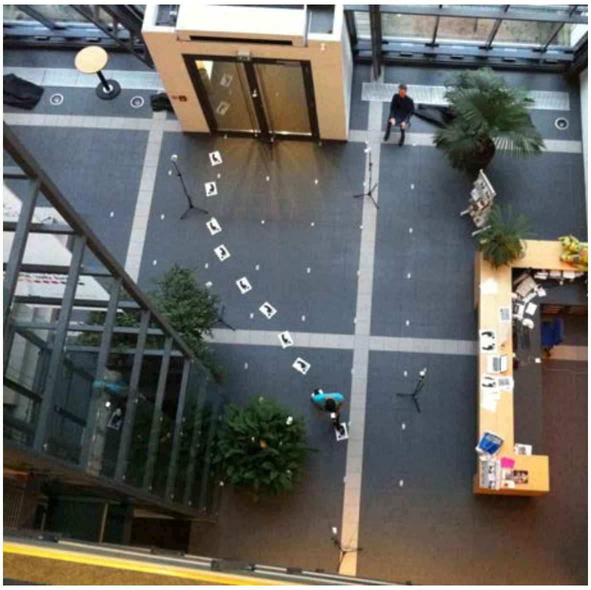


Figure 4.31: Traces were laid out on the floor and followed while carrying a mobile device running LORIOT.

4.4.4.4 Calculation of the Error Distance

As indicated above, the extracted traces from the ground truth acquisition contain highly accurate data for each single footstep. LORIOT on the other hand, does not measure footprints. It was designed to estimate the position of the user. The question arises what the position of a user is, if the positions of his feet are known. For the evaluation, it was assumed that the user's position is somewhere on the line between two successive foot positions.

This consideration is important, since LORIOT computes a new position every time a new measurement is taken, meaning that time-stamps of derived positions do not necessarily coincide with time-stamps of ground truth traces. Thus, a way had to be found to find the user's ground truth position at an arbitrary time-stamp.

Figure 4.32 exemplifies the situation. The two footprints indicate two subsequent footsteps of a ground truth, TS_R and TS_L are the time-stamps for the right and left foot. The blue dot shows the position derived by LORIOT, derived at time-stamp TS_{LORIOT}. According to the exemplary given time-stamps, LORIOT's position was derived 0.325 seconds after the right foot reached the ground and 0.375 seconds before the left foot will reach the ground in the ground truth. The user's position in the ground truth is thus somewhere in between.

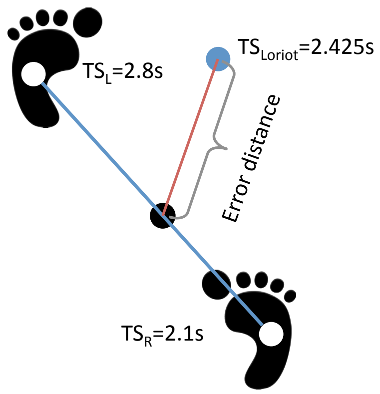


Figure 4.32: Ground truth time-stamps of single footsteps and LORIOT time-stamps of user positions do not necessarily coincide.

To interpolate where the user's position was in the ground truth at time T_{LORIOT}, the current velocity v is calculated by dividing the distance between the two footsteps with the time difference between the two footsteps:

$$v = \frac{\sqrt{(x_L - x_R)^2 + (y_L - y_R)^2}}{\text{TS}_L - \text{TS}_R} \tag{4.13}$$

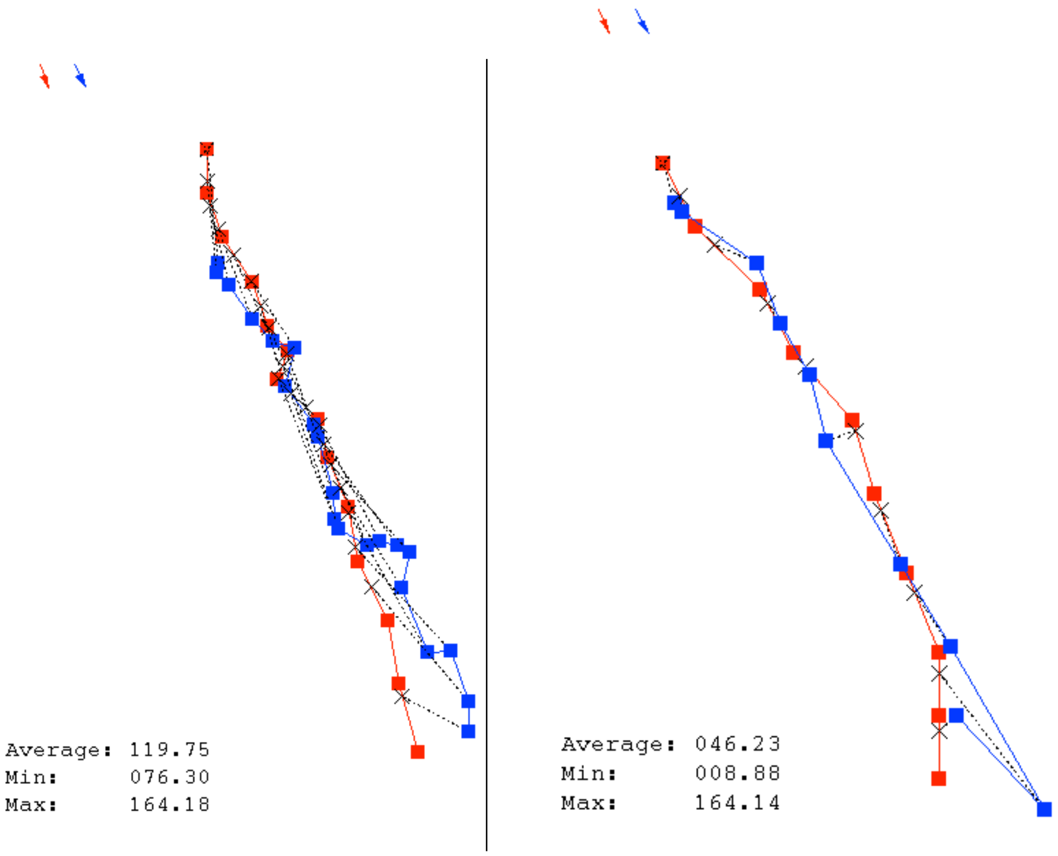
where (x_L, y_L) and (x_R, y_R) are the coordinates of the left and right foot. By multiplying this velocity with the time difference between TS_{LORIOT} and TS_R, the distance d that the user has covered since putting their right foot down can be derived:

$$d = v \times (\text{TS}_{\text{LORIOT}} - \text{TS}_R) \tag{4.14}$$

The user's position $P_{\text{groundtruth}}$ at time TS_{LORIOT} in the ground truth is estimated to be at distance d from the right footstep on the line between the two footsteps. The positioning error is thus the distance from LORIOT's derived position to $P_{\text{groundtruth}}$. $P_{\text{groundtruth}}$ is indicated as a black dot in Figure 4.32.

4.4.4.5 Results

Figure 4.33 shows two comparisons of system traces with their respective ground truth: Trace 2 in the *only RFID, with cache* condition and Trace 3 in the *RFID & IR, with cache* condition. The red squares represent the footsteps of the ground truth. The blue boxes depict the user position as derived by LORIOT. The black crosses show the interpolated user position on the ground truth. Each interpolated user position is connected via a black dotted line with the corresponding system position. The red and blue arrows show the general walking direction of the ground truth and system trace respectively. The average positioning error as well as the minimum



(a) Trace 2 in the *only RFID, with cache* condition.

(b) Trace 3 in the *RFID & IR, with cache* condition.

Figure 4.33: Two example results from the evaluation. The red boxes depict the ground-truth steps. The blue boxes represent the positions derived by LORIOT. The black crosses show the interpolated user steps, which are connected by black dotted lines with their respective user position.

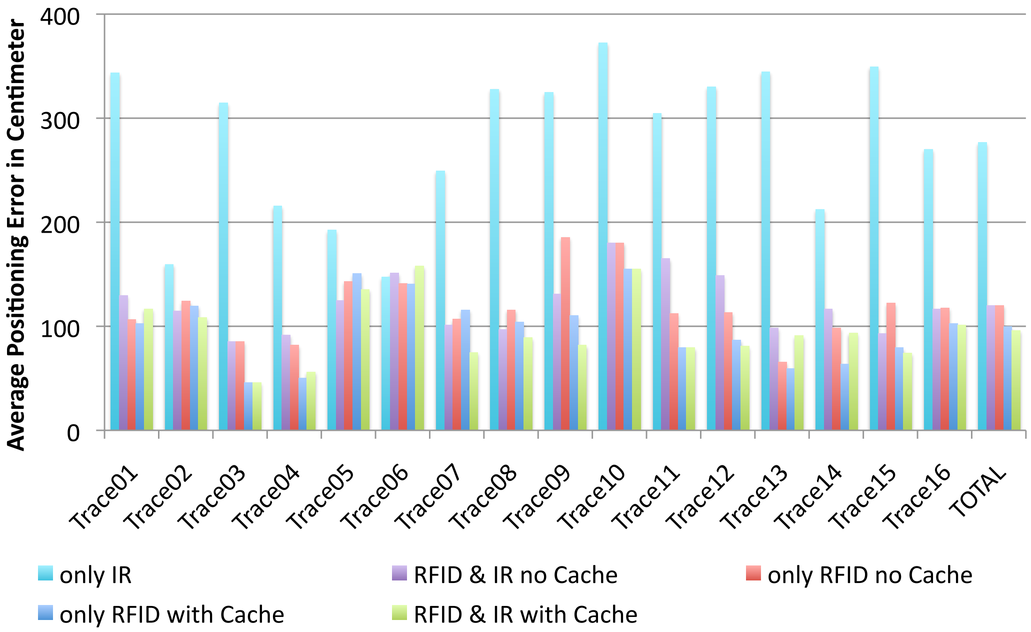


Figure 4.34: The average positioning error of all traces with original velocities and with respect to the five tested conditions.

and maximum positioning error of the trace is printed in the bottom left corner. All graphical representations of each trace in each of the five conditions can be found in Appendix A.1.1 and Appendix A.1.2.

The average positioning error in centimeters for each trace and each condition is summarized in Figure 4.34. The last column, labeled TOTAL, shows the average error over all traces for each condition. Table 4.2 summarizes the key values for each condition. The entries are ordered top to bottom by their average positioning error over all traces (from lowest to highest). The standard error as well as the 95% confidence interval is given for each condition. A repeated measures ANOVA was performed over the differences of each trace and for each condition, and showed an overall significance with $F(4, 180) = 47.3, p < .001$.

How is the accuracy influenced if the caching algorithm is enabled or disabled? Table 4.2 shows that both cached conditions (*'only RFID with cache'* and *'RFID & IR with cache'*) outperform all other conditions. With 99.79 centimeters, the average positioning error in the *'only RFID with cache'* condition is 20.57 centimeters lower than in the *'only RFID no cache'* condition. A Bonferroni adjusted pairwise comparison shows that this difference is significant with $p < .001$. The difference between the average positioning error of the two *RFID & IR* conditions amounts to 23.82 cen-

Condition	**Average** in cm	Std. Error in cm	95% Confidence Interval Lower Bound in cm	Upper Bound in cm
RFID & IR with cache	**96.31 (1)**	4.00	88.42	104.20
only RFID with cache	**99.79 (2)**	4.13	91.64	107.94
RFID & IR no cache	**120.13 (3)**	5.80	108.68	131.57
only RFID no cache	**120.36 (4)**	4.97	110.55	130.17
only IR	**276.67 (5)**	14.03	248.99	304.36

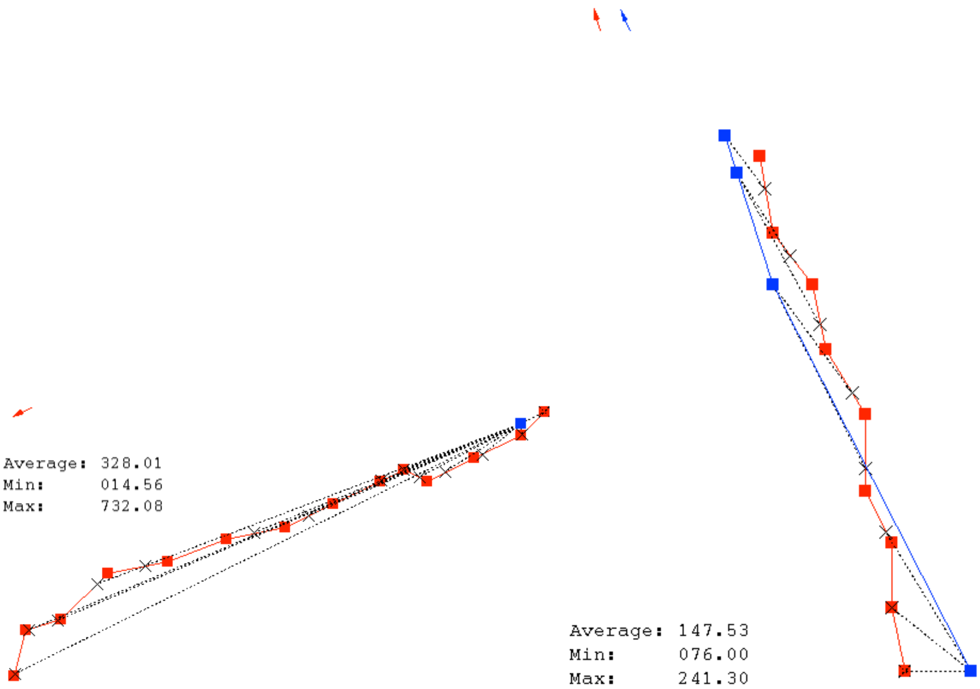
Table 4.2: Comparison of positioning error when following the ground truth in original velocity. The numbers in parenthesis show the ranking of each value.

timeters in favor of the *with cache* condition and is also significant with $p < .001$. It can thus be concluded that the caching algorithm improves the positioning accuracy by approximately 20 centimeters in average.

How is the accuracy influenced if only IR beacons are considered in the positioning evaluation? When only considering IR beacons, LORIOT could only achieve an average accuracy of 2.77 meters, which is the highest measured average positioning error measured in this evaluation. The difference to all other conditions is significant with $p < .001$ for all pairwise comparisons.

The minimal positioning error was 14 centimeters and the maximum was 7.32 meters. Both values were achieved in Trace 8, which is shown in Figure 4.35a. Only one IR beacon was received in this test and thus only one position was fixed by LORIOT. Analyzing all IR only traces shows that in 11 out of the 16 traces only one IR beacon was detected during the test walks. Two IR beacons were detected in three traces. Three and four beacons where detected in only one trace each. In Trace 6, four beacons were detected and, with 1.48 meter, this trace also shows the lowest average positioning error for all traces in the *'only IR'* condition.

The low accuracy in the *'only IR'* condition was to be expected and is due to the comparably sparse instrumentation of the testfield with IR beacons. IR beacons are advantageous at precise points of interest, like exhibits in a museum, particular shelves in a retail environment or decision points in a narrow corridor. Furthermore, the *'only IR'* condition provides a special case since without active RFID tags no coordinate



(a) Result for Trace 8 in the *'only IR'* condition.

(b) Results for Trace 6 in the *'only IR'* condition.

Figure 4.35: The worst (a) and best (b) result for the *'only IR'* condition. In Trace 8 only one IR beacon was detected. Trace 6 contains 4 detected IR beacons.

information can be stored in the environment (as described in Section 4.4.2.1). Thus, a list containing the beacon IDs and their coordinates has to be stored on the mobile device. Installing only IR beacons in a large area with nearly no walking restrictions is therefore only recommended for special applications, like museums or shops.

How is the accuracy influenced if only RFID tags are considered in the position estimation? The *'only RFID with cache'* condition shows the second best accuracy, with an average positioning error of 99.79 centimeters. The minimum positioning error in this condition was 3.88 centimeters (Trace 15) and the maximum was 276.88 centimeters (Trace 5). The *'only RFID no cache'* condition ranked second to last, with an average positioning error of 120.36 centimeters and minimum and maximum error of 65.91 (Trace 13) centimeters and 185.67 (Trace 9) centimeters respectively. The average is still 156.32 centimeters better than the *'only IR'* condition and this difference is significant with $p < .001$. Since caching already proved to be advantageous, it can be concluded that LORIOT can achieve a positioning accuracy

Condition	Average in cm	Minimum in cm	Maximum in cm
RFID & IR, with cache	**24.81 (1)**	13.21 (2)	44.26 (3)
only RFID, with cache	**25.00 (2)**	12.60 (1)	39.10 (1)
only RFID, no cache	**30.33 (3)**	16.05 (3)	43.80 (2)
RFID & IR, no cache	**31.39 (4)**	19.34 (4)	48.77 (4)

Table 4.3: Comparison of positioning errors when following the traces in slow velocity.

of approximately 1 meter in an environment that is densely instrumented with only active RFID tags.

How is the accuracy influenced if RFID tags and IR beacons are considered in the position estimation? Table 4.2 shows the lowest average positioning error in the case of combined RFID and IR instrumentation and with enabled caching. With 96.31 centimeters, the average positioning error is approximately 3 centimeters lower than RFID alone (with enabled caching). However, a pairwise Bonferroni adjusted comparison shows that this difference is not significant. The difference of 0.23 centimeter when comparing *only RFID* and *RFID & IR*, both with caching, is negligible and also not significant. These low, not significant differences can also be contributed to the sparse IR beacon instrumentation as well as to the high walking speed of the ground truth, which makes it less probable that an IR beacon will be properly detected.

What is the influence of walking speed on the position accuracy? To answer this question, the raw sensor data log-files of the slowly walked traces where analyzed. Because of the different velocities of the ground traces and the re-walked traces, there is no direct relation between their time-stamps, and thus the calculation of the error distance had to be adapted accordingly.

For the slow velocity traces, for every calculated user position the nearest footstep in the ground truth was found and the distance to that footstep was taken as the positioning error. If a footstep in the ground truth had already been used as reference point, it was not used again and only footsteps with a higher time-stamp than the last footstep were allowed. This method is thus analogous to a comparison of graphical similarity.

Table 4.3 summarizes the average, minimum and maximum positioning error for each of the four conditions. The results when walking slowly are greatly improved.

Condition	Average cm	Minimum cm	Maximum cm
RFID & IR, with cache	**57.48 (1)**	27.24 (1)	135.76 (2)
only RFID, with cache	**61.69 (2)**	28.77 (3)	159.63 (3)
only RFID, no cache	**73.11 (3)**	33.44 (4)	127.73 (1)
RFID & IR, no cache	**83.73 (4)**	27.30 (2)	219.23 (4)

Table 4.4: Comparison of positioning error when comparing the graphical similarity of the system to the ground truth.

The best result was achieved with RFID & IR and enabled caching. This condition led to an average positioning error of only 24.81 centimeters. The highest average positioning error was measured in the condition where RFID and IR was used without caching and amounts to 31.39 centimeters.

A part of this improvement is due to relaxed measurement of the positioning error. To test if the improvement can be attributed to the different measurement method alone, the traces that were followed based on the time-stamps of the ground truth were re-analyzed using the same method.

Table 4.4 shows the results of the analysis. The results are indeed an improvement over the time-stamp based analysis, but not as good as the measurements that were based on the slow velocity traces. In the worst case (*'RFID & IR, no caching'*), the average positioning error is 83.73 centimeters. Compared to the 31.39 centimeters when walking slowly, this average is approximately two times higher.

The lowest achieved average positioning-error was 57.48 centimeters and was measured with RFID & IR and enabled caching. This positioning error is also approximately two times higher than the best average when walking slowly.

It can thus be concluded that the accuracy of LORIOT is higher at slow walking speeds.

How accurate is LORIOT on average? Considering the above results, LORIOT achieves its highest accuracy with enabled caching and with either RFID alone or with combined RFID and IR instrumentation. The average positioning error over all traces of *'only RFID with cache'* and *'RFID & IR with cache'* results in 98.05 centimeters at normal walking speed. The accuracy is higher at slow walking speeds. As a slower walking speed can be expected if a person is walking through unknown territory, while exploring their surroundings or when trying to find their way, this higher accuracy will most likely be available, when a person is using a location-based service.

4.4.5 Summary

LORIOT is an onboard/egocentric positioning system designed for instrumented environments. It follows the Always Best Positioned paradigm by sensor fusion of an active RFID reader and an infrared sensor. LORIOT was rigorously evaluated by using natural walking traces with step-accuracy as ground-truth. In a densely instrumented environment, an accuracy of ≈ 1 meter can be achieved.

As the geoDBN core-algorithm of LOCATO does not incorporate signal-strength information, LORIOT is robust against environmental factors, like air-humidity and the number of people in the environment. Furthermore, as no trilateration or triangulation is performed, a single RFID tag or IR beacon in the receiving range is sufficient to determine a position. The high accuracy of the system is traded against high cost of ownership for the operator as well as for the user.

In low quantities, a single active RFID tag costs ≈ 20 Euro, a single IR beacon ≈ 80 Euro. The needed active RFID reader on the user's side costs $\approx 1,500$ Euro, which is mainly due to the low manufacturing quantities of this type of RFID reader. As active RFID readers do not contain any costly parts, there is no obvious reason for the high price, except the development costs. In large quantities, it should be possible to manufacture such a reader in the range of tens of Euro.

Another cost factor for the operator are the batteries of the infrared beacons and the active RFID tags, which includes the costs for the batteries themselves as well as the costs for the manual labor to replace them. A possible to solution to reduce the maintenance costs is to either use solar cells or energy harvesting. The latter technology is available from the company Powercast[6] and enables devices to draw their power from special RF-based power transmitters. However, further research is needed to test the possible interference of the power transmitters with the active RFID signals.

With respect to usability and applicability, LORIOT runs on any standard Windows Mobile device with infrared capabilities and the additional active RFID reader. In order to make use of the infrared beacons, the device has to have a line of sight to a beacon, i.e. it has to rest in the hand of the user. This is compensated by the active RFID tags, which can also be read while carrying the device in a pocket or bag. Although the algorithms of LORIOT are optimized for low computational complexity, the repeated scanning of the active RFID reader has an impact on the device's battery. This impact can be lowered, by decreasing the scanning frequency, which will delay the position determination.

[6] http://www.powercastco.com

5 Location-Based Services Realized with the LocaTo Platform

Several applications and systems were realized using LocaTo and the example positioning-systems. In the following, first a scenario will be described that ties selected realized applications together. In the second part, each application will be described in more detail.

5.1 Example Scenario

5.1.1 Hermione's Lazy Saturday

Preparations and a Nap

Imagine a sunny Saturday. Hermione, the example user in this scenario, is invited to a friend's party and she has promised to bring a good whiskey and the ingredients of her favorite cocktail: White Russian. Since a new shopping mall has opened not far from her friend's place, she has planned to spend a few hours there, before heading for her friend's home. In preparation of her day, she sets an appointment in her location-aware task planner, pointing to the address of her friend and setting the time she has promised to arrive. Because she still has to buy the whiskey and the ingredients for the White Russian, she also enters a task, specifying the items she has to buy. She doesn't add a specific time or place to the task, but rather specifies that any shop that carries the ingredients will do and that the task has to be completed before she arrives at the party. Hermione decides to take a short nap, before starting her trip. After she has lain down for a while in her bedroom, her mobile phone determines her position using UbiSpot and automatically sets itself to silent-mode.

An hour later, she wakes up refreshed and as she prepares for her trip to the mall, her mobile phone switches back into its normal mode.

Finding a Parking Spot

As Hermione enters her car, her mobile phone automatically connects to the car's entertainment system via a Bluetooth connection. Because of the car's Bluetooth ID, UBISPOT can infer that she is now in the car and asks the car to forward its GPS coordinates. The phone navigates her to the shopping mall, using the car's GPS, which has a better reception than the phone's GPS receiver. When entering the parking deck, the car loses its GPS reception, but detects the presence of active RFID tags. Using LORIOT to determine its position, the car navigates Hermione to a free parking space. As she turns off the car's engine, the Bluetooth connection gets lost and the mobile phone automatically stores the current position as the car's parking position. With the help of active RFID tags and infrared beacons, the mobile phone is able to guide Hermione to the next exit and the few hundred meters outdoors into the shopping mall.

Finding a Malt in the Mall

As she enters the mall, her mobile phone informs her that she can use the public infrastructure by registering the phone's Bluetooth address with the malls application server and allowing restricted access to her user profile. As Hermione likes exploratory shopping tours, she agrees because she knows that this will cause the public displays in her vicinity to display selected information to her, for example, special offers or things that she has stored in her 'things I would buy if I would accidentally stumble over them'-list.

After mindlessly poking around in the mall for quite a while, she passes by a grocery store. Her mobile phone automatically checks if a good whiskey and the ingredients for White Russians are available and triggers an alarm, notifying Hermione that this is a good opportunity to buy the items in her task list. The phone also periodically checks the distance of Hermione's current position to her friend's home, approximates the driving time and automatically sets a reminder when she has to leave the mall. Because of this, Hermione realizes that she has to hurry. She enters the grocery store, which is an affiliate of her favorite department-store chain and grabs a shopping trolley. The trolley is equipped with a touch-screen and Hermione can identify herself to the trolley as a loyal customer by quickly dragging her wallet, which contains her NFC enabled customer card, over the touch-screen. The shop's floor is instrumented with a grid of passive RFID tags and the trolley, which is equipped with a passive RFID reader, can thus determine its exact location inside the store. Through the shop's infrastructure, the trolley has access to Hermione's shopping list and calculates an optimal route to buy the whiskey and needed ingredients. With the help of the trolley's navigation instructions, Hermione is able to complete her shopping task in record-breaking time and leaves for the parking deck. 'Good thing my phone remembers where I parked my car, because I bloody didn't', she mumbles as her phone guides her back.

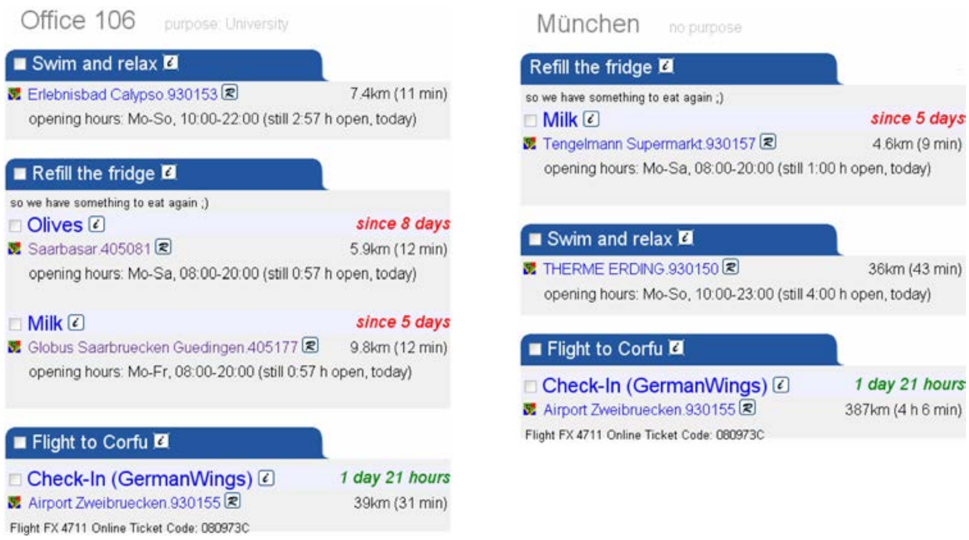
I Can't Hear You but I Can Hear my Phone

She arrives at the party right on schedule. Her friend – Ron – is an avid technology fan and is burning to show Hermione his newest acquisition: a location-adaptive audio-notification service. As most of Ron's friends are addicted to emails, SMS and traditional phone calls, they are usually afraid to miss any message notifications and, very much to Ron's dismay, tend to ask him to turn the music down so they can hear their phones.

'The location-adaptive audio-notification service', Ron shouts at Hermione over the loud music, 'changes that. If you connect your phone to this service, you can choose a personal audio notification pattern. If you receive a message or phone call, this audio pattern will be seamlessly integrated into the music on the speaker that is nearest to you.' After registering her phone with the notification service, Hermione heads straight into the kitchen to mix her first White Russian of the evening. 'A speaking cocktail shaker? Ron, you're kidding me!' (although the speaking cocktail shaker is not directly related to this thesis, interested readers can find more about this fine piece of gadgetry in [Schmitz, 2010]).

5.2 UBIDOO: Location-Aware Task Planner

The calendar, in which Hermione enters her appointment and the ingredients she needed to buy, was implemented by [Fickert, 2007] in his master's thesis. The system is called UBIDOO, which stands for UBIquitous to-DO Organizer. It realizes a ubiquitous task planner, which integrates a calendar and a to-do list. In contrast to conventional calendars and to-do lists, UBIDOO does not only allow to set reminders for appointments and tasks to a specific date and time, but also to places. Moreover, since UBIDOO uses the spatial ontology of UBISWORLD (see Section 2.4.2.1), besides specific places also more general concepts, like 'Store' or 'Grocery Store' can be specified. As indicated in the scenario, this enables UBIDOO to trigger a reminder if a user passes by a location that fulfills the specified role. Additionally, general ac-



(a) To-do list view in Saarbrücken (b) To-do list view in Munich

Figure 5.1: The here-and-now view of the same to-do list at different locations [Stahl et al., 2007].

tivities, e.g. 'Swimming', can be specified in the same way, using an appropriate ontology from UBISWORLD. In such a case, the system will automatically search for suitable locations for the activity, e.g. a nearby lake or swimming pool.

UBIDOO was implemented as a web-service and has direct access to UBISWORLD, where it can also retrieve the current position of the user from the user model. UBIDOO is therefore fully compatible to all positioning systems that report their position determination to UBISWORLD. With this position information, the to-do manager can constantly check for nearby places that could be used to fulfill any of the user's tasks. Moreover, the service calculates a so-called 'here-and-now' view, which sorts the to-do list according to the current time and the current position of the user.

Examples of the here-and-now view of the same to-do list at two different positions are shown in Figure 5.1. The list contains three main tasks, Swim and relax, Refill the fridge and Flight to Corfu. The Refill the fridge task has two specified sub-tasks: buying olives and buying milk. Buying olives was assigned to a specific shop in Saarbrücken by the user, whereas buying milk was just specified as a general shopping task. Swim and relax was associated to any place that allows swimming and the Flight to Corfu was of course assigned to a specific airport.

Figure 5.1a shows the to-do list when the user is in their office in Saarbrücken. The list is ordered according to the time that is needed to reach the associated location (the time is shown in parenthesis next to the distance). As can be seen, the swimming task is on the top of the list, although the distance to the appropriate location is higher than the distance to the shop, where the user intends to buy olives. This is due to the better (faster) reachability of the swimming-location. Figure 5.1b shows the same to-do list, when the user is in Munich. The task to buy olives is omitted, since the specified shop is too far away from the current location. The shop in which to buy milk has been replaced by a shop near to the user, and so was the location to 'swim and relax'. It can also be seen, that the time to reach the flight to Corfu has been adjusted from 31 minutes to 4 hours and 6 minutes.

UBIDOO derives these traveling times through a web-service called eRoute, which is provided by the company PTV AG[1]. This service calculates routes between two given locations and estimates the driving time. Traveling times through buildings, e.g. from the current position to the exit, are estimated using YAMAMOTO. Besides the traveling time, UBIDOO also takes opening and closing times of places like shops into account and either choses places that are currently open or will add the time until a place is opening again to the traveling time.

Besides organizing tasks and appointments for single users, UBIDOO also allows to manage group tasks. For example if a group of people is planning a party, they can set up several tasks and assign them to different people or subgroups of people. If somebody in a subgroup marks a task as completed, this task will automatically be removed from their task list (cf. [Stahl et al., 2007, Fickert, 2005]).

5.3 Parking-Deck Navigation

In his master thesis, [Gholamsaghaee, 2007] developed a parking-deck navigation called PARKNAVI, which uses LORIOT as positioning engine. PARKNAVI was designed to provide arriving cars with a route to an empty parking space in a car park and to guide passengers from their parked car to the exit as well as to guide them from the entrance back to their parked car.

The application was tested in P20, a multistory car-park at Munich Airport Center (MAC), Germany. P20 has approximately 6400 indoor parking-spaces on 11 levels and is the second biggest multistory car-park in Germany. This car park was already equipped with optical sensors that are used by the operating company to detect empty parking spaces. This information can be used by PARKNAVI to choose the nearest empty parking space to the passenger's next destination, e.g. Terminal 2, to minimize their walking distance. To ensure position information while navigating the user from their car to their destination, or back to their car, LORIOT is used in conjunction with active RFID tags.

[1] http://www.ptv.de

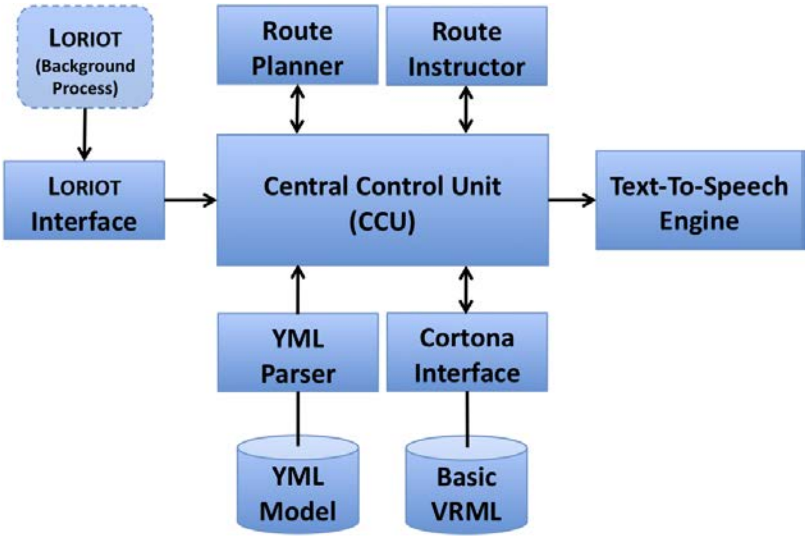


Figure 5.2: Components of PARKNAVI. The current user position is read from LORIOT to update route instructions accordingly [Gholamsaghaee, 2007].

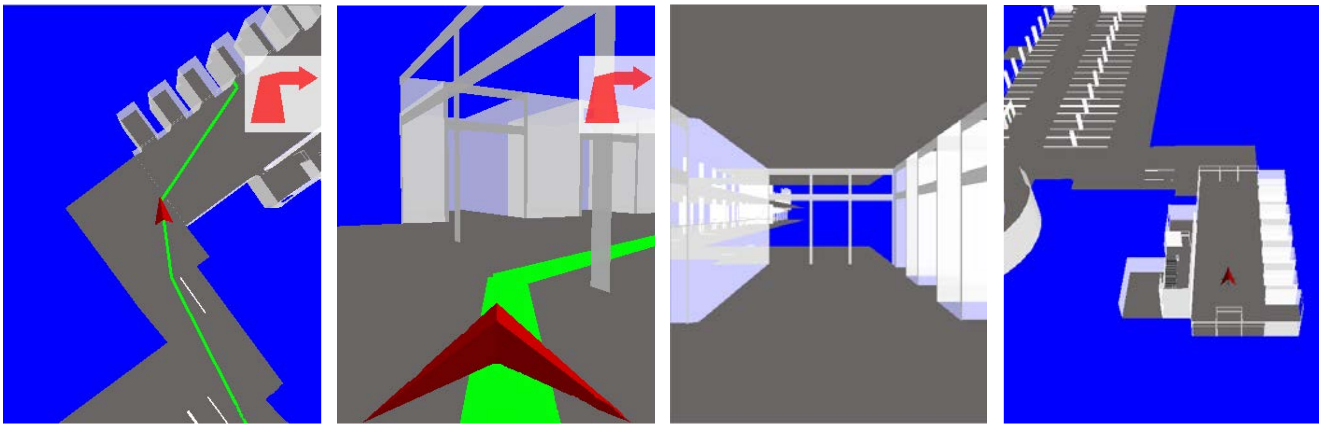
To test PARKNAVI, parts of the third and fourth story of the car park were modeled with YAMAMOTO and equipped with 70 active RFID tags. These tags were installed at the ceiling using styrofoam blocks to insulate them from the ferroconcrete structure of the car park, which tends to attenuate the radio signal of the tags.

Figure 5.2 shows the components of the PARKNAVI system. The Central Control Unit (CCU) retrieves the current user position via a socket connection to LORIOT, as described in Section 4.4.3. The CCU also reads and parses a YAMAMOTO model (YML, Yamamoto Modeling Language) of the car park and forwards this data to a VRML renderer. PARKNAVI uses the *Pocket Cortona* VRML-Viewer by Parallel Graphics[2] to accomplish the rendering. A route planner calculates the shortest path from the user's current position to their destination. The route planner module was originally developed by [Waßmuth, 2006] to plan pedestrian routes in YAMAMOTO models. Car parks however represent a special case – unlike normal buildings they are used by pedestrians as well as by vehicles. Since PARKNAVI should be able to navigate both, YAMAMOTO and the route planner had to be extended accordingly: the first to represent one-way routes and the latter to take into account these routes when planning for a vehicle. Furthermore, the planner was extended to allow for the use of elevators, which are also present in P20.

Using the planned route and the model, PARKNAVI is then able to create graphical and verbal route instructions. IBM's Embedded ViaVoice©[3] was used to provide spoken instructions.

[2]http://www.parallelgraphics.com

[3]http://www-01.ibm.com/software/pervasive/embedded_viavoice/



(a) Plan view (b) Tethered view (c) Immersed view (d) Monitoring view

Figure 5.3: PARKNAVI screenshots [Gholamsaghaee, 2007]

For graphical route instructions, the user can choose between four options (shown in Figure 5.3): plan view, tethered view, immersed view or monitoring view. According to [Baus, 2003], each of these visualizations meets different demands of user. The plan view shows a part of the map from a bird's eyes perspective. This view is best in situations in which a user needs to gain knowledge about the structure and layout of the current environment. The tethered view provides a visualization as if a camera would be placed above and behind a user. The immersed view shows a representation of the environment from the perspective of the user itself. The latter two views are best for efficient navigation presentations, as they show the environment in a fashion that is close to the perception of a user. The monitoring view is a 3D representation of the map, shown from the perspective of a camera that is mounted at an arbitrary point. This view combines the advantages of a 3D representation, i.e. a view that is close to the perception, with the advantage of the plan view, i.e. a good overview on the structure and layout of the environment.

5.4 Hybrid Navigation-Visualization on Nomadic Devices

HYBNAVI was developed by [Mutafchiev, 2008] and is an extension of PARKNAVI, and thus shares the basic software design (see Figure 5.4). HYBNAVI is an abbreviation for *HYB*rid *NA*vigation *VI*sualization and it enables route finding and guiding not only within one building, but also from rooms in one building to other rooms in different buildings, e.g. on a university campus or large factory premises. This involves

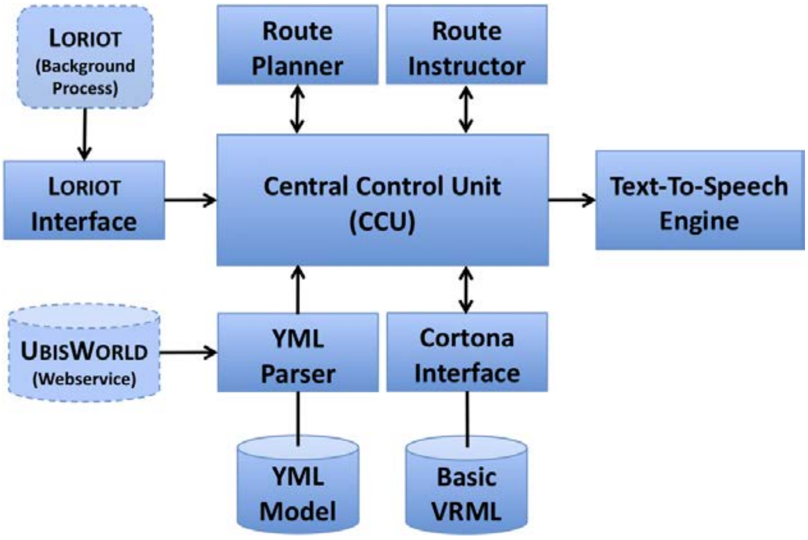
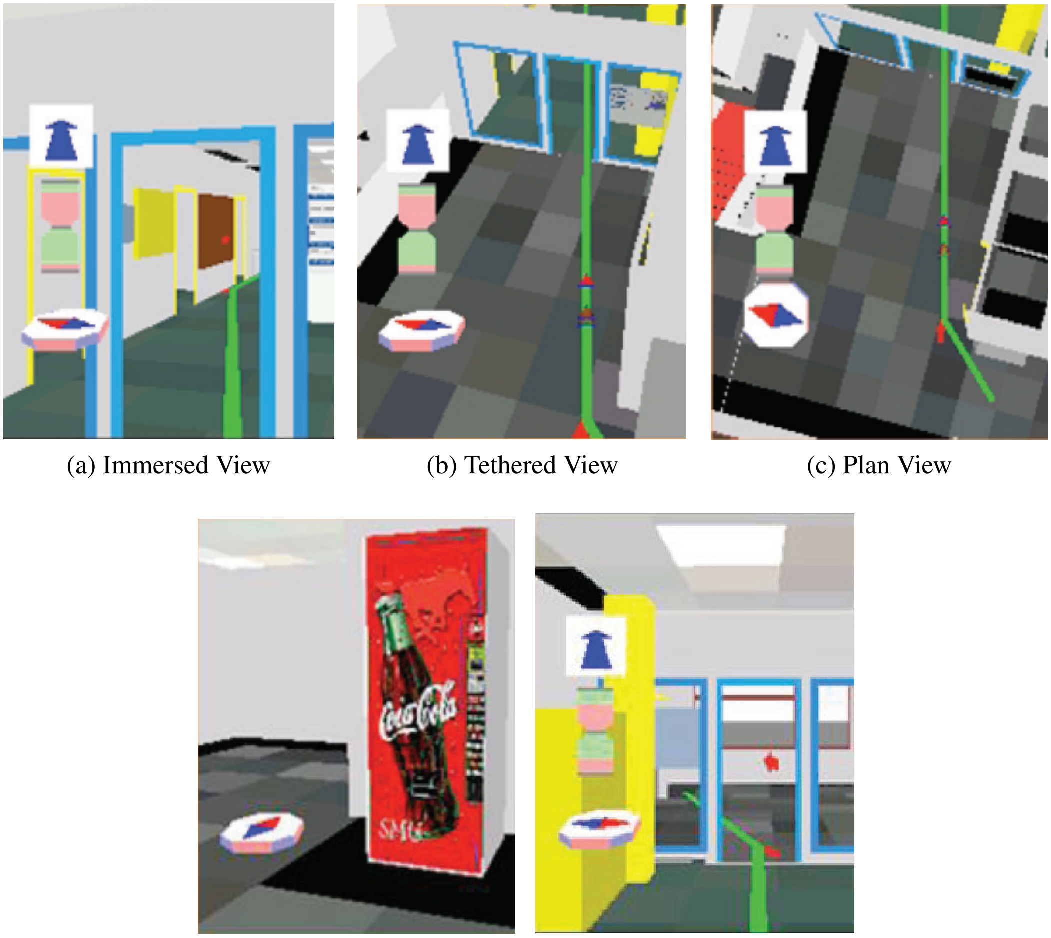


Figure 5.4: HYBNAVI is an extension of PARKNAVI [Mutafchiev, 2008].

indoor as well as outdoor positioning and navigation. Hybrid positioning is already integrated in LORIOT, but to accomplish hybrid route-finding, several components of PARKNAVI had to be extended: to enable the user to choose destinations that are outside of the currently loaded building model, HYBNAVI provides a connection to UBISWORLD which also allows to load missing models via a web connection. Furthermore, the route planner was extended by [Waßmuth, 2008] to dynamically load and parse new models while searching the shortest route between points in different buildings.

Like PARKNAVI, HYBNAVI provides spoken route instructions and different camera perspectives for the visual route description: immersed, tethered and plan view (see Figures 5.5a to c). Moreover, the system is capable of rendering eye-catching objects that can act as landmarks, e.g. soda machines or lockers (as depicted in Figures 5.5d and e). According to [Aginsky et al., 1997], the visual recognition of the connection of landmarks and directions seems to be the dominant strategy for spatial orientation of humans. Experimental studies described in [Krüger et al., 2004] and [Aslan et al., 2006] support this hypothesis, thus the ability of showing landmarks – especially those that are close to decision points – provides additional help while navigating. Furthermore, these landmarks are used in the spoken route instructions, e.g. 'turn left at the soda dispenser'.

To also allow for navigation in buildings that do not provide a positioning instrumentation, planned routes can be played back as videos showing an egocentric 3D animation of the movement through the building. Users can start, stop and rewind the video at will and can thus use the system to memorize short parts of the route



(a) Immersed View (b) Tethered View (c) Plan View

(d) Soda dispenser machine (e) Yellow lockers

Figure 5.5: HYBNAVI supports three different navigation perspectives and is capable of rendering eye-catching objects that can act as landmarks [Mutafchiev, 2008].

or try to walk the path in the same speed the video is showing it. The latter can be seen as 'reversed positioning': instead of the system recognizing where the user is, users try to position themselves where the system is showing them. Especially in this mode, HYBNAVI's capability of showing landmarks plays an important role, since it helps users to find the depicted positions.

The idea of using videos as indoor navigation aid was tested against traditional maps and printed picture sequences of decision points by [Münzer and Stahl, 2008]. An experiment with 48 participants (24 male and 24 female) was conducted, where each condition was tested with 16 participants. The result showed that the number of wayfinding errors made while using videos was significantly lower than in the two other conditions (only two out of 16 participants made critical errors in the video condition, in contrast to nine out of 16 in the picture sequence condition and seven out of 16 in the map condition).

In the scenario above, it is HYBNAVI with the included PARKNAVI component that helps Hermione to find a free parking space and that navigates her to the mall as well as back to her parked car again.

5.5 IPLAY BLUE: User-Adaptive Public Displays

The technology that provides Hermione with personalized information while poking around the shopping mall was realized by [Schöttle, 2006] during an advanced practical course. The implemented system is called IPLAY BLUE and is based on the OUT OF THE BLUE component of LOCATO.

The basic idea is to provide users with adapted content of public displays, as they are now common in public buildings. As the system should be accessible to as many people as possible, the Bluetooth based exocentric positioning method was chosen, because even cheap cellphones provide Bluetooth functionality. In order for IPLAY BLUE to work, each public display is itself Bluetooth enabled and runs the OUT OF THE BLUE core algorithm, which scans its environment for nearby Bluetooth devices and provides this information to the local infrastructure in form of events via the blackboard architecture (see Section 4.2).

IPLAY BLUE runs on a server and subscribes to the OUT OF THE BLUE events. Without further knowledge about the received Bluetooth IDs, IPLAY BLUE can at least approximate how many people are in its vicinity. In order to adapt the content of single displays however, it needs further information about who the Bluetooth ID belongs to and about special interests of the owner of the Bluetooth device. Users can therefore register with IPLAYBLUE and provide restricted access to their user profiles. For example, one might only reveal their gender in order to get informed about gender-(stereo)typical items, e.g. electronic gadgets for males and more clothing specific things for females.

In order to protect the privacy of users, their names will not appear on the public displays. Instead users can freely choose icons or pictures, for example a picture of their favorite cartoon character, which will be displayed next to relevant information. In order to make it difficult for observers to assign pictures or icons to bypassing people, pseudo information can be shown of imaginative users, such that situations are avoided in which only one picture or icon is shown.

IPLAY BLUE closely interacts with UBIDOO and can thus adapt the view of a public display in a shop to the tasks of a user or of a group of users. An example view is

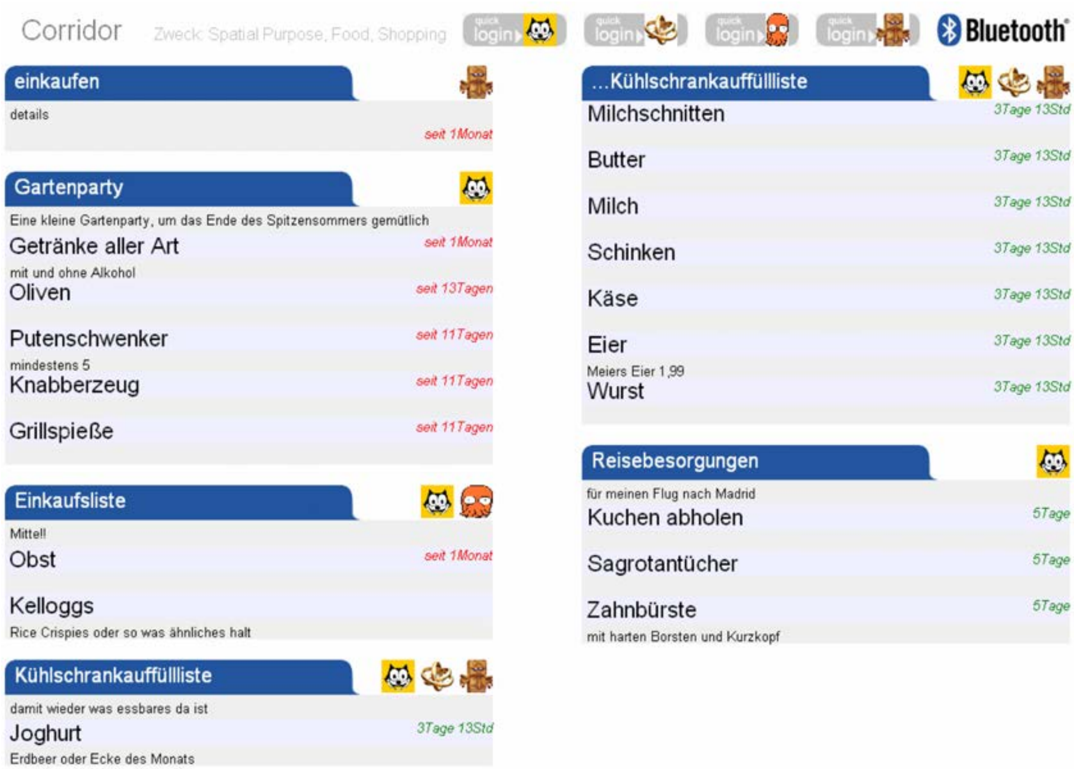


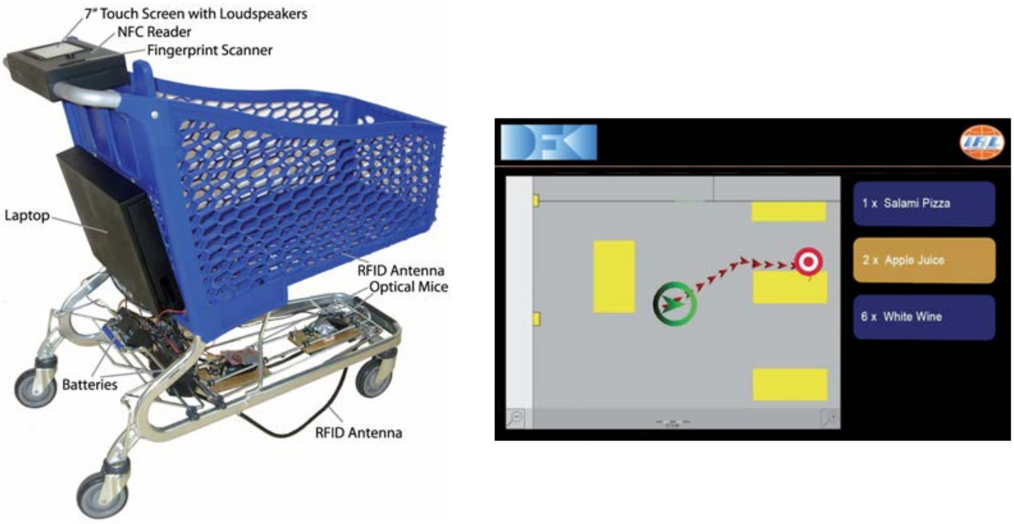
Figure 5.6: Example view of IPLAY BLUE on a public display [Schöttle, 2006].

shown in Figure 5.6. Here a collaborative shopping list is depicted, which is adapted to the group members that are currently in front of a public display. The tasks are grouped according to the assignment of each task to individuals or subgroups. The icon for each assigned group-member is shown besides a general task description.

The design of the presentation is such that more general information is shown in large fonts, so that users can grasp this information quickly while passing by a screen. More detailed information is shown in smaller fonts and are intended for users who want to focus on a specific task.

Because of these collaborative features, IPLAY BLUE is also valuable at home, to organize various household tasks. At work it can be used to organize and to inform about group meetings or about the availability of individual employees.

A special view, called Iplay Ad, was implemented for stores to show personalized advertisements. This view shows items that are on the shopping lists of users nearby the public display, without assigning them to a specific user. If only one user is detected, the list will be filled other items that are currently on sale.



(a) The IRL SmartCart (b) Screenshot of the IRL navigation service

Figure 5.7: The IRL SmartCart enables product-related navigation in shopping environments [Kahl et al., 2011].

5.6 Navigation in Retail Environments

The Innovation Retail Lab (IRL) is a collaboration of between DFKI Saarbrücken and the German retailer GLOBUS SB-Warenhaus Holding. IRL is located at St. Wendel, Germany, and focuses on application-oriented research, mostly related to intelligent shopping assistance. Developed systems get thoroughly tested not only in the lab itself, but also in a real retail store ([Spassova et al., 2009]).

Amongst various other innovations, a smart shopping cart was developed, dubbed the IRL SmartCart. Figure 5.7a shows the SmartCart and its instrumentation. To enable user interaction, a touchscreen with speakers, an NFC reader and a fingerprint scanner are mounted at the cart's handle. A laptop is currently attached to the cart as the main computing device, which also enables communication with the environment via WiFi.

Two passive RFID readers are mounted at the base of the cart to which two antennae are attached: one that enables reading RFID tags of products that are placed inside the cart, and one close to the floor, which is used for positioning via passive RFID tags. Accordingly, the lab's floor is instrumented with several hundred passive RFID tags, which were laid out in a dense grid beneath the floor cover. The IDs of all RFID tags and their position were modeled using YAMAMOTO. Unfortunately, some of the RFID tag got damaged during the installation of the floor cover, so that additional tags had been attached directly to the shelves. As the SmartCart is intended to be used in a specific shop, the complete list of RFID IDs and their coordinates can be stored directly on the cart.

The position determination of the SmartCart was realized using the geoDBN package of LOCATO, where the CPTs were adjusted to the higher confidence values of passive RFID technology, i.e. sensing a passive RFID tag results in a very high belief that the SmartCart is currently at the position of the tag. In Figure 5.7a two optical mice are shown in addition to the lower RFID antenna. These mice are intended to be used as inertial sensors, i.e. they provide information about the moving direction and velocity of the cart. The derived position information can then be fused with the RFID information through the geoDBNs. The position calculation is done on the SmartCart itself and with the allowance of the user, this position can be communicated to the shops service-infrastructure via WiFi connection.

The positioning is used to provide customers with navigation support, similar to what is described in the scenario above. Users can identify themselves by several means, e.g. with their fingerprint or with an NFC enabled customer card. This identification enables a user adaption of the cart's user interface, e.g. using bigger fonts for elderly people, as well as an adaption of the provided services, e.g. automatic warnings if a user intends to buy a product they are allergic to. A further adaption is the synchronization with the user's current shopping list. If such a shopping list is present, a user can either click on a specific item and is presented with a calculated route to the appropriate shelf, or they can chose to be navigated to all items on their shopping list. For the latter situation, the items on the list are ordered according to a predefined basic route through the shop, and the complete route is calculated by subsequently navigating to each item on this ordered list. A screenshot of the IRL navigation service can be seen in Figure 5.7b. This view is presented to a user on the touchscreen of the SmartCart.

If a customer comes close to an item on their list, the shopping cart can trigger various forms of public displays to show further navigation clues, e.g. by highlighting the product in question with a steerable projector ([Spassova, 2011]). The latter is called Micro Navigation, in contrast to Macro Navigation, which is the former described approach of guiding a user to the correct shelf.

Both, the macro and the micro navigation are intended for users who are not familiar with the shop's layout, as it is the case with Hermione in the example scenario. Customers who are familiar with the shop, can switch to a so-called Passive Navigation mode. In this mode, the cart's display shows an overview of their proximate environment and issues a reminder a soon as they come into the vicinity of an item on their shopping list. Customers can thus enjoy an exploratory shopping experience, while getting helpful assistance to do their weekly shopping (cf. [Kahl et al., 2011, Schwartz, 2010, Kahl et al., 2008, Kahl, 2007, Stahl et al., 2005]).

5.7 Location-Adaptive Human-Centered Audio Email Notification Service

The location-adaptive audio-notification service that Ron uses to keep his party music at high volume without alienating his friends is described in [Jung and Schwartz, 2007a] and [Jung and Schwartz, 2007b], although its intended use is much broader. Audio notification sounds, like telephone ring-tones, stand out because they differ significantly from other sounds in the environment. This is of course usually intended as such a sound should attract attention, however it should ideally attract the attention of the addressee of the notification and not the attention of everybody in the vicinity. In some situations, like business meetings or conferences, notification sounds are highly inappropriate. The common solution is to use vibration alerts, but since sound is pressure oscillation transmitted through air (or any other medium), even vibration alerts are audible, especially in quiet surroundings. Awkward moments usually occur in these situations because the source of the sound is clearly locatable and even if not, the following actions, like desperately trying to cancel a call or leaving the room, give away the culprit of the diversion.

The basic idea of the audio notification system is to mask audio-signals by integrating them into artificial ambient soundscapes, i.e. soundscapes that are played through an audio system, such as functional music or so-called Muzak. Functional music is low in complexity, only slightly above the environmental noise level and in a tempo that is close to the resting pulse rate. It is especially composed to not distract people and to have a calm and smoothing effect. For the implemented Ambient Email Notification service (AEMN) three such compositions were recorded and the service has full control over single tracks of these recordings, i.e. the service can switch on or off specific instruments, like piano, guitar, strings or even the hi-hat of the drum section. Users can choose their personal notification instrument and can choose their preferred ambient music. The AEMN service is web-based and is connected to the local blackboard architecture as well as to a multi-speaker sound-system, over which the ambient soundscapes can be played back. The current location of each user is determined through LORIOT, which runs on the mobile device of the user and forwards the positioning information to the blackboard (if allowed by the user).

If a user enters an empty room, their preferred ambient soundscape will be started. If other people are already present, the current soundscape will be kept, but the notification instrument of the user will from now on be omitted from the currently playing soundscape. AEMN constantly checks the inbox of user-specified email accounts and checks via configurable filters if an email arrives that requires notification of the user. If this happens, AEMN will insert the user's notification instrument into the soundscape at the speaker that is nearest to the user and in a musical fashion, such that other people in the room will perceive the audio signal as a part of the composition. The nearest speaker to the user is chosen in order to ease the perception of the signal.

The perception of these embedded audio signals was tested in a user study with 25 persons. Each participant learned two notification instruments (piano and drums) and a conventional alarm sound (a knocking sound) in a preparatory phase. During the experiment, the participants had to perform mental arithmetic under time-pressure, while soundscapes where played, which contained the learned notification instruments as well as the conventional alarm sound. The participants were instructed to click on a specific button as soon as they perceived either one of the notification instruments or the conventional alarm sound. The conventional alarm sound was recognized in 79% of all cases. The drum-sound notification was even recognized in 86% of all cases, while the piano notification reached 78% and was thus slightly lower than the conventional alarm sound. The reaction times, i.e. the time it took participants to click on the button after the notification sound was first played, were also measured. As it was the case with the perception of the sounds, the drum notification provided the best result: 2.1 seconds in average. With 2.54 seconds, the conventional sound provided a slightly higher average reaction time. The piano notification led to an average reaction time of 6.59 seconds. All in all, the performance of this type of notification is comparable or even better than conventional audio notification, without the risk of embarrassing moments (cf. [Jung, 2009]).

Part IV

Conclusion

6 CONCLUSION

In this thesis, the Always Best Positioned paradigm was defined (see Section 1.1.6) and the Localization Toolkit LOCATO was developed, which enables the efficient development of egocentric and exocentric positioning systems that can be executed either onboard or offboard. LOCATO provides two novel core algorithms, which are designed according to the Always Best Positioned paradigm:

- **Frequency-Of-Appearance Fingerprinting** omits the usage of signal-strength information in fingerprints. It thus provides higher robustness against environmental influences and allows to create and share device-independent fingerprints.
- **Geo-Referenced Dynamic Bayesian Networks** enable the easy fusion of different sensor technologies. They mimic the behavior of particle filters by creating hypotheses for possible positions, but in contrast to regular particle filters they automatically adapt the number of hypotheses according to the sensors in the environment and are easy to extend with additional sensor technologies.

Both algorithms are optimized for resource-limited devices, such as mobile phones. Three positioning systems were implemented using LOCATO:

- **Out of the Blue** is a cost effective offboard/exocentric indoor-positioning system, designed for user-adaptive public displays.
- **UbiSpot** is an opportunistic onboard/egocentric outdoor/indoor positioning system following the Always Best Positioning paradigm. It uses mobile-phone network-cells, WiFi access points and Bluetooth devices to determine its own position and reaches room-level accuracy.
- **Loriot** is a high accuracy onboard/egocentric outdoor/indoor positioning system for instrumented environments, which uses infrared beacons and active

RFID tags. The system's accuracy is highly configurable through variation of the density and mixture of the deployed senders. LORIOT can achieve an average position-accuracy of 1 meter.

Both onboard/egocentric positioning systems were rigorously evaluated, according to their specifications. As UBISPOT was designed to reach room-level accuracy, it was evaluated in a dense environment with small rooms. LORIOT was designed for high accuracy, real-time positioning and was thus evaluated using natural, step-accurate traces as ground-truth. Since the used senders tend to overreach, the evaluation took place in a wide-spaced environment, which encouraged overreach.

Furthermore, six applications were presented that either base on one of the developed positioning systems or were designed using LOCATO. The parking-deck navigation was developed in cooperation with BMW and was deployed and tested in the P20 multistory car park at Munich Airport Center. The shopping-cart positioning was developed in cooperation with the IRL in St. Wendel.

6.1 Scientific Contributions

In order to develop the toolkit LOCATO, the following research questions were answered:

- **What are the basic methods for position determination in natural organisms?** In Section 2.1.1, concepts of neuropsychology were examined to derive a classification of senses, which were identified as a basic component for self-positioning. Furthermore, proximity sensing was identified as a fundamental positioning method, and the occurrence and importance of sensor fusion in animals was explored on the example of ants.

- **How can natural self-position awareness be replicated through methods of Artificial Intelligence?** In Section 2.2, the findings from natural position determination were transferred into the field of Artificial Intelligence through the use of agent theory. Based on the classification of senses, a classification of sensors was derived.

- **How can technical positioning methods be classified and what are the implications of the classification?** In Section 2.3, positioning systems were classified by analyzing the possible spatial distributions of sensors, senders and computational devices. Four basic designs for positioning systems could be derived:

- egocentric (or onboard/egocentric)
- exocentric (or offboard/exocentric)
- offboard/egocentric
- onboard/exocentric

- **How should a positioning system be designed to protect the privacy of its users?** In Section 2.3.2.5, the data-flow of the four basic designs was analyzed with respect to the implications on the privacy protection of users. The onboard/egocentric approach was identified as being the most privacy-protecting design.

- **What are possible methods to build positioning systems following the Always Best Positioned paradigm ?** In Section 2.6, Kalman filters, particle filters and dynamic Bayesian networks were analyzed with respect to their suitability for the Always Best Positioned Paradigm. Dynamic Bayesian networks were identified as being the most general concept and thus being the preferred candidate.

- **How far do state-of-the-art positioning systems comply with the derived design criteria and the Always Best Positioned paradigm?** In Chapter 3, the state of the art of positioning systems was analyzed, and the discussed systems were classified regarding the four basic designs and their ability for sensor fusion.

- **How can positioning systems be evaluated?** Based on the analysis in Chapter 3, two new systematic evaluation-methods for positioning systems were developed. Each method was designed to emphasize the possible weak spots:

  - Section 4.3.4 describes an evaluation method for positioning systems based on fingerprinting. As fingerprinting provides an accuracy of several meters, such systems should be evaluated according to their ability to determine if a user is in a specific meaningful area, e.g. a room. In contrast to other evaluation methods in literature, which measure the distance of the derived position to known reference points, this evaluation tested the success rate of determining the correct room in subsequent measurements. The evaluation was conducted in an environment with small rooms, which were close to each other and thus maximizing the probability of failure.
  - Section 4.4.4 describes an evaluation method for real-time high accuracy positioning systems. For the first time in literature, natural footstep-accurate traces were used as ground truth for the evaluation. The footstep accuracy ensures that the ground truth has a higher accuracy than what

can be derived by the positioning system and the natural traces ensure that test users do not consciously or unconsciously adapt their positions to the capabilities of the positioning system.

6.2 Impact on Industry, Press and Research Community

A first prototype of LORIOT was presented at the industrial congress Advanced Navigation at the Kempinski Hotel, Berlin (invited speaker together with Norbert Reithinger). LORIOT was also presented on the CeBit'07 exhibition in Hanover with follow-up press coverage in radio (WDR Computer Club), television (RTL Nachtjournal), and newspapers (FAZ). LOCATO is also one of the building-blocks of the startup company Schwartz&Stahl indoor navigation solutions[1], which was founded in December 2008.

Furthermore, the Always Best Positioned paradigm as well as UBISPOT and LORIOT were presented on the seventh info-forum of the SmartFactory^{KL} in Kaiserslautern, Germany.

Together with researchers from DFKI Saarbrücken, the Helsinki Institute for Information Technology and the University of Haifa, the international workshop on Location Awareness for Mixed and Dual Reality (LAMDa) was founded in conjunction with IUI'11, in which the impact of positioning systems on mixed and dual reality was discussed. The workshop will be repeated in conjunction with IUI'12. An invitation to present the findings of this thesis at the Ubiquitous User Modeling Workshop at the University of Haifa followed the fruitful collaboration at LAMDa'11.

6.3 Outlook

- UBISPOT is currently extended through NFC readers, which are becoming more common in smart phones. The basic idea is to provide NFC tags at door sills, which a user can read in by swiping their phone over them. The reading of such an NFC tag will be incorporated into the current fingerprint with a very high weight, causing UBISPOT to choose that fingerprint as the current location and automatically updating the fingerprint with the most current readings of all other sensors. As a user will not always read NFC tags when changing their location, the weight will be gradually reduced over time. The rate of

[1] http://schwartz-stahl.de/

this weight decline will be a function of accelerometer measurements, i.e. the weight will be kept up so long as a user does not move. The incorporation of the NFC sensor will ease the process of crowd-sourcing up-to-date reference fingerprints.

- The upcoming 4G cells will provide higher data-rates, opening up new possibilities for Always Best Positioning systems. LOCATO will be extended to automatically download new sensor extensions on the fly, i.e. new geoDBN templates incorporating new sensors when they become available. This could also be done in a Web2.0 fashion, or in the sense of application stores, where developers can upload their designed geoDBNs for the community.

- The geoDBN core algorithm will more over be refined to further ease the process of adding new sensors. This can possibly be done by adding new sensor nodes as time-slices rather than as evidence nodes. The inter-time-slice CPTs would then have to be adapted accordingly, which needs further research. This approach would also ease the process of sharing new sensor nodes.

- IMU sensors can already be integrated using the geoDBN core algorithm. Modern smart phones already provide some of the sensors that can usually be found in IMUs, but the accuracy of the sensors is not as high as a commercial grade IMU. Further research is needed on how to refine inertial positioning using smart phone sensors and what would be a good complementary sensor to gain the required position fixes.

- In analogy to Car2Car, where highly equipped cars can share their sensor data with less capable cars, the Always Best Positioned paradigm can be extended to incorporate other users' position information. [Rosa et al., 2011] (see Section 3.1.11) proposed a relative positioning system using the WiFi capabilities of mobile devices. If one or more of these devices know and share their own position, this approach can be extended such that other devices can calculate their position in the same coordinate system as these already positioned devices. When using a standardized protocol to exchange such data, such as proposed by [Gschwandtner and Schindhelm, 2011] (see Section 3.1.3), geoDBNs can be used to realize such a positioning.

Part V

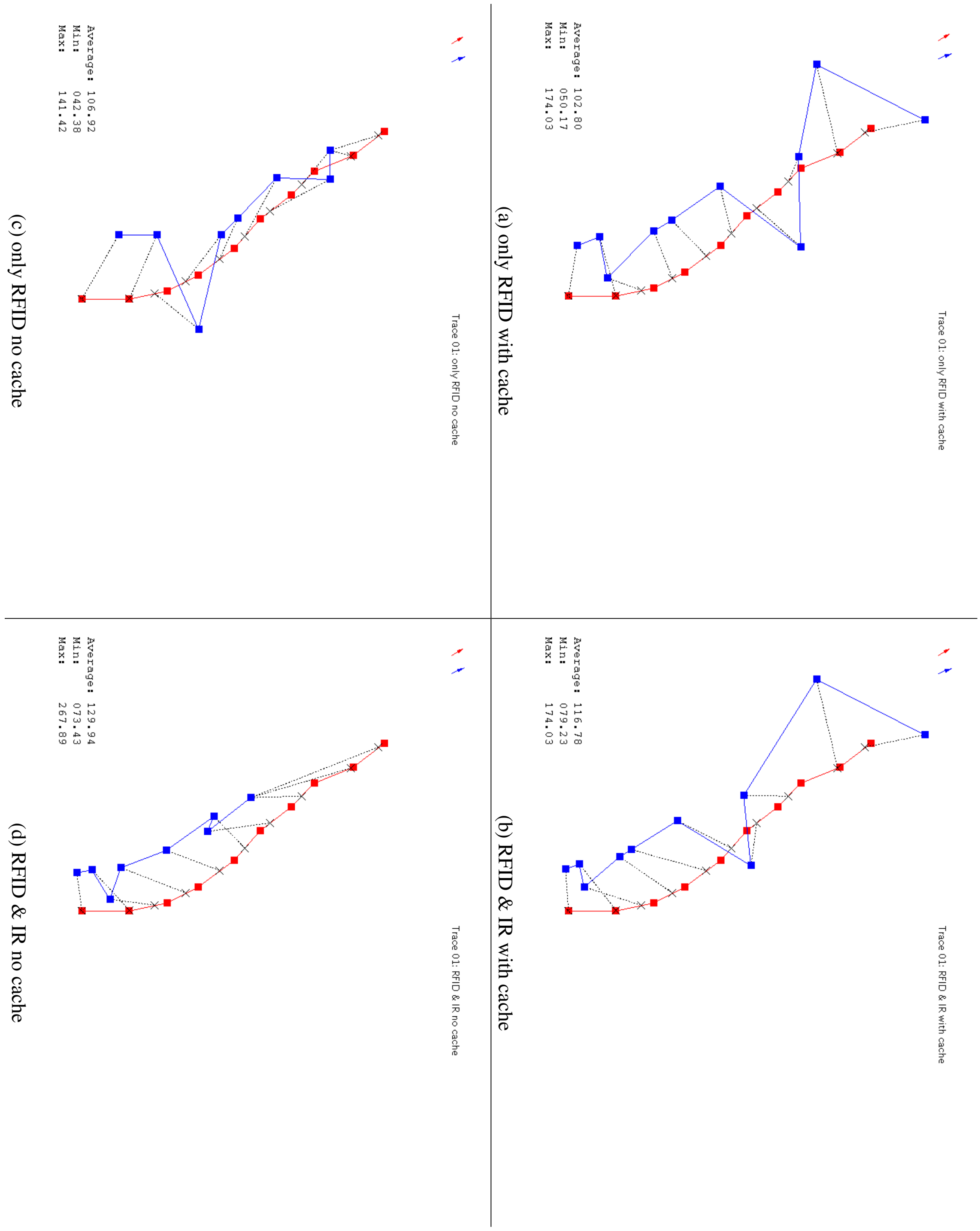
Appendix

A Appendix - Loriot

A.1 Evaluation Traces

The following figures show the results of the evaluation of LORIOT. For each of the 16 traces the four conditions (only RFID with cache, RFID & IR with cache, only RFID no cache, RFID & IR no cache) are shown in separate figures. Each figure shows the ground-truth as red squares connected by a red line. The interpolated positions are marked as black crosses. The blue squares show the user positions as derived by LORIOT (system positions). The dotted black lines lead from each system position to their corresponding interpolated user position on the ground-truth. In the top left corner of each figure, the general direction of the system trace and the ground-truth trace is shown. In the bottom left corner, the maximum, minimum and average error distance are shown.

A.1.1 Traces for only RFID and RFID & IR Conditions



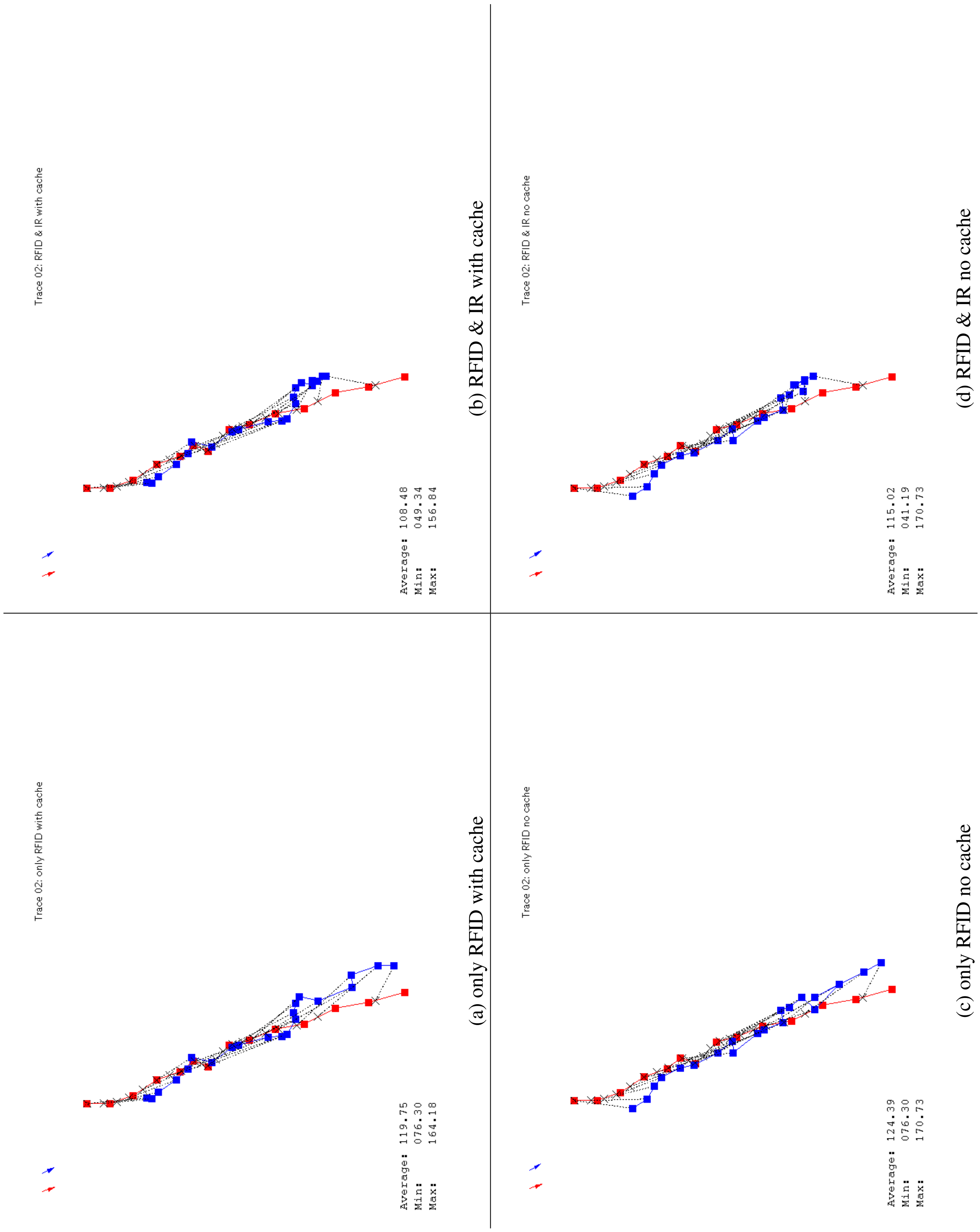
(a) only RFID with cache

(b) RFID & IR with cache

(c) only RFID no cache

(d) RFID & IR no cache

Figure A.1: Trace 01



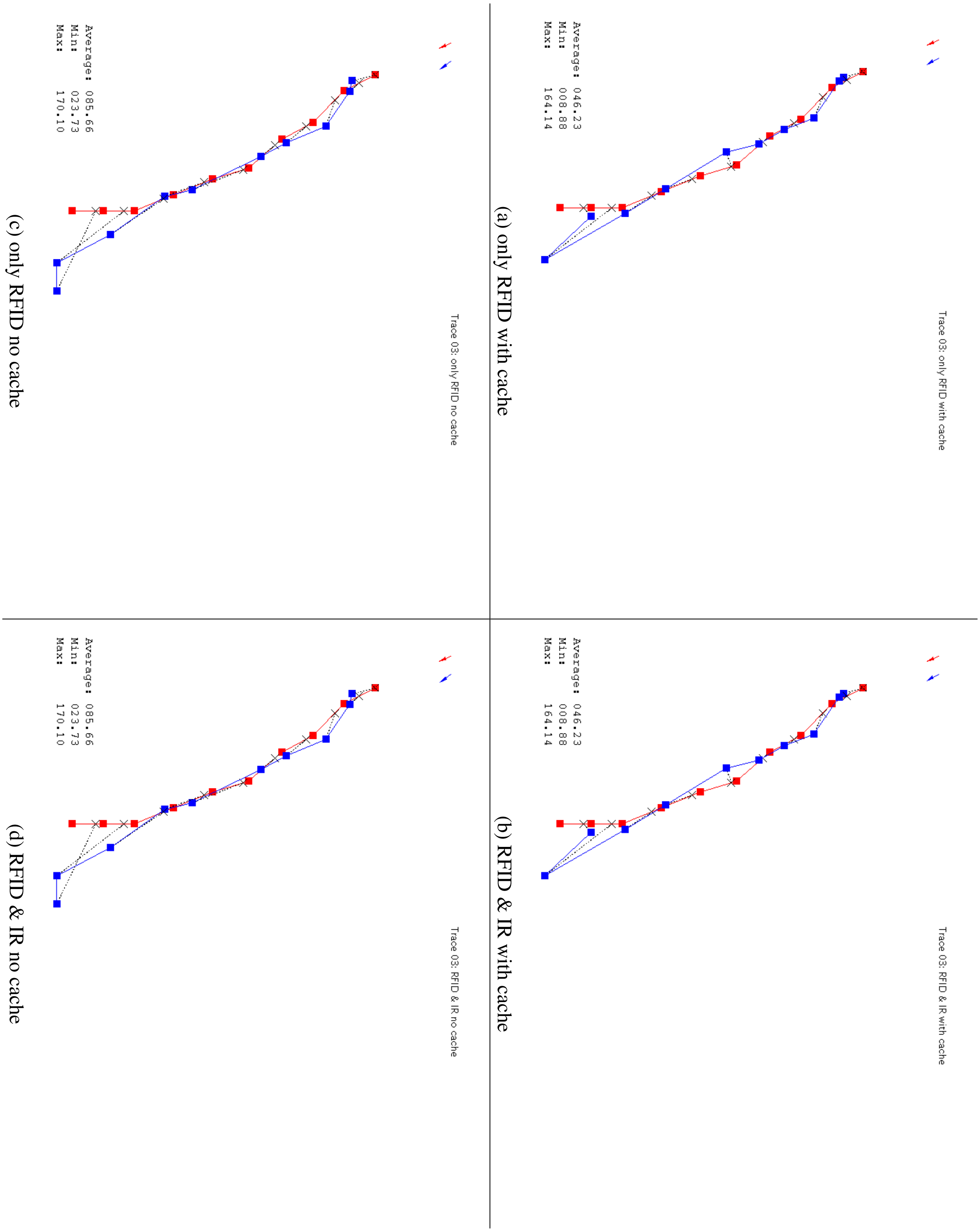
(a) only RFID with cache

(b) RFID & IR with cache

(c) only RFID no cache

(d) RFID & IR no cache

Figure A.2: Trace 02



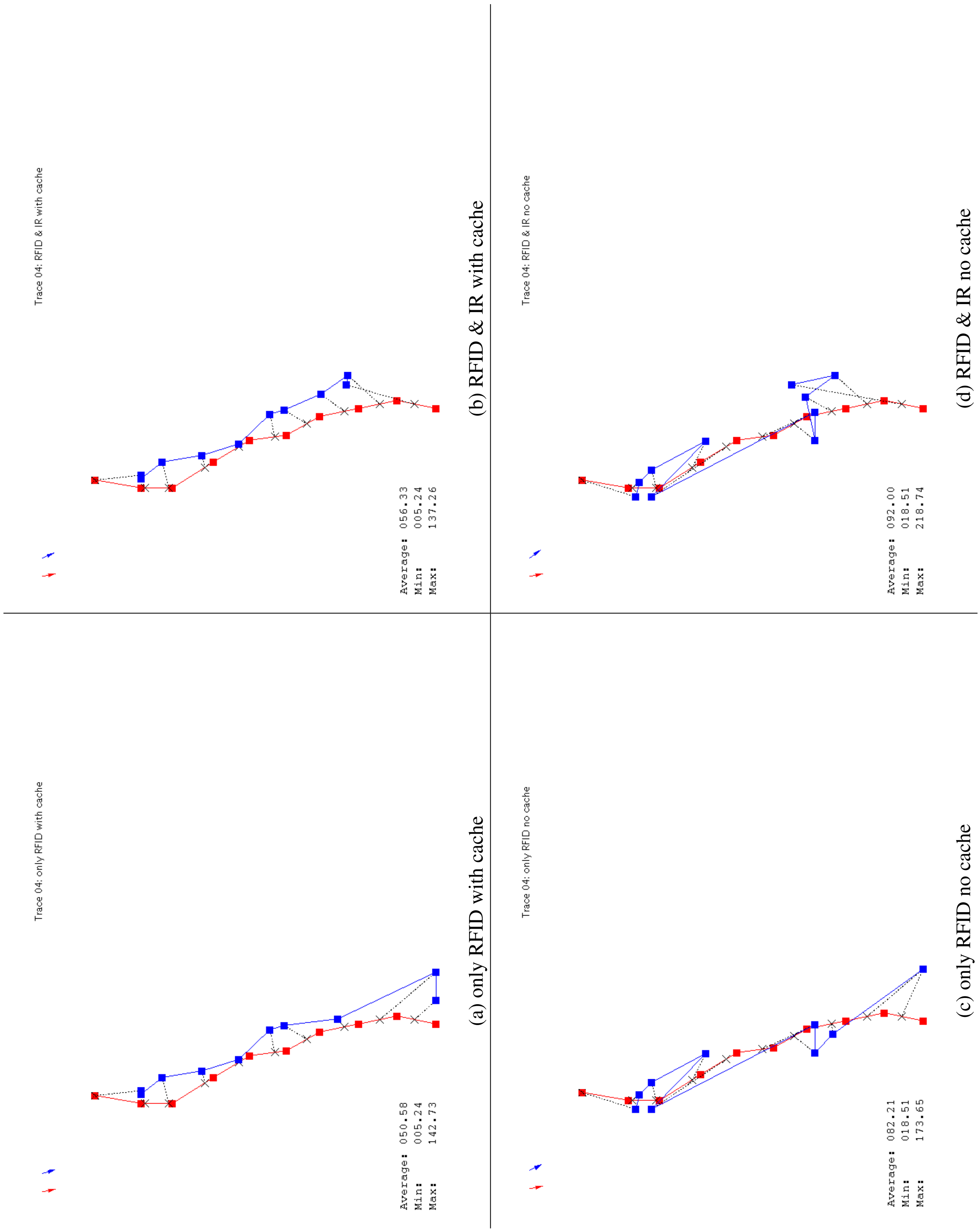
(a) only RFID with cache

(b) RFID & IR with cache

(c) only RFID no cache

(d) RFID & IR no cache

Figure A.3: Trace 03



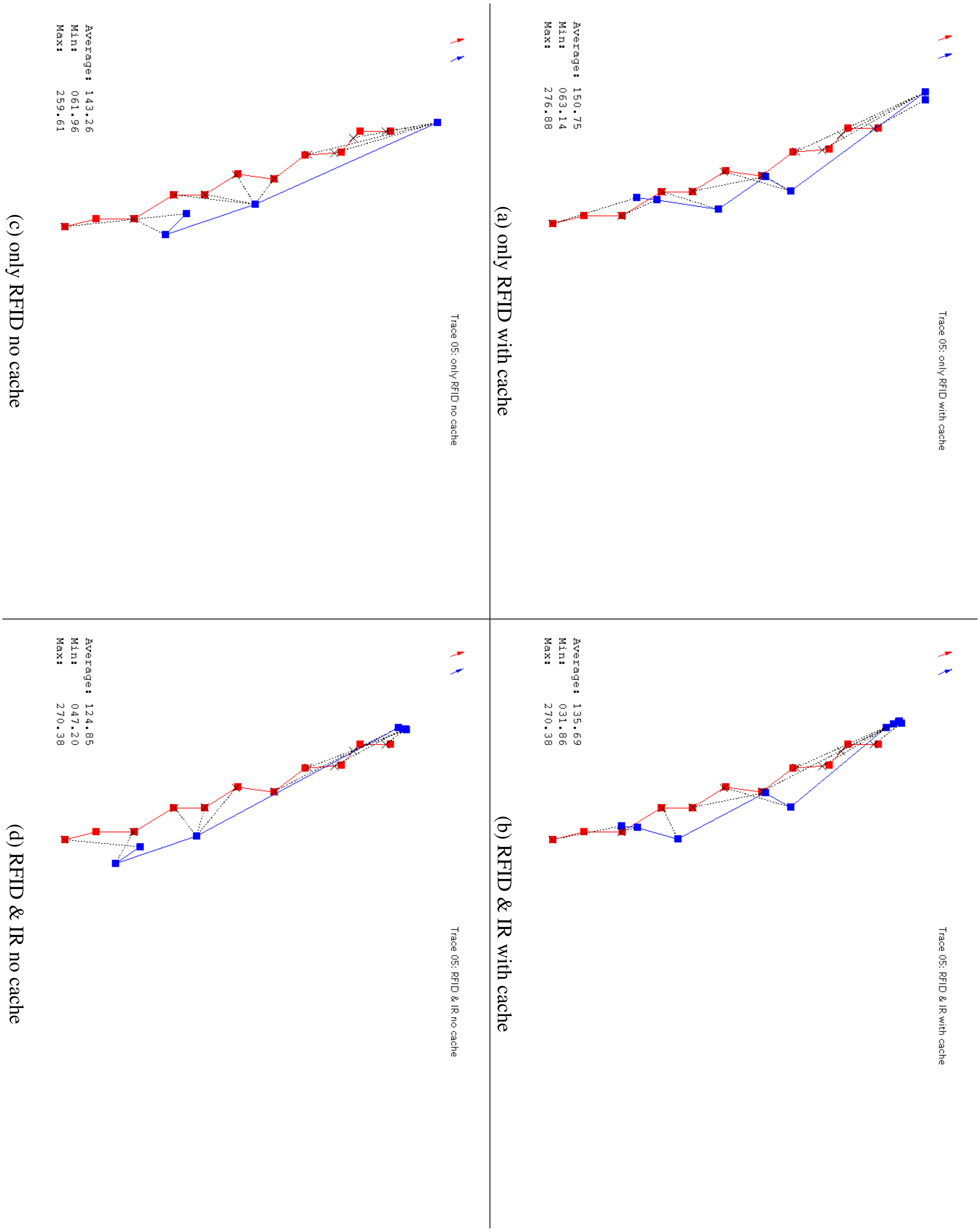
(a) only RFID with cache

(b) RFID & IR with cache

(c) only RFID no cache

(d) RFID & IR no cache

Figure A.4: Trace 04



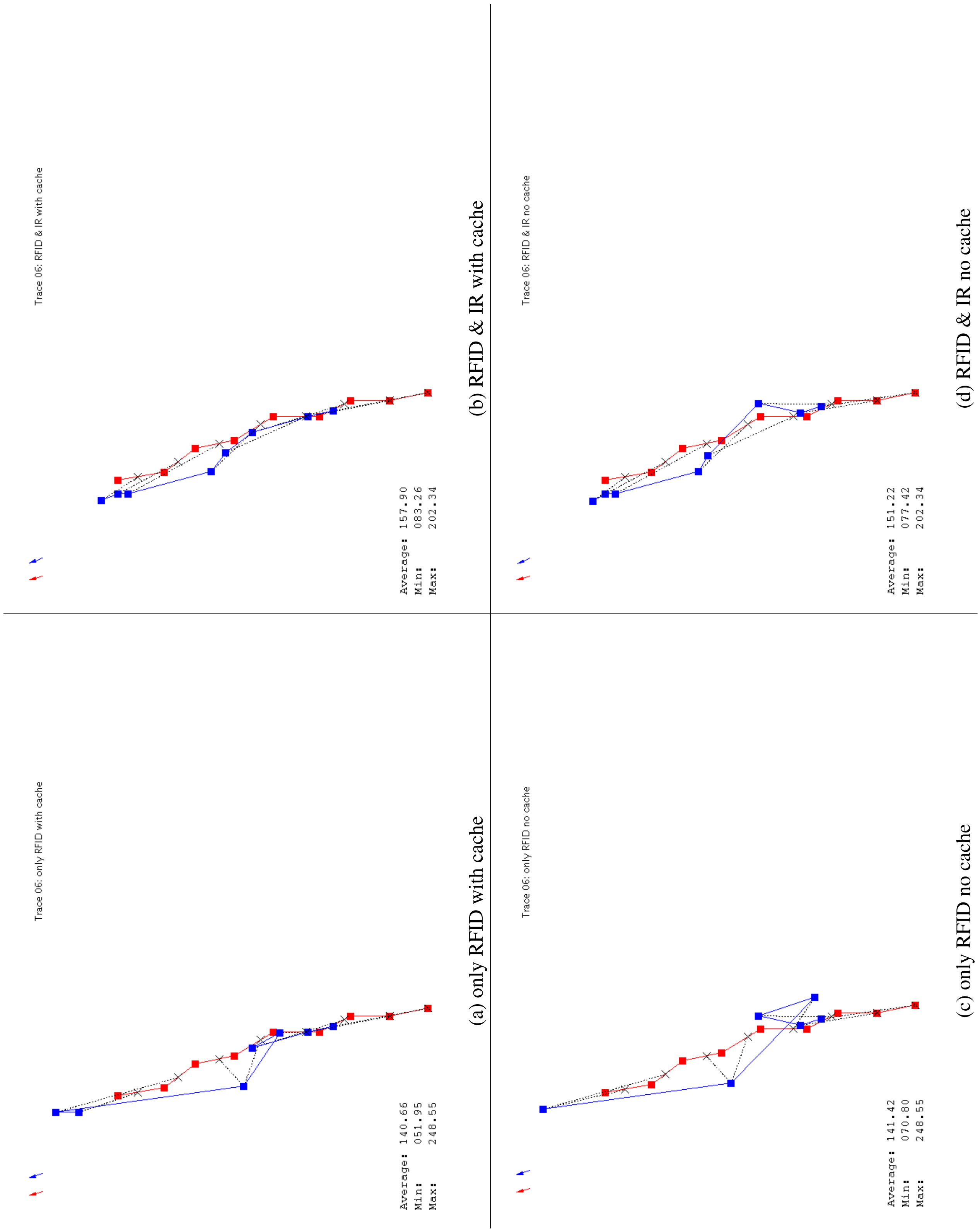
(a) only RFID with cache

(b) RFID & IR with cache

(c) only RFID no cache

(d) RFID & IR no cache

Figure A.5: Trace 05



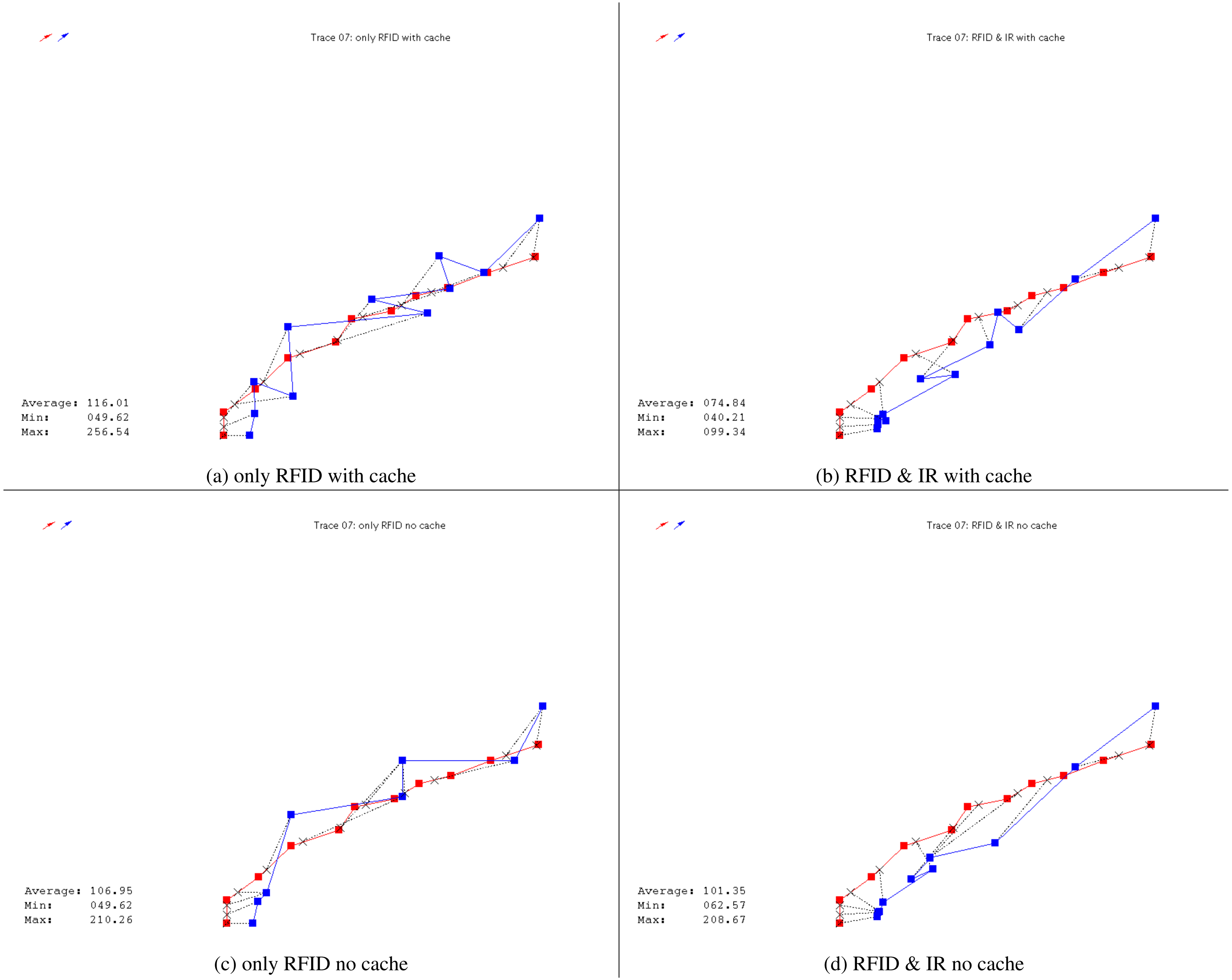
(a) only RFID with cache

(b) RFID & IR with cache

(c) only RFID no cache

(d) RFID & IR no cache

Figure A.6: Trace 06



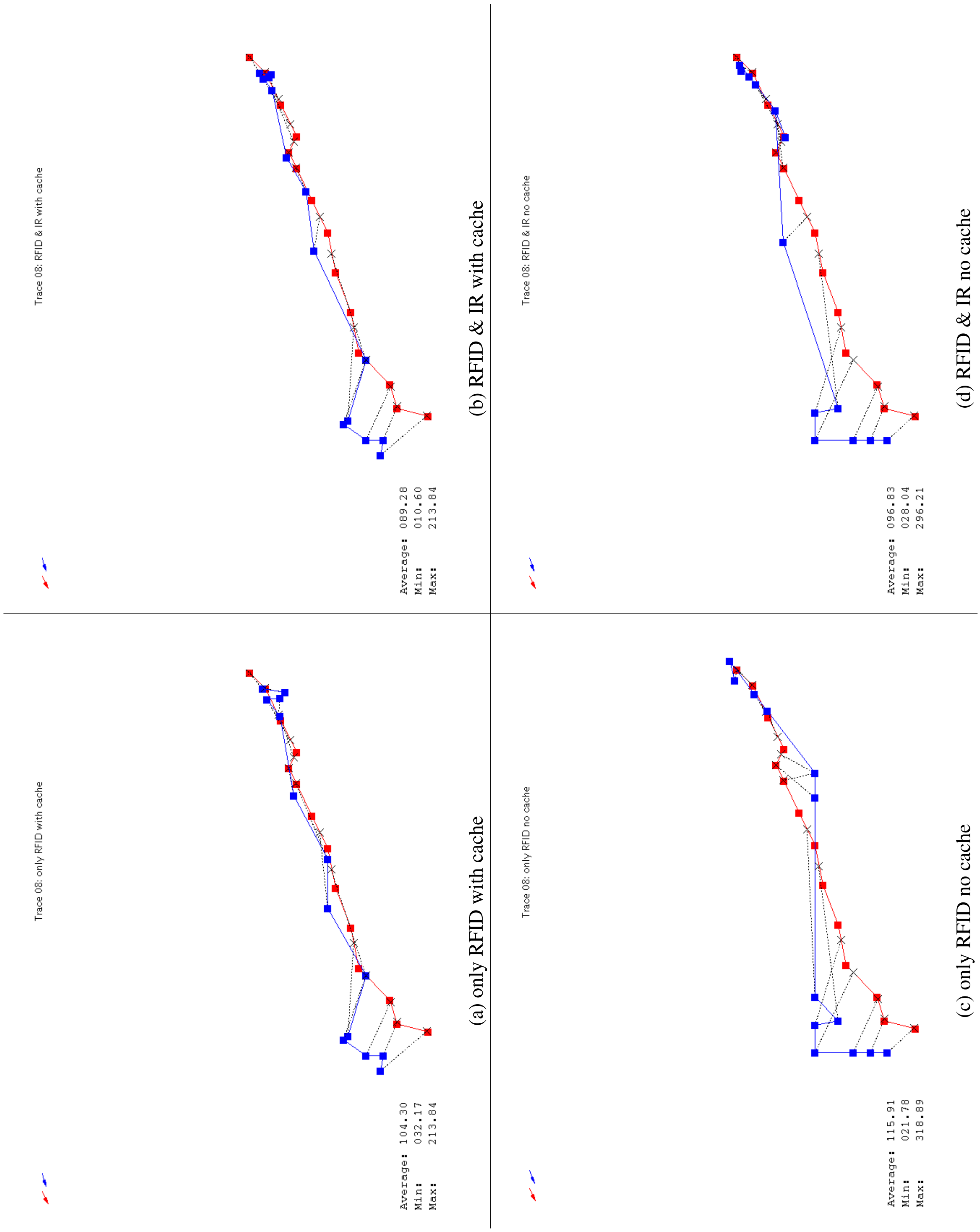
(a) only RFID with cache

(b) RFID & IR with cache

(c) only RFID no cache

(d) RFID & IR no cache

Figure A.7: Trace 07



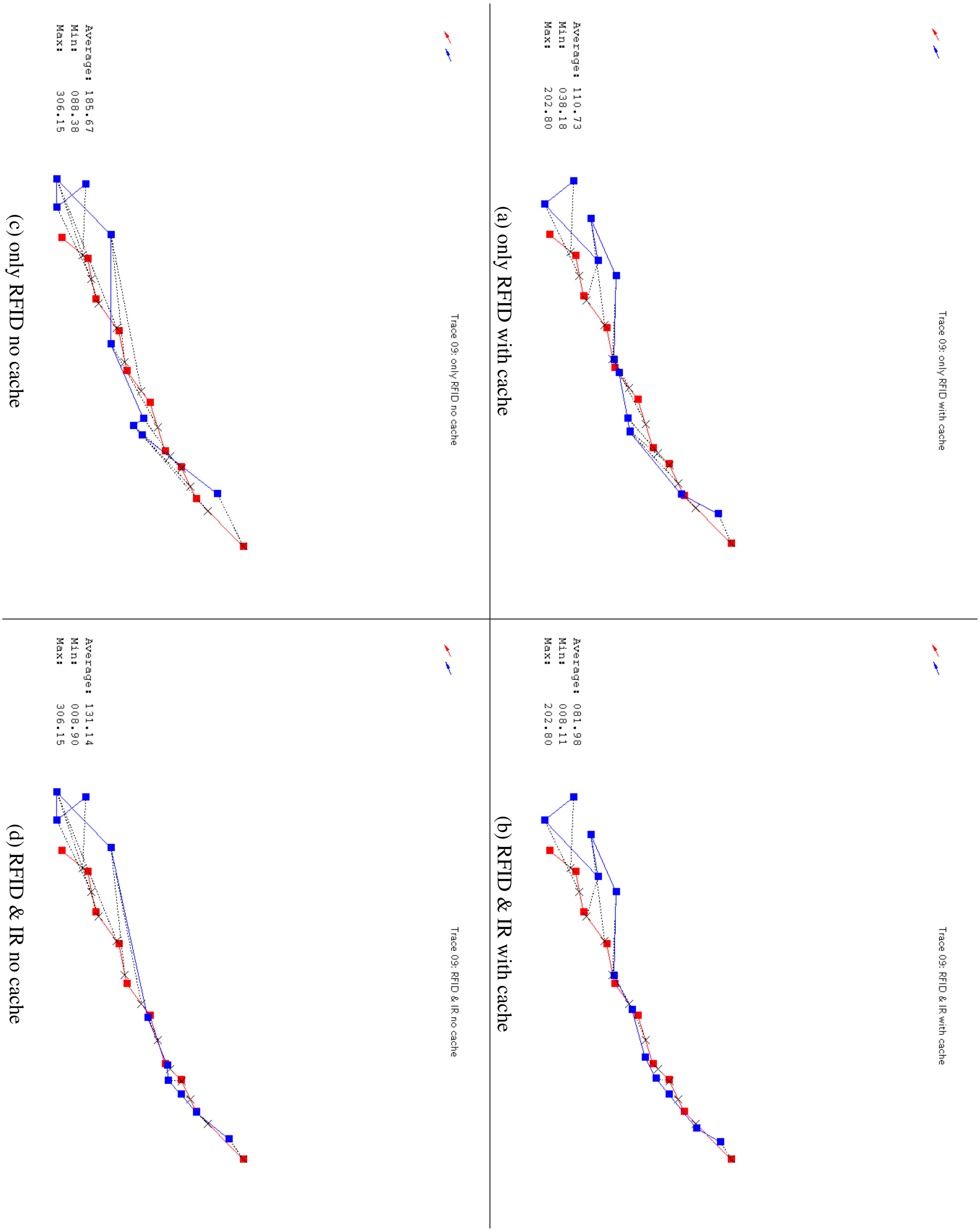
(a) only RFID with cache

(b) RFID & IR with cache

(c) only RFID no cache

(d) RFID & IR no cache

Figure A.8: Trace 08



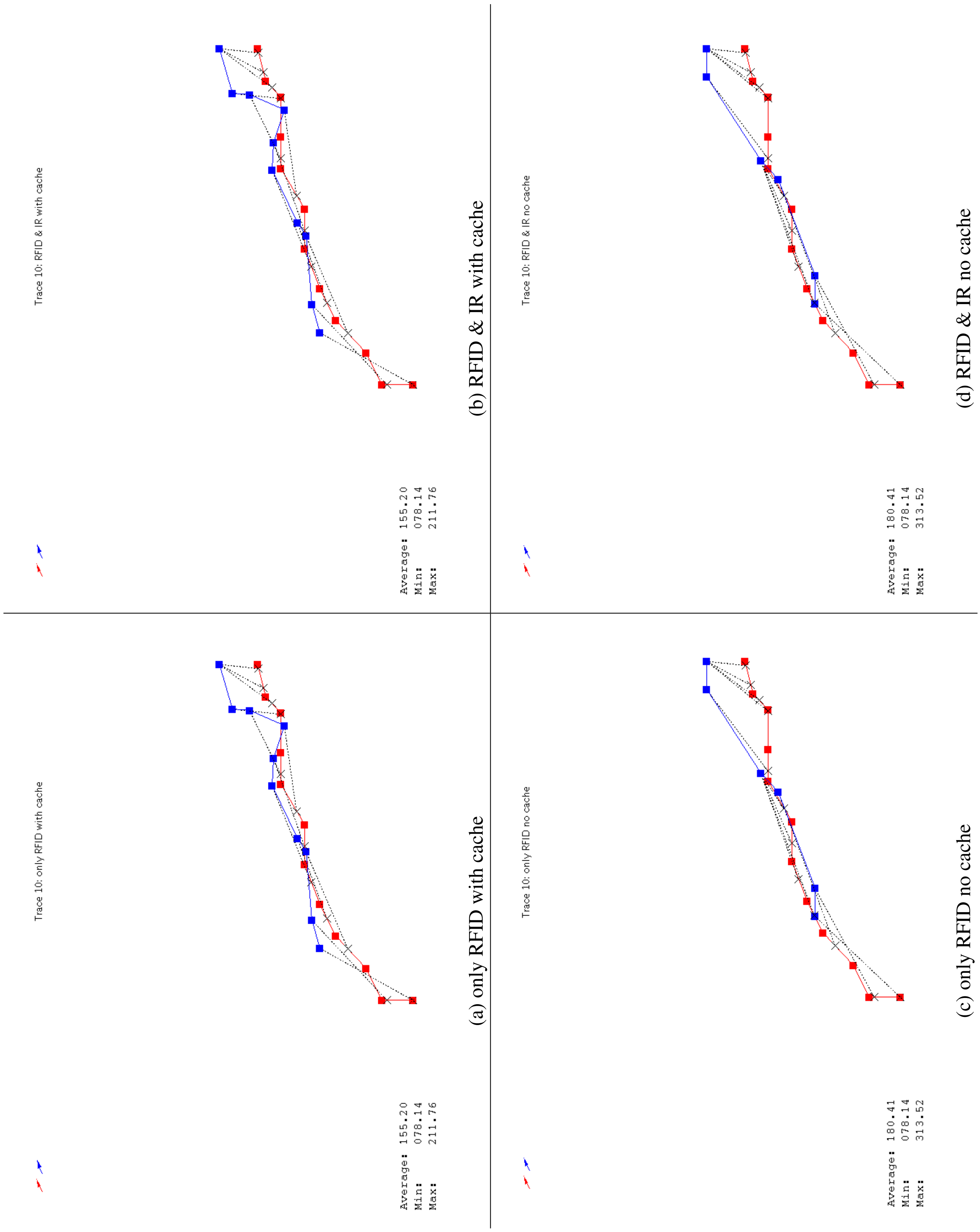
(a) only RFID with cache

(b) RFID & IR with cache

(c) only RFID no cache

(d) RFID & IR no cache

Figure A.9: Trace 09



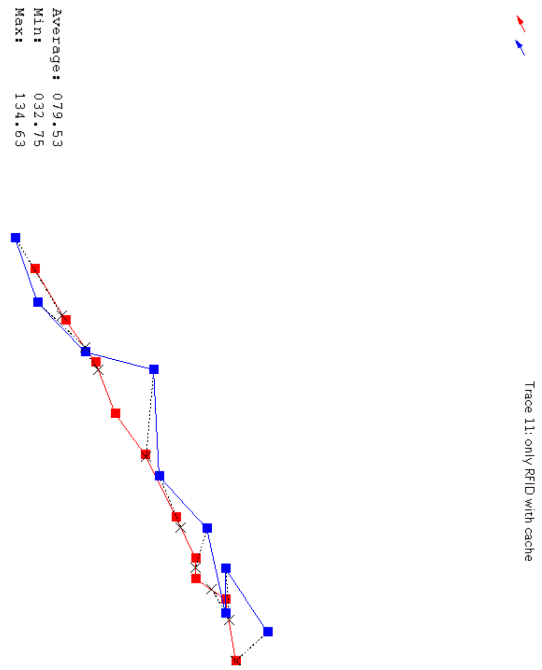
(a) only RFID with cache

(b) RFID & IR with cache

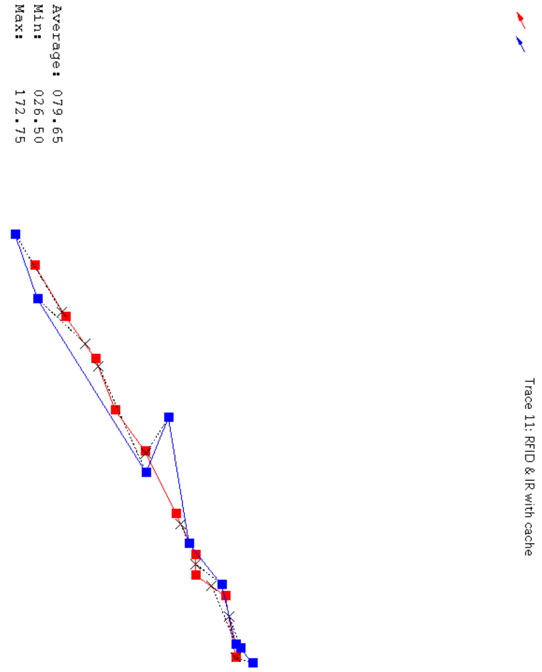
(c) only RFID no cache

(d) RFID & IR no cache

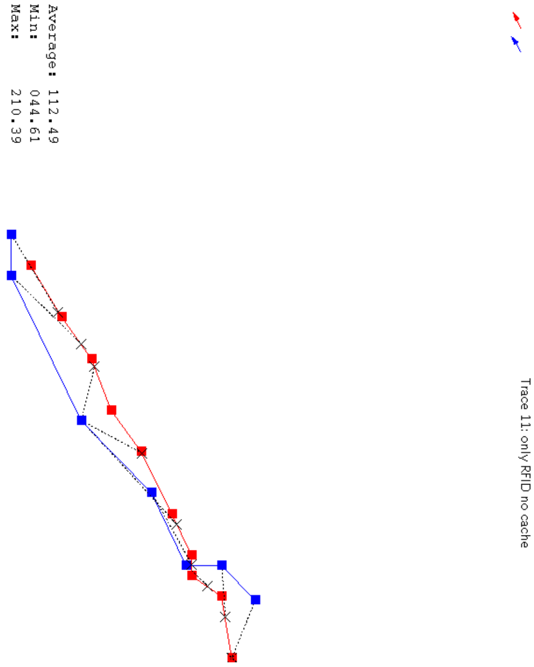
Figure A.10: Trace 10



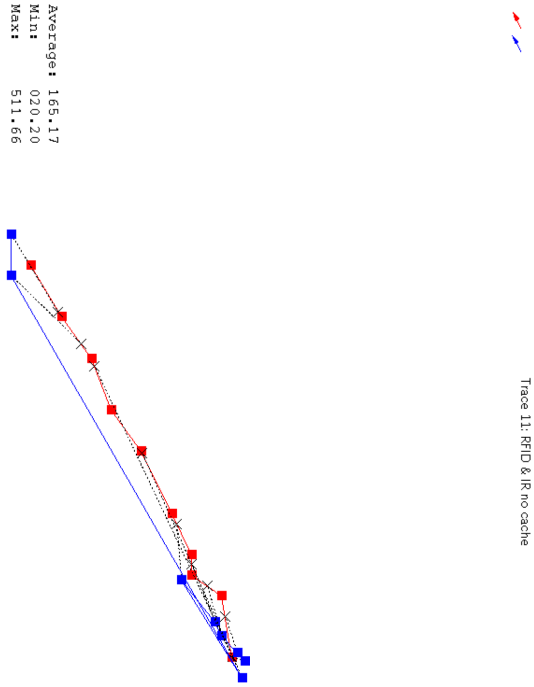
(a) only RFID with cache



(b) RFID & IR with cache

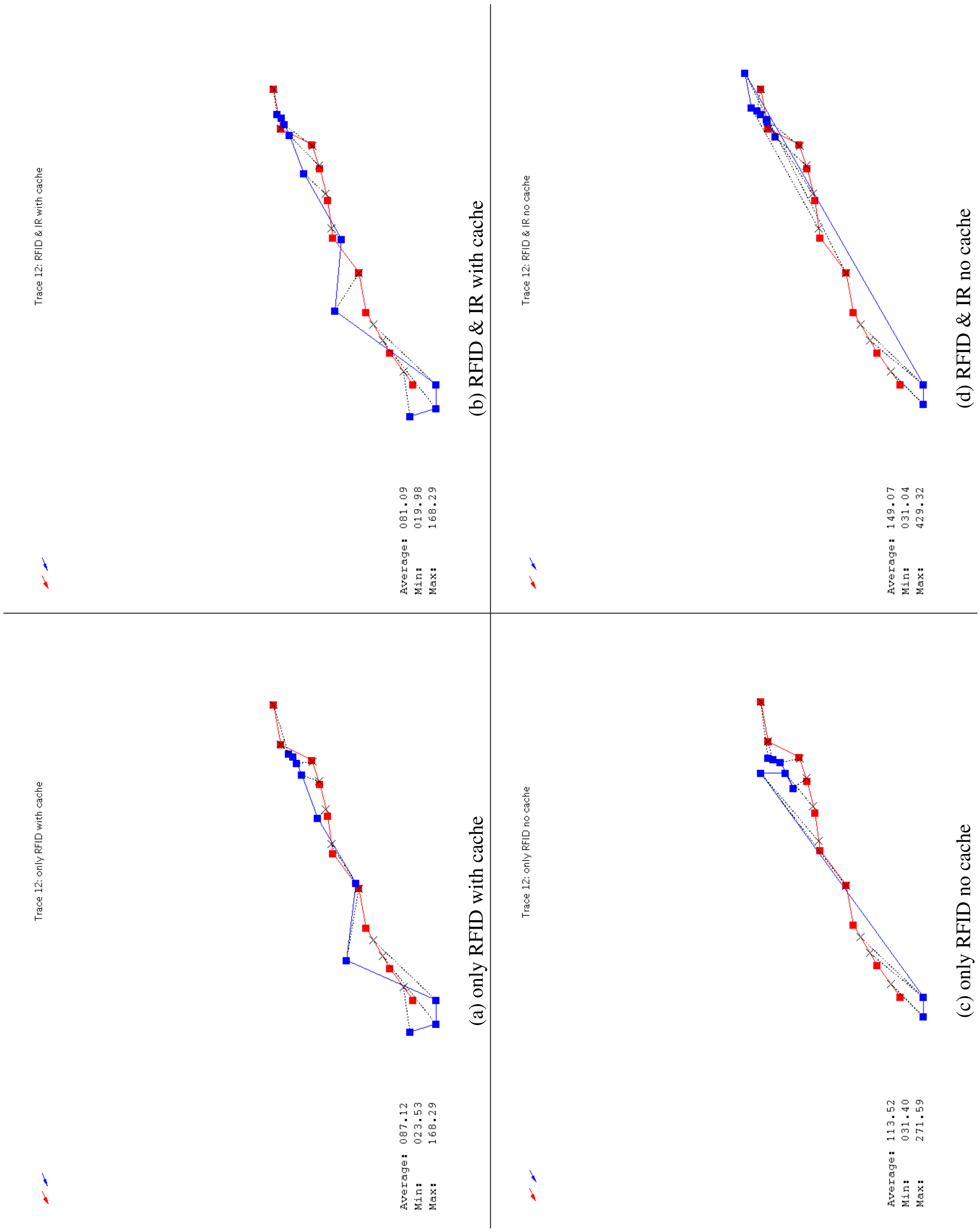


(c) only RFID no cache



(d) RFID & IR no cache

Figure A.11: Trace 11



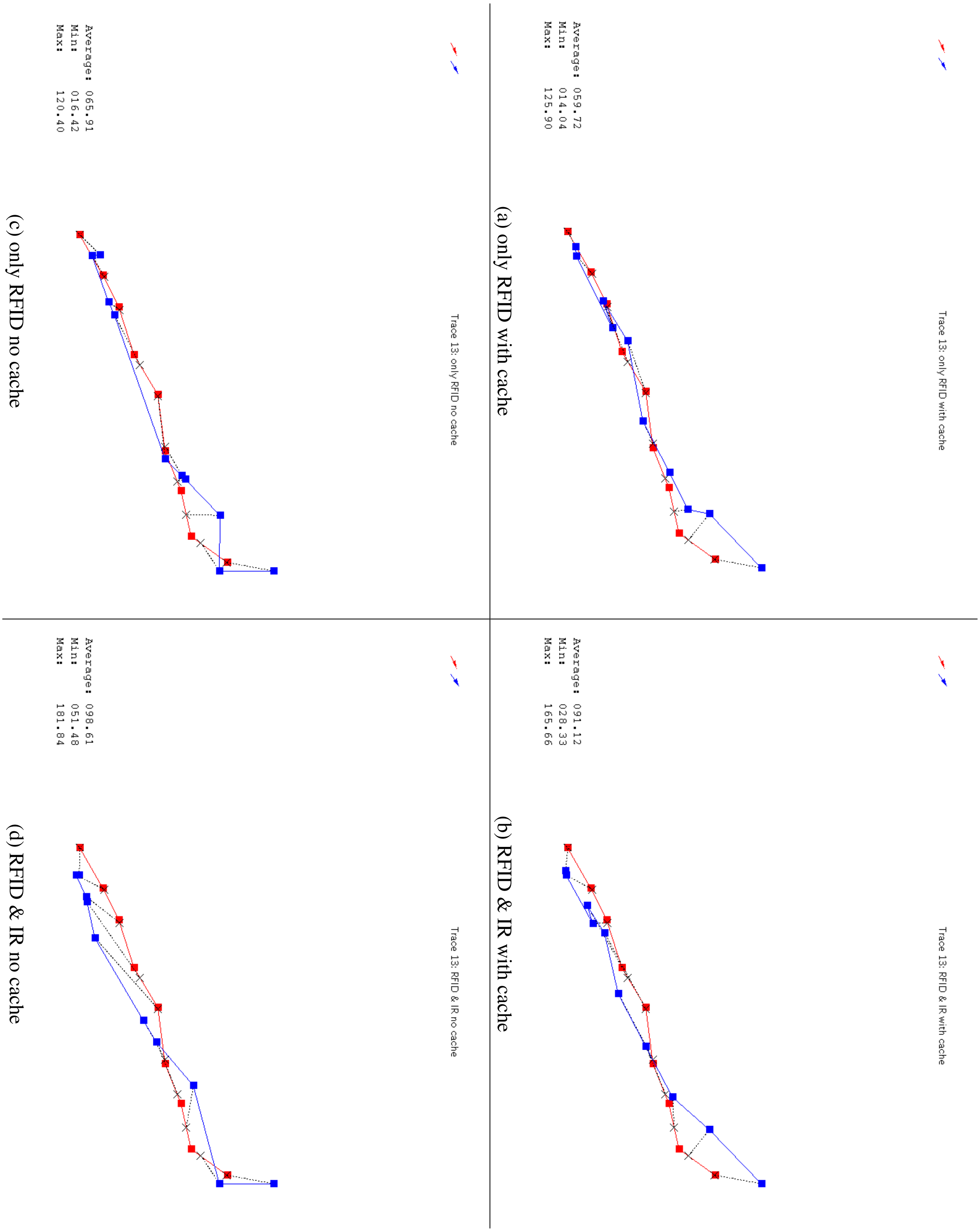
(a) only RFID with cache

(b) RFID & IR with cache

(c) only RFID no cache

(d) RFID & IR no cache

Figure A.12: Trace 12



(a) only RFID with cache

(b) RFID & IR with cache

(c) only RFID no cache

(d) RFID & IR no cache

Figure A.13: Trace 13

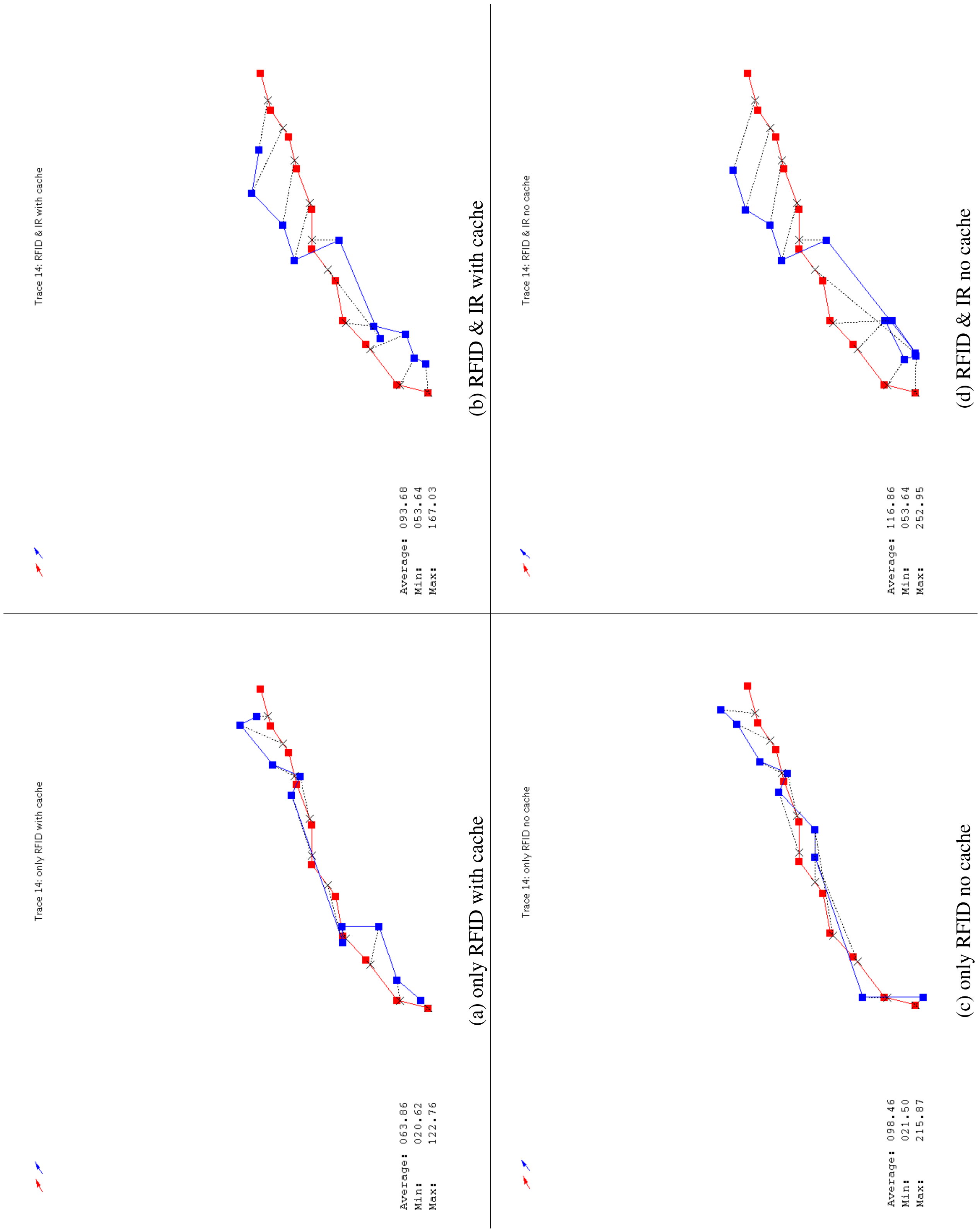
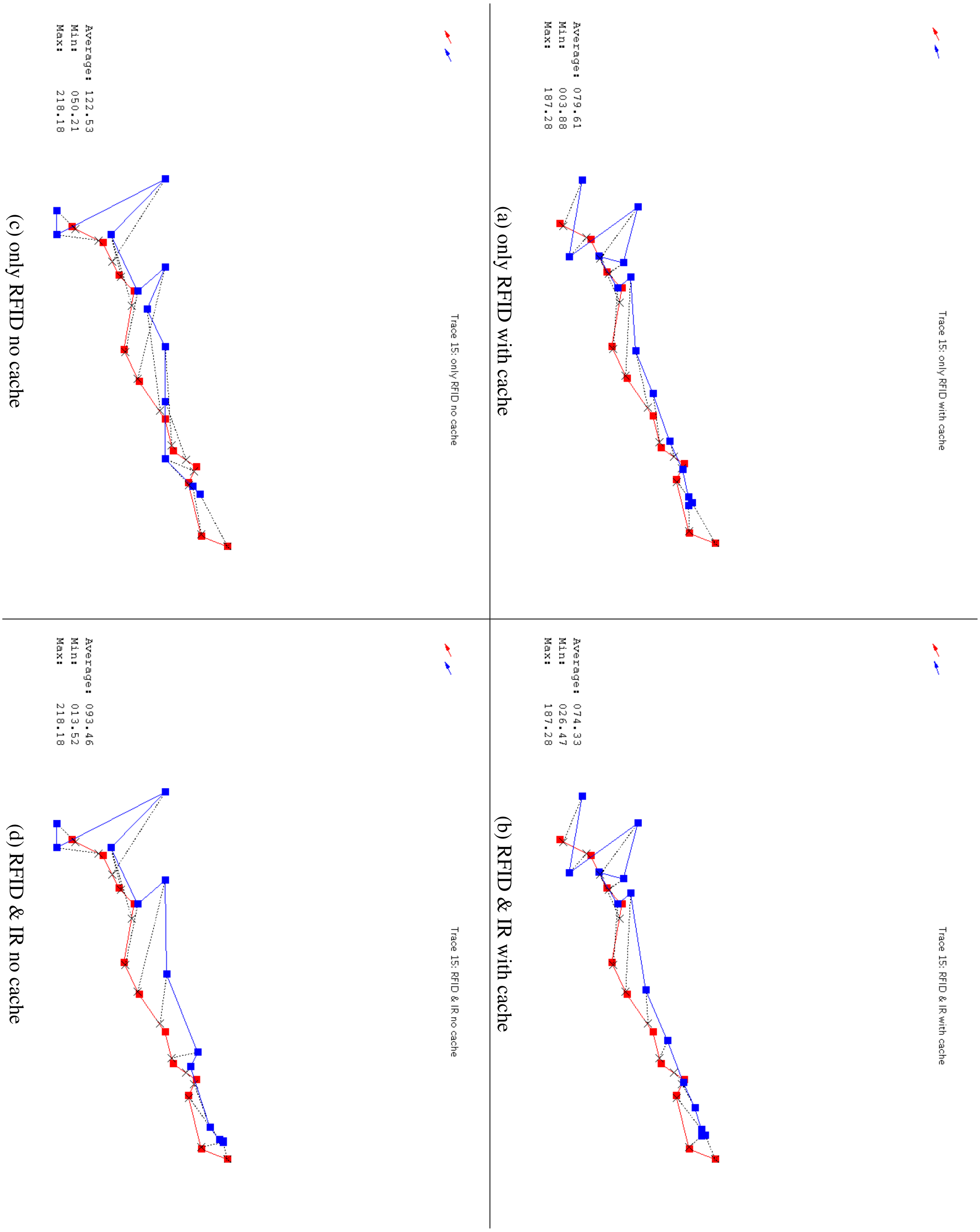


Figure A.14: Trace 14



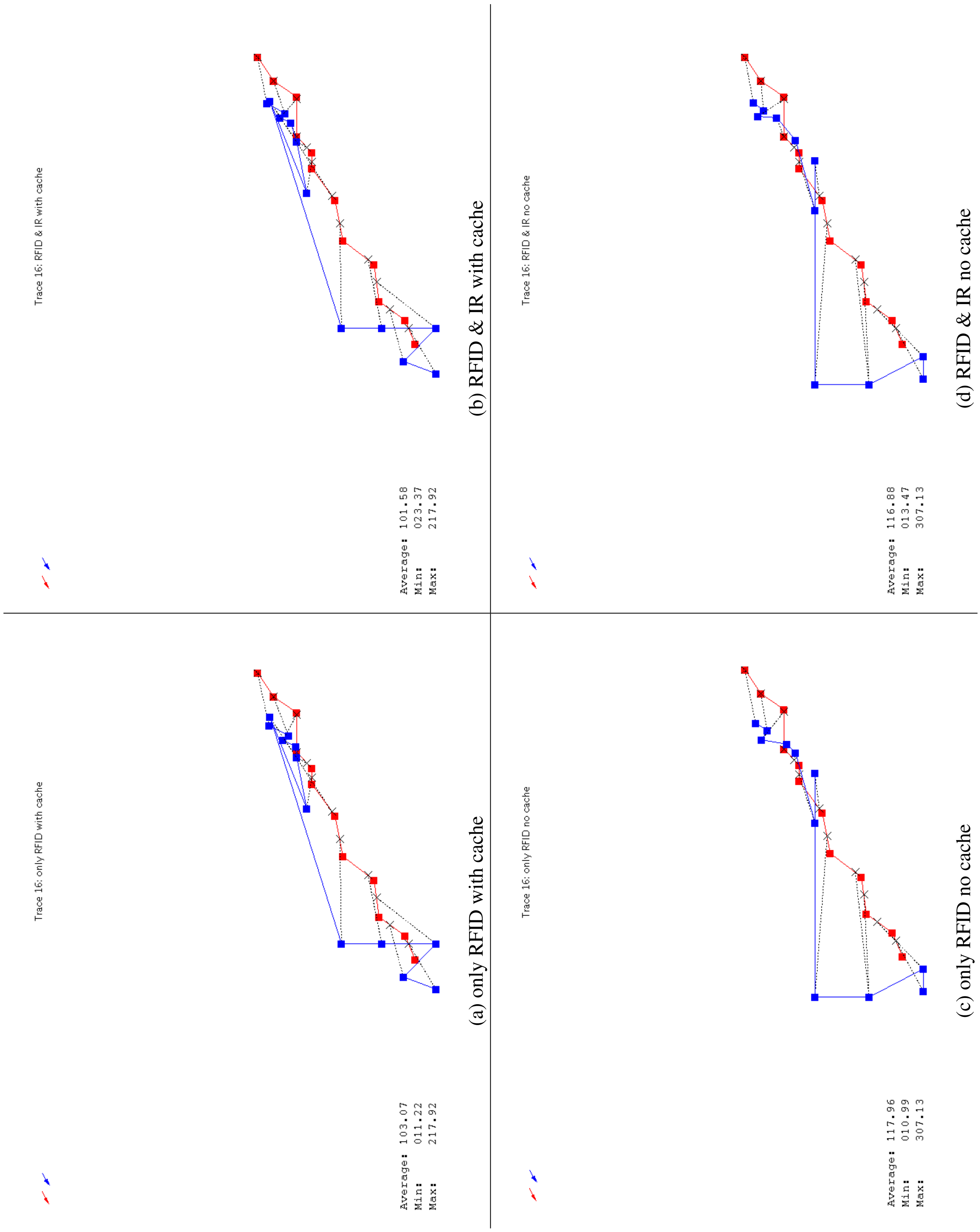
(a) only RFID with cache

(b) RFID & IR with cache

(c) only RFID no cache

(d) RFID & IR no cache

Figure A.15: Trace 15



(a) only RFID with cache

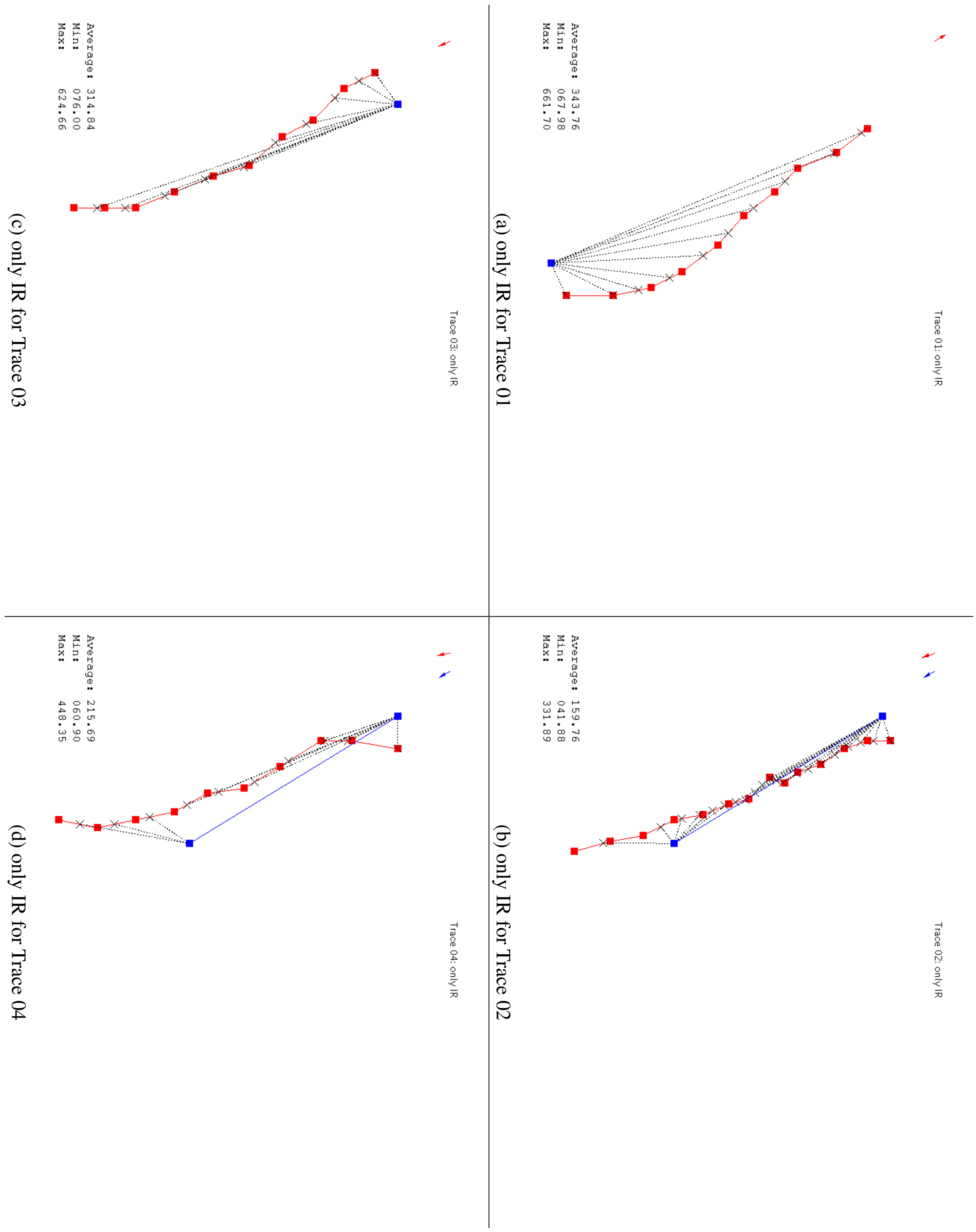
(b) RFID & IR with cache

(c) only RFID no cache

(d) RFID & IR no cache

Figure A.16: Trace 16

A.1.2 Traces for only IR Conditions



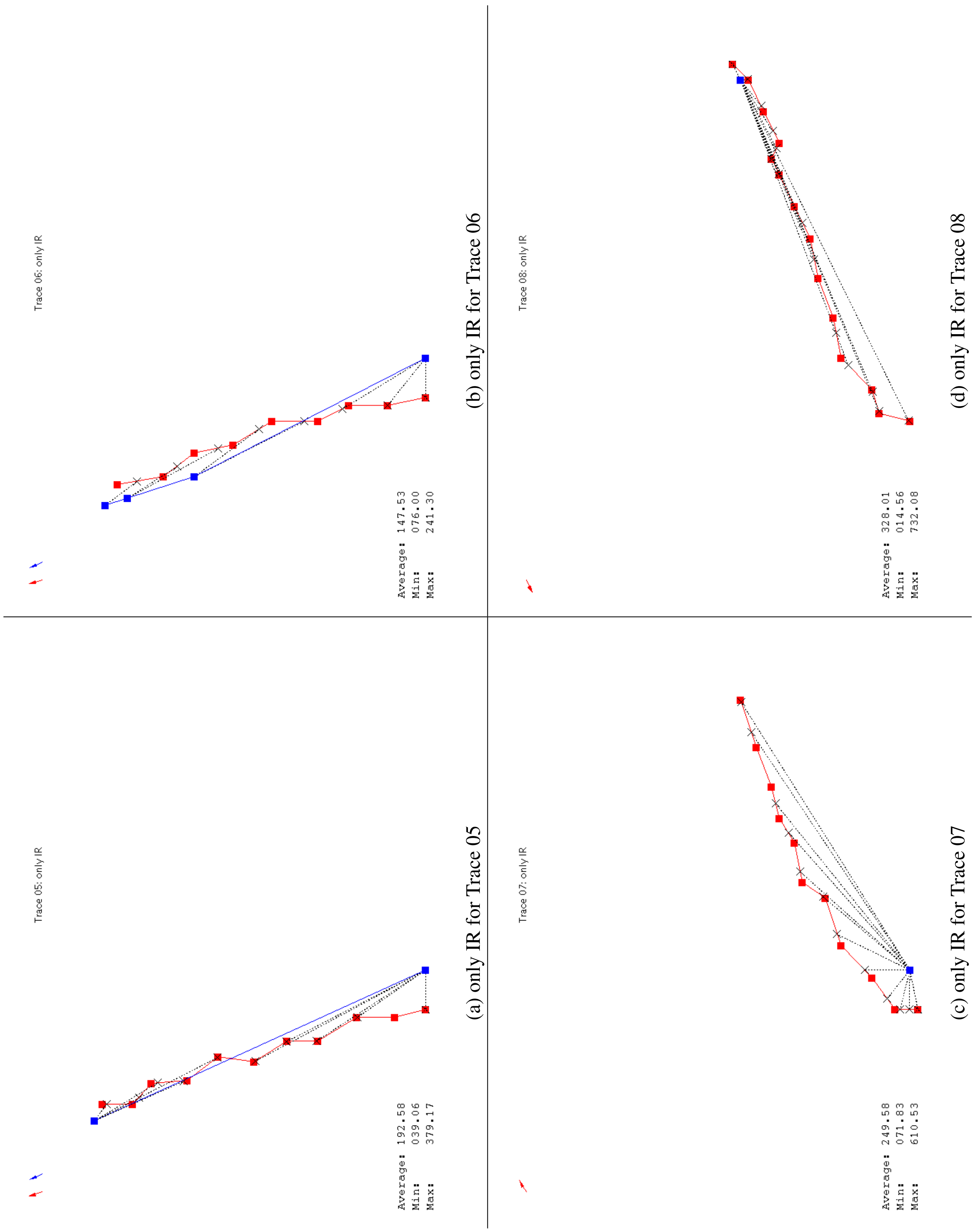
(a) only IR for Trace 01

(b) only IR for Trace 02

(c) only IR for Trace 03

(d) only IR for Trace 04

Figure A.17: IR Traces 01, 02, 03 and 04



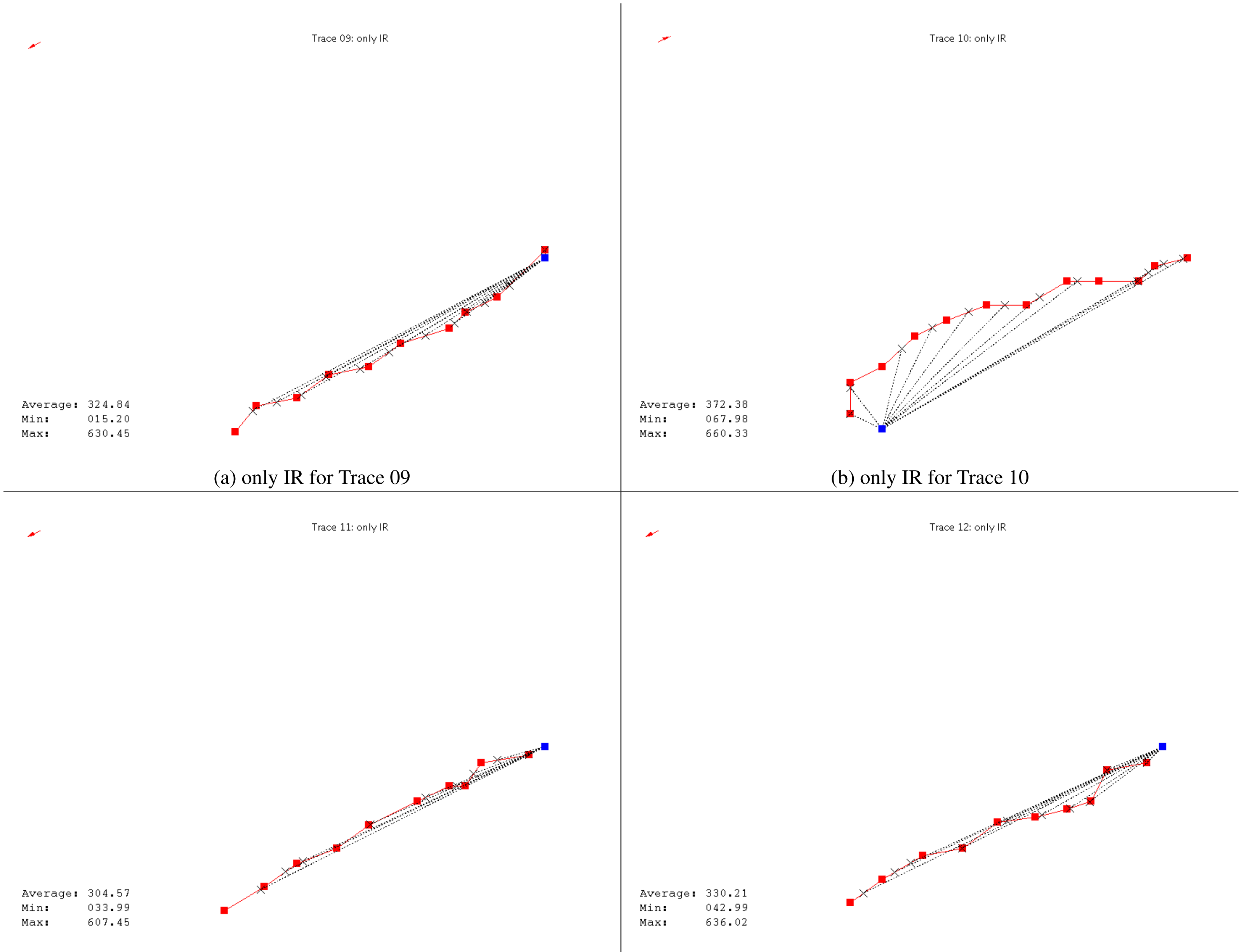
(a) only IR for Trace 05

(b) only IR for Trace 06

(c) only IR for Trace 07

(d) only IR for Trace 08

Figure A.18: IR Traces 05, 06, 07 and 08



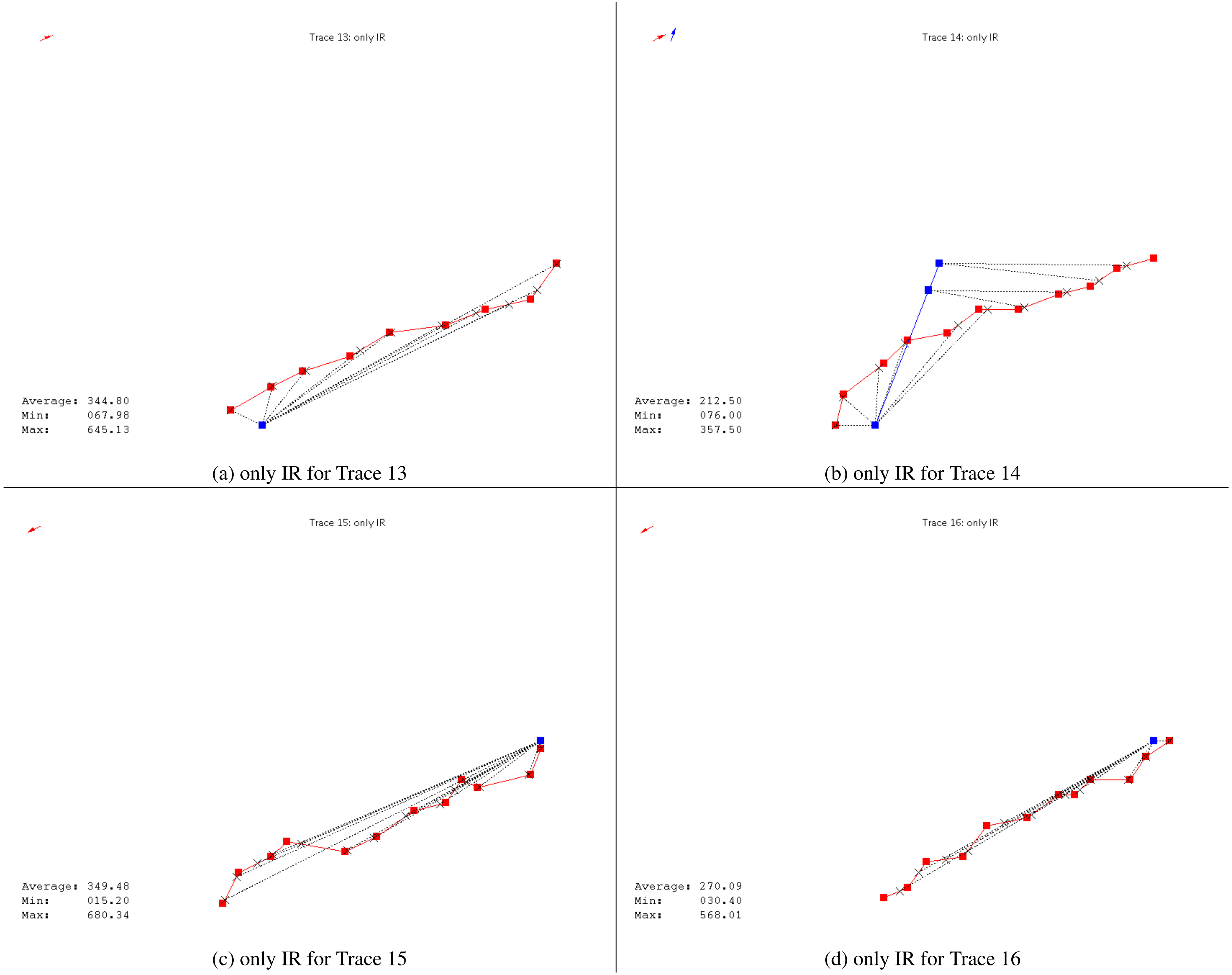
(a) only IR for Trace 09

(b) only IR for Trace 10

(c) only IR for Trace 11

(d) only IR for Trace 12

Figure A.19: IR Traces 09, 10, 11 and 12



(a) only IR for Trace 13

(b) only IR for Trace 14

(c) only IR for Trace 15

(d) only IR for Trace 16

Figure A.20: IR Traces 13, 14, 15 and 16

References

[3GPP, 1999] 3GPP (1999). 3GPP TS 05.10; Technical Specification Group GSM/EDGE Radio Access Network; Digital cellular telecommunications system (Phase 2+); Radio subsystem synchronization (Release 1999). Technical report, 3GPP.

[3GPP, 2010] 3GPP (2010). 3GPP TS 45.010; Technical Specification Group GSM/EDGE Radio Access Network; Radio subsystem synchronization (Release 10). Technical report, 3GPP.

[Addlesee et al., 2001] Addlesee, M., Curwen, R., Hodges, S., Newman, J., Steggles, P., Ward, A., and Hopper, A. (2001). Implementing a Sentinent Computing System. *IEEE Computer Magazine*, 34(8):50–56.

[Aginsky et al., 1997] Aginsky, V., Harris, C., Rensink, R., and Beusmans, J. (1997). Two Strategies for Learning a Route in a Driving Simulator. *Journal of Environmental Psychology*, 17:317–331.

[Akyildiz et al., 2002] Akyildiz, I., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). Wireless Sensor Networks: a Survey. *Computer Networks*, 38(4):393–422.

[Allen et al., 1978] Allen, G. L., Siegel, A. W., and Rosinski, R. R. (1978). The Role of Perceptual Context in Structuring Spatial Knowledge. *Journal of Experimental Psychology: Human Learning and Memory*, 4(6):617–630.

[Appleyard et al., 1988] Appleyard, S., Linford, R., and Yarwood, J. (1988). *Marine Electronic Navigation*. New York: Routledge & Kegan Paul, second enlarged edition.

[Arulampalam et al., 2002] Arulampalam, S., Maskell, S., Gordon, N., and Clapp, T. (2002). A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188.

[Ascher et al., 2011] Ascher, C., Zwirello, L., Zwick, T., and Trommer, G. (2011). Integrity Monitioring for UWB/INS Tightly Coupled Pedestrian Indoor Scenarios. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN'11)*, pages 1–6 (online publication, no pagination). IEEE.

[Aslan et al., 2006] Aslan, I., Schwalm, M., Baus, J., Krüger, A., and Schwartz, T. (2006). Acquisition of Spatial Knowledge in Location Aware Mobile Pedestrian Navigation Systems. In *Proceedings of the 8th International Conference on Human Computer Interaction with Mobile Devices and Services (Mobile HCI 2006)*, pages 105–108. ACM.

[Bahl and Padmanabhan, 2000] Bahl, P. and Padmanabhan, V. (2000). RADAR: An In-Building RF-based User Location and Tracking System. In *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2000)*, volume 2, pages 775–784. IEEE.

[Bartelmus, 2002] Bartelmus, C. (2002). Entwurf und Bau einer Kommunikationsinfrastruktur für ein ein verteiltes Präsentationssystem auf Basis von Infrarot-Datenübertragung. Diploma thesis, Saarland University.

[Baunach et al., 2007] Baunach, M., Kolla, R., and Mühlberger, C. (2007). SNoW Bat: A High Precise WSN Based Location System. Technical report, University of Würzburg.

[Baus, 2003] Baus, J. (2003). *Ressourcenadaptierende hybride Personennavigation*, volume 268 of *DISKI*. Berlin: Aka.

[Ben-Gal, 2008] Ben-Gal, I. (2008). Bayesian Networks. In Ruggeri, F., Faltin, F., and Kenett, R., editors, *Encyclopedia of Statistics in Quality and Reliability*. New Jersey: Wiley.

[Blankenbach et al., 2011] Blankenbach, J., Norrdine, A., and Hellmers, H. (2011). Adaptive Signal Processing for a Magnetic Indoor Positioning System. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN'11)*, pages 1–4 (online publication, no pagination). IEEE.

[Bluetooth SIG, 2008] Bluetooth SIG (2008). About the Bluetooth Special Interest Group. http://www.bluetooth.com/Bluetooth/SIG.

[Bluetooth SIG, 2010] Bluetooth SIG (2010). Specification of the Bluetooth System. http://developer.bluetooth.org/.

[Borio et al., 2011] Borio, D., O'Driscoll, C., and Fortuny-Guasch, J. (2011). Pulsed Pseudolite Signal Effects on Non-Participating GNSS Receivers. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN'11)*, pages 1–6 (online publication, no pagination). IEEE.

[Borkowski and Lempiäinen, 2006] Borkowski, J. and Lempiäinen, J. (2006). Practical Network-Based Techniques for Mobile Positioning in UMTS. *EURASIP Journal on Advances in Signal Processing*, 2006:1–15. Article ID 12930.

[Borkowski et al., 2004] Borkowski, J., Niemelä, J., and Lempiäinen, J. (2004). Performance of Cell ID+RTT Hybrid Positioning Method for UMTS Radio Networks. In *Proceedings of the 5th European Wireless Conference Mobile and Wireless Systems beyond 3G*, pages 232–238. VDE.

[Brandherm, 2006] Brandherm, B. (2006). *Eingebettete dynamische Bayessche Netze n-ter Ordnung*. PhD thesis, Saarland University.

[Brandherm and Schwartz, 2005] Brandherm, B. and Schwartz, T. (2005). Geo Referenced Dynamic Bayesian Networks for User Positioning on Mobile Systems. In Strang, T. and Linnhoff-Popien, C., editors, *Proceedings of the First International Workshop on Location- and Context-Awareness (LoCA)*, volume 3479 of *LNCS*, pages 223–234. Berlin, Heidelberg: Springer.

[Brunelli et al., 2007] Brunelli, R., Brutti, A., Chippendale, P., Lanz, O., Omologo, M., Svaizer, P., and Tobia, F. (2007). A Generative Approach to Audio-Visual Person Tracking. In *Multimodal Technologies for Perception of Humans: International Evaluation Workshop on Classification of Events, Activities and Relationships (CLEAR 2006)*, volume 4122 of *LNCS*, pages 55–68. Berlin, Heidelberg: Springer. revised selected papers.

[Bucher and Misra, 2002] Bucher, R. and Misra, D. (2002). A Synthesizable VHDL Model of the Exact Solution for Three-dimensional Hyperbolic Positioning System. *VLSI Design*, 15(2):507–520.

[Chawathe, 2009] Chawathe, S. S. (2009). Low-Latency Indoor Localization Using Bluetooth Beacons. In *Proceedings of the International Conference on Intelligent Transportation Systems*, pages 1–7 (online publication, no pagination). IEEE.

[Chernoff and Nielsen, 2010] Chernoff, K. and Nielsen, M. (2010). Weighting of the k-Nearest-Neighbors. In *Proceedings of the International Conference on Pattern Recognition (ICPR'10)*, pages 666–669. IEEE.

[China Satellite Navigation Office, 2011] China Satellite Navigation Office (2011). BeiDou Navigation Satellite System – Signal in Space Interface Control Document. Technical Report BeiDou-SIS-ICD-Test, China Satellite Navigation Office.

[Clauss and Ebner, 1975] Clauss, G. and Ebner, H. (1975). *Statistische Maßzahlen für bivariable Verteilungen: Grundlagen der Statistik*. Frankfurt am Main: Harri Deutsch.

[Cohen, 1988] Cohen, J. (1988). *Differences between Correlation Coefficients: Statistical Power Analysis for the Behavioral Sciences*. New Jersey: Lawrence Erlbaum, 2nd edition.

[Cole, 1995] Cole, J. O. (1995). *Pride and a Daily Marathon*. Cambridge: MIT Press, first edition.

[Cooper, 1990] Cooper, G. F. (1990). The Computational Complexity of Probabilistic Inferences using Bayesian Belief Networks. *Artificial Intelligence*, 42(2-3):393–405.

[Dempsey et al., 2011] Dempsey, S., Jennings, B., Brazil, J., and Madigan, J. (2011). Using Ambient Intelligence to Infer Availability and Meta-status in a Picocell Environment. In *Proceedings of the International Conference on Intelligent Environments (IE'11)*, pages 25–31. IEEE.

[Dempster, 2009] Dempster, A. (2009). QZSS's Indoor Messaging System – GNSS Friend or Foe? *InsideGNSS*, 4(1):37–40.

[Dettori, 2008] Dettori, E. (2008). Optical Trackingsystem with 3D Position- and Orientation-Estimation. Diploma thesis, Saarland University.

[Dey, 2001] Dey, A. K. (2001). Understanding and Using Context. *Personal and Ubiquitous Computing*, 5(1):4–7.

[Diard et al., 2003] Diard, J., Bessière, P., and Mazer, E. (2003). A Survey of Probabilistic Models, Using the Bayesian Programming Methodology as a Unifying Framework. In *Proceedings of the International Conference on Computational Intelligence, Robotics and Autonomous Systems (CIRAS'03)*, pages 1–8 (online publication, no pagination). IEEE.

[Dim et al., 2011] Dim, E., Kuflik, T., Lanir, J., Stock, O., and Corra', M. (2011). Indoor Positioning: Group Mutual Positioning. In Kahl, G., Schwartz, T., Nurmi, P., Dim, E., Brandherm, B., and Forsblom, A., editors, *Proceedings of the IUI'11 Workshop on Location Awareness for Mixed and Dual Reality*, pages 1–4 (online publication, no pagination). DFKI. available at http://www.dfki.de/LAMDa/2011/accepted/LAMDa11.pdf.

[Dimitrov, 2007] Dimitrov, V. (2007). An In-/Out-door Positioning Client for Symbian OS. Master's thesis, Saarland University.

[Dressen, 2004] Dressen, D. (2004). Considerations for RFID Technology Selection. *Atmel Applications Journal*, 3:45–47.

[E. D'Atri and et al., 2007] E. D'Atri and, C. M., Serbanati, A., and Ceipidor, U. (2007). A System to Aid Blind People in the Mobility: A Usability Test and its Results. In *Proceedings of the Second International Conference on Systems (ICONS'07)*, pages 35–35. IEEE.

[Endres et al., 2010] Endres, C., Feld, M., Schwartz, T., and Mü, C. (2010). Cinematic Analysis of Automotive Personalization. In *Proceedings of the International Workshop on Multimodal Interfaces for Automotive Applications (MIAA'10) in Conjunction with IUI'10*, pages 1–6 (online publication, no pagination). ACM.

[Endres et al., 2011] Endres, C., Schwartz, T., and Müller, C. (2011). Geremin: 2D Micogestures for Drivers Based on Electric Field Sensing. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI'11)*, pages 327–330. ACM.

[Fang, 1986] Fang, B. T. (1986). Trilateration and Extension to Global Positioning System Navigation. *Journal of Guidance, Control and Dynamics*, 9(6):715–717.

[Faria et al., 2010] Faria, J., Lopes, S., Fernandes, H., Martins, P., and Barroso, J. (2010). Electronic White Cane for Blind People Navigation Assistance. In *Proceedings of the World Automation Congress (WAC'10)*, pages 1–7. IEEE.

[Feld et al., 2010] Feld, M., Schwartz, T., and Müller, C. (2010). This Is Me: Using Ambient Voice Patterns for In-Car Positioning. In *Proceedings of the International Joint Conference on Ambient Intelligence (AmI'10)*, volume 6439 of *LNCS*, pages 290–294. Berlin, Heidelberg: Springer.

[Fickert, 2005] Fickert, O. (2005). Ein Ubiquitärer Aufgabenplaner für Benutzergruppen. Bachelor's thesis, Saarland University.

[Fickert, 2007] Fickert, O. (2007). Ein Ortsbezogener Termin- und Aufgabenplaner mit Routenwissen. Master's thesis, Saarland University.

[Fox et al., 2001] Fox, D., Thrun, S., Burgard, W., and Dellaert, F. (2001). Particle Filters for Mobile Robot Localization. In Doucet, A., de Freitas, N., and Gordon, N., editors, *Sequential Monte Carlo Methods in Practice*, pages 499–516. New York: Springer.

[Fung and Chang, 1989] Fung, R. M. and Chang, K.-C. (1989). Weighting and Integrating Evidence for Stochastic Simulation in Bayesian Networks. In *Proceedings of the Annual Conference on Uncertainty in Artificial Intelligence (UAI '89)*, pages 209–220. Amsterdam: North-Holland.

[Gallagher et al., 2011] Gallagher, T., Li, B., Dempster, A. G., and Rizos, C. (2011). Power Efficient Indoor/Outdoor Positioning Handover. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN'11)*, pages 1–4 (online publication, no pagination). IEEE.

[Ghinamo and Gangyi, 2011] Ghinamo, G. and Gangyi, T. (2011). Accuracy Enhancement in High Sensitivity GNSS with Particle Filter Approach. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN'11)*, pages 1–4 (online publication, no pagination). IEEE.

[Gholamsaghaee, 2007] Gholamsaghaee, E. (2007). Entwicklung eines mobilen Parkhaus-Navigationssystems basierend auf aktiven RFID-Tags. Master's thesis, Saarland University.

[Gibson, 1979] Gibson, J. J. (1979). *The Ecological Approach to Visual Perception*. Hillsdale: Lawrence Erlbaum.

[Gordon et al., 1993] Gordon, N., Salmond, D., and Smith, A. (1993). Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation. *IEE Proceedings F – Radar and Signal Processing*, 140(2):107–113.

[Gschwandtner and Schindhelm, 2011] Gschwandtner, F. and Schindhelm, C. K. (2011). Spontaneous Privacy-friendly Indoor Positioning Using Enhanced WLAN Beacons. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN'11)*, pages 1–4 (online publication, no pagination). IEEE.

[Gustafsson and Jonsson, 2003] Gustafsson, E. and Jonsson, A. (2003). Always Best Connected. *IEEE Wireless Communications*, 10(1):49–55.

[Gustafsson et al., 2002] Gustafsson, F., Gunnarsson, F., Bergman, N., Forssell, U., Jansson, J., Karlsson, R., and Nordlund, P.-J. (2002). Particle Filters for Positioning, Navigation and Tracking. *IEEE Transactions on Signal Processing*, 50(2):425–437.

[Haddad and Porrat, 2009] Haddad, Y. and Porrat, D. (2009). Femtocell: Opportunities and Challenges of the Home Cellular Base Station for the 3G. In *Proceedings of the International Conference on Informatics 2009*, pages 209–213. IADIS.

[Haidinger, 1844] Haidinger, W. (1844). Über das directe Erkennen des polarisirten Lichts und der Lage der Polarisationsebene. *Annalen der Physik*, 139(9):29–39.

[Hare et al., 2008] Hare, J., Lewis, P., Gordon, L., and Hart, G. (2008). MapSnapper: Engineering an Efficient Algorithm for Matching Images of Maps from Mobile Phones. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 6820, pages 21–31. SPIE.

[Harter et al., 1999] Harter, A., Hopper, A., Steggles, P., Ward, A., and Webster, P. (1999). The Anatomy of a Context-Aware Application. In *Proceedings of the Annual International Conference on Mobile Computing and Networking (MOBICOM '99)*, pages 59–68. IEEE.

[Hauschildt and Kirchhof, 2010] Hauschildt, D. and Kirchhof, N. (2010). Advances in Thermal Infrared Localization: Challenges and Solutions. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN'10)*, pages 1–8 (online publication, no pagination). IEEE.

[Hecht, 1994] Hecht, D. (1994). Embedded Data Glyph Technology for Hardcopy Digital Documents. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 2171, pages 341–352. SPIE.

[Heckmann, 2006] Heckmann, D. (2006). *Ubiquitous User Modeling*, volume 297 of *DISKI*. Berlin: Aka.

[Heckmann et al., 2005a] Heckmann, D., Brandherm, B., Schmitz, M., Schwartz, T., and von Wilamowitz-Moellendorf, M. (2005a). GUMO - the General User Model Ontology. In *Proceedings of the International Conference on User Modeling*, volume 3538 of *LNAI*, pages 428–432. Berlin, Heidelberg: Springer.

[Heckmann et al., 2005b] Heckmann, D., Schwartz, T., Brandherm, B., and Kröner, A. (2005b). Decentralized User Modeling with UserML and GUMO. In Dolog, P. and Vassileva, J., editors, *Decentralized, Agent Based and Social Approaches to User Modeling, Workshop DASUM-05 at 9th International Conference on User Modelling, UM2005*, pages 61–66, Edinburgh, Scotland. Online. http://people.cs.aau.dk/dolog/dasum/DASUM-proceedings.pdf.

[Henrion, 1986] Henrion, M. (1986). Propagating Uncertainty in Bayesian Networks by Probabilistic Logic Sampling. In *Proceedings of the Annual Conference on Uncertainty in Artificial Intelligence (UAI-86)*, pages 149–163. Elsevier Science.

[Henze and Boll, 2008] Henze, N. and Boll, S. (2008). Snap and Share Your Photobooks. In *Proceedings of the International Conference on Multimedia (ACMMM'08)*, pages 409–418. ACM.

[Herranz et al., 2011] Herranz, F., Muthukrishnan, K., and Langendoen, K. (2011). Camera Pose Estimation using Particle Filters. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN'11)*, pages 1–4 (online publication, no pagination). IEEE.

[Hornyak, 2008] Hornyak, T. (2008). RFID Powder. *Scientific American*, 298(2):68–71.

[IEEE, 1999] IEEE (1999). IEEE Standard for Information Technology – Telecommunications and Information Exchange Between Systems– Local and Metropolitan Area Networks – Specific Requirements; Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.

[Itoh et al., 2003] Itoh, H., Nakada, T., Hiratsuka, S., Nakamura, Y., Nishimura, T., Kunifuji, S., Nakashima, H., and Iwasaki, N. (2003). Indoor Location and Communication System for the Last-Ten-Meter Information Services. In *Proceedings of the Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM'03)*, pages 482–485. IEEE.

[Ji, 2011] Ji, H. (2011). Simulation and Evaluation of UbiSpot. Bachelor's thesis, Saarland University.

[Johanson and Fox, 2002] Johanson, B. and Fox, A. (2002). The Event Heap: A Coordination Infrastructure for Interactive Workspaces. In *Proceedings of the Workshop on Mobile Computing Systems and Applications*, pages 83–93. IEEE.

[Johanson et al., 2002] Johanson, B., Fox, A., and Winograd, T. (2002). The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms. *Pervasive Computing*, 1(2):67–74.

[Juels, 2006] Juels, A. (2006). RFID Security and Privacy: A Research Survey. *IEEE Journal on Selected Areas in Communications*, 24(2):381–394.

[Jung, 2009] Jung, R. (2009). *Ambiente Audionotifikation - Ein System zur kontextsensitiven Integration von nicht-intrusiven Notifikationssignalen in emotionsklassifizierten Soundscapes*. PhD thesis, Saarland University.

[Jung and Schwartz, 2007a] Jung, R. and Schwartz, T. (2007a). A Location-Adaptive Human-Centered Audio Email Notification Service for Multi-User Environments. In Jacko, J. A., editor, *Human-Computer Interaction*, volume 4552 of *LNCS*, pages 340–348. Berlin, Heidelberg: Springer.

[Jung and Schwartz, 2007b] Jung, R. and Schwartz, T. (2007b). Peripheral Notification With Customized Embedded Audio Cues. In Martens, W. L., editor, *Proceedings of the International Conference on Auditory Display (ICAD'07)*, pages 221–228. Schulich School of Music McGill University.

[Kahl, 2007] Kahl, G. (2007). PEG and MaMiNa: Erkennung von Zeigegesten in einer instrumentierten Umgebung und auf Landmarken basierte Wegsuche zur Navigation. Master's thesis, Saarland University.

[Kahl et al., 2011] Kahl, G., Spassova, L., Schöning, J., Gehring, S., and Krüger, A. (2011). IRL SmartCart – A User-adaptive Context-aware Interface for Shopping

Assistance. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI'11)*, pages 359–362. ACM.

[Kahl et al., 2008] Kahl, G., Wasinger, R., Schwartz, T., and Spassova, L. (2008). Three Output Planning Strategies for Use in Context-Aware Computing Scenarios. In *Proceedings of the AISB 2008 Symposium on Multimodal Output Generation (MOG 2008)*, pages 46–49. University of Aberdeen.

[Kálmán, 1960] Kálmán, R. E. (1960). A New Approach to Linear Filtering and Predicition Problems. *Transactions of the ASME – Journal of Basic Engineering*, 82(Series D):35–45.

[Kato and Billinghurst, 1999] Kato, H. and Billinghurst, M. (1999). Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System. In *Proceedings of the International Workshop on Augmented Reality '99*, pages 85–94. IEEE Press.

[Kiers et al., 2011] Kiers, M., Krajnc, E., Dornhofer, M., and Bischof, W. (2011). Evaluation and Improvements of an RFID Based Indoor Navigation System for Visually Impaired and Bild People. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN'11)*, pages 1–4 (online publication, no pagination). IEEE.

[Kirchhof, 2011] Kirchhof, N. (2011). Approximation of Spatial and Temporal Temperature Behavior for Indoor Devices based on Thermal Recordings. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN'11)*, pages 1–4 (online publication, no pagination). IEEE.

[Klebesadel et al., 1982] Klebesadel, R. W., Evans, W. D., Fenimore, E. E., Laros, J. G., and Terrell, J. (1982). Cosmic Gamma Ray Bursts – A Continuing Mystery. *Los Alamos Science*, 3(2):5–23.

[Kohtake et al., 2011] Kohtake, N., Morimoto, S., Kogure, S., and Manandhar, D. (2011). Indoor and Outdoor Seamless Positioning using Indoor Messaging System and GPS. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN'11)*, pages 1–4 (online publication, no pagination). IEEE.

[Kolb and Whishaw, 2003] Kolb, B. and Whishaw, I. Q. (2003). *Fundamentals of Human Neuropsychology*. New York: W.H. Freeman, 5th revised edition.

[Köppe et al., 2011] Köppe, E., Bartolmai, M., Liers, A., and Schiller, J. (2011). Tracking Persons using a Radio-Based Multi-Sensor System. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN'11)*, pages 1–4 (online publication, no pagination). IEEE.

[Kray, 2003] Kray, C. (2003). *Situated Interaction on Spatial Topics*, volume 274 of *DISKI*. Berlin: Aka.

[Krüger et al., 2004] Krüger, A., Aslan, I., and Zimmer, H. (2004). The Effects of Mobile Pedestrian Navigation Systems on the Concurrent Aquisition of Route and Survey Knowledge. In *Mobile Human-Computer Interaction – MobileHCI 2004*, volume 3160 of *LNCS*, pages 446–450. Berlin, Heidelberg: Springer.

[Kuflik et al., 2011a] Kuflik, T., Lanir, J., Dim, E., Wecker, A., Corra', M., Zancanaro, M., and Stock, O. (2011a). Indoor Positioning: Challenges and Solutions for Indoor Cultural Heritage Site. In *Proceedings of the International Conference on Intelligent User Interfaces (IUI'11)*, pages 375–378. ACM.

[Kuflik et al., 2011b] Kuflik, T., Stock, O., Zancanaro, M., Gorfinkel, A., Jbara, S., Kats, S., and Kashtan, J. S. N. (2011b). A Visitor's Guide in an Active Museum: Presentations, Communications, and Reflection. *ACM Journal on Computing and Cultural Heritage*, 3(3):11:1–11:25.

[Landt, 2005] Landt, J. (2005). The History of RFID. *IEEE Potentials*, 24(4):8–1.

[Laoudias et al., 2011] Laoudias, C., Michaelides, M., and Panayiotou, C. (2011). Fault Detection and Mitigation in WLAN RSS Fingerprint-Based Positioning. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN'11)*, pages 1–7 (online publication, no pagination). IEEE.

[Lauritzen and Spiegelhalter, 1988] Lauritzen, S. and Spiegelhalter, D. (1988). Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems. *Journal of the Royal Statistical Society*, B 50(2):157–224.

[Ledlie et al., 2011] Ledlie, J., geun Park, J., Curtis, D., Calvacante, A., Camara, L., and Vieira, R. (2011). Molé: A Scalable, User-Generated WiFi Positioning Engine. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN'11)*, pages 1–10 (online publication, no pagination). IEEE.

[Link et al., 2011] Link, J. A. B., Smith, P., and Wehrle, K. (2011). FootPath: Accurate Map-based Indoor Navigation Using Smartphones. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN'11)*, pages 1–8 (online publication, no pagination). IEEE.

[Löchtefeld et al., 2009] Löchtefeld, M., Schöning, J., Rohs, M., and Krüger, A. (2009). Marauders Light : Replacing the Wand with a Mobile Camera Projector Unit. In *Proceedings of the International Conference on Mobile and Ubiquitous Multimedia (MUM'09)*, pages 1–4 (online publication, no pagination). ACM.

[Loskyll et al., 2009] Loskyll, M., Heckmann, D., and Glahn, C. (2009). Visualization of Spatial Knowledge with Ontology Trees and Adaptable Search Result Grids in the Era of Web 3.0. In Tochtermann, K. and Maurer, H., editors, *Proceedings of the International Conference on Knowledge Management and Knowledge Technologies (I-Know'09)*, pages 385–390. Verlag der Technischen Universität Graz.

[Lynch, 1960] Lynch, K. (1960). *The Image of the City*. Cambridge: MIT Press.

[Martínez et al., 2011] Martínez, M., Villanueva, F., Santofímia, M., and López, J. (2011). A Multimodal Distributed Architecture for Indoor Localization. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN'11)*, pages 1–4 (online publication, no pagination). IEEE.

[Masters and Michael, 2005] Masters, A. and Michael, K. (2005). Humancentric Applications of RFID Implants: The Usability Contexts of Control, Convencience and Care. In *Proceedings of the International Workshop on Mobile Commerce and Services (WMCS'05)*. IEEE.

[Maybeck, 1979] Maybeck, P. S. (1979). *Stochastic Models, Estimation and Control*, volume 1, chapter Chapter 1. New York, San Francisco, London: Academic Press.

[Moghtadaiee et al., 2011] Moghtadaiee, V., Dempster, A. G., and Lim, S. (2011). Indoor Localization Using FM Radio Signals: A Fingerprinting Approach. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN'11)*, pages 1–7 (online publication, no pagination). IEEE.

[Münzer and Stahl, 2008] Münzer, S. and Stahl, C. (2008). Learning of Visual Route Instructions for Indoor Wayfinding. In Hölscher, C., editor, *Spatial Cognition 2008 Poster Proceedings. International Conference on Spatial Cognition*, SFB/TR8 Report No. 016-08/2008, pages 1–4 (online publication, no pagination). University of Bremen / University of Freiburg.

[Murphy and Hereman, 1995] Murphy, W. S. and Hereman, W. (1995). Determination of a Position in Three Dimensions using Trilateration and Approximate Distances. Tech. report mcs-95-07, Mathematical and Computer Sciences Department, Colorado School of Mines. 21 pages.

[Mutafchiev, 2008] Mutafchiev, S. (2008). Hybride Navigationsvisualisierung auf mobilen Geräten. Diploma thesis, Saarland University.

[Mušicki and Koch, 2008] Mušicki, D. and Koch, W. (2008). Geolocation using TDOA and FDOA Measurements. In *Proceedings of the International Conference on Information Fusion*, pages 1987–1994. IEEE.

[Najib et al., 2011] Najib, W., Klepal, M., and Wibowo, S. B. (2011). MapUme: Scalable Middleware for Location Aware Computing Applications. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN'11)*, pages 1–6 (online publication, no pagination). IEEE.

[Nakamura et al., 2003] Nakamura, Y., Nishimura, T., Itoh, H., and Nakashima, H. (2003). ID-CoBIT: A Battery-Less Information Terminal with Data Upload Capacity. In *Conference of the IEEE Industrial Electronics Societey (IECON'03*, volume 3, pages 2511–2516. IEEE.

[Nakashima, 2007] Nakashima, H. (2007). Cyber Assist Project for Ambient Intelligence. In Augusto, J. C. and Shapiro, D., editors, *Advances in Ambient Intelligence*, volume 164 of *Frontiers in Artificial Intelligence and Applications*. Amsterdam: IOS.

[Nakashima and Hasida, 2010] Nakashima, H. and Hasida, K. (2010). Cyber Assist Project as Service Science. *Synthesiology – English Edition*, 3(2):107–123.

[National Communications System Technology & Standards Division, 1996] National Communications System Technology & Standards Division (1996). Federal Standard 1037C: Telecommunications: Glossary of Telecommunication Terms. Technical report, General Services Administration Information Technology Service, Arlington, USA.

[National Imagery and Mapping Agency, 2000] National Imagery and Mapping Agency (2000). Department of Defense World Geodetic System 1984. Technical Report Third Edition, National Imagery and Mapping Agency.

[NavCen, 1996] NavCen (1996). *NAVSTAR GPS User Equipment Introduction*. US Department of Homeland Security, public release edition. http://www.navcen.uscg.gov/pubs/gps/gpsuser/gpsuser.pdf.

[NavCen, 2001] NavCen (2001). *2001 Federal Radionavigation Systems*. US Department of Defence and US Department of Transportation. http://www.navcen.uscg.gov/pdf/frp/frp2001/FRS2001.pdf.

[Ndili, 1994] Ndili, A. (1994). GPS Pseudolite Signal Design. In *Proceedings of the ION Global Positioning System GPS'94 Conference*, pages 1375–1382. ION (Institute of Navigation).

[Nekoogar, 2011] Nekoogar, F. (2011). *Ultra-Wideband Communications: Fundamentals and Applications*. New Jersey: Pearson.

[Ni et al., 2004] Ni, L., Liu, Y., Lau, Y., and Patil, A. (2004). LANDMARC: Indoor Location Sensing using Active RFID. *Wireless Networks*, 10(6):701–710.

[Omologo et al., 1998] Omologo, M., Svaizer, P., and Mori, R. D. (1998). Acoustic Transduction. In Mori, R. D., editor, *Spoken Dialogues with Computers*, pages 24–68. London: Academic Press.

[Ordnance Survey, 2010] Ordnance Survey (2010). A Guide to Coordinate Systems in Great Britain. Technical Report D00659 v2.1, Southampton: Ordnance Survey.

[Passas et al., 2006] Passas, N., Paskalis, S., Kaloxylos, A., Bader, F., Narcisi, R., Tsontsis, E., Jahan, A. S., Aghvami, H., O'Droma, M., and Ganchev, I. (2006). Enabling Technologies for the 'Always Best Connected Concept'. *Wireless Communications and Mobile Computing*, 6(4):523–540.

[Patton et al., 2005] Patton, B. K., Aukerman, R., and Shorter, J. D. (2005). Wireless Technologies, Wireless Fidelity (Wi-Fi) & Worldwide Interoperability for Microwave Access (WiMax). *Issues in Information Systems*, VI(2):364–370.

[Pearl, 1985] Pearl, J. (1985). Bayesian Networks: A Model of Self-Activated Memory for Evidential Reasoning. In *Proceedings of the 7th Conference of the Cognitive Science Society*, pages 329–334. University of California.

[Pearl, 1986] Pearl, J. (1986). Fusion, Propagation and Structuring in Belief Networks. *Artificial Intelligence*, 29(3):241–288.

[Pearl, 1987] Pearl, J. (1987). Evidential Reasoning Using Stochastic Simulation of Causal Models. *Artificial Intelligence*, 32(2):245–257.

[Peng et al., 2011] Peng, J., Zhu, M., and Zhang, K. (2011). New Algorithms Based on Sigma Point Kalman Filter Technique for Multi-Sensor Integrated RFID Indoor/Outdoor Positioning. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN'11)*, pages 1–4 (online publication, no pagination). IEEE.

[Pereira et al., 2011a] Pereira, C., Guenda, L., and Carvalho, N. B. (2011a). A Smart-Phone Indoor/Outdoor Localization System. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN'11)*, pages 1–4 (online publication, no pagination). IEEE.

[Pereira et al., 2011b] Pereira, F., Theis, C., Moreira, A., and Ricardo, M. (2011b). Evaluating Location Fingerprinting Methods for Underground GSM Networks Deployed over Leaky Feeder. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN'11)*, pages 1–4 (online publication, no pagination). IEEE.

[Piekarski et al., 2003] Piekarski, W., Avery, B., Thomas, B., and Malbezin, P. (2003). Hybrid Indoor and Outdoor Tracking for Mobile 3D Mixed Reality. In

International Symposium on Mixed and Augmented Reality (ISMAR'03), pages 266–267. IEEE.

[Poole et al., 1998] Poole, D., Mackworth, A., and Goebel, R. (1998). *Computational Intelligence, A Logical Approach*. New York: Oxford University.

[Pottie and Kaiser, 2000] Pottie, G. and Kaiser, W. (2000). Wireless Integrated Network Sensors. *Communications of the ACM*, 43(5):51–58.

[QZSS, 2009] QZSS (2009). *Interface Specification for QZSS (IS-QZSS Ver. 1.1)*. Japan Aerospace Exploration Agency.

[Rabinowitz and Spilker, 2005] Rabinowitz, M. and Spilker, J. J. (2005). A new Positioning System using Television Synchronization Signals. *IEEE Transactions on Broadcasting*, 51(1):51–61.

[Ranta-aho, 2010] Ranta-aho, K. (2010). Performance of 3GPP Rel-9 LTE Positioning Methods. http://www.cwins.wpi.edu/workshop10/pres/std_2.pdf. Presentation Slides of the Invitational Workshop on Opportunistic RF Localization for Next Generation Wireless Devices.

[Röfer et al., 2009a] Röfer, T., Laue, T., and Gersdorf, B. (2009a). iWalker – An Intelligent Walker providing Services for the Elderly. In *Proceedings of the European Conference on Technically Assisted Rehabilitation (TAR'09)*, pages 1–4 (online publication, no pagination). VDe/VDI.

[Röfer et al., 2009b] Röfer, T., Mandel, C., and Laue, T. (2009b). Controlling an Automated Wheelchair via Joystick/Head-Joystick Supported by Smart Driving Assistance. In *Proceedings of the International Conference on Rehabilitation Robotics (ICORR'09)*, pages 743–748. IEEE.

[Ropars et al., 2011] Ropars, G., Gorre, G., Floch, A. L., Enoch, J., and Lakshminarayanan, V. (2011). A Depolarizer as a Possible Precise Sunstone for Viking Navigation by Polarized Skylight. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*. published online before print.

[Rosa et al., 2011] Rosa, F. D., Xu, L., Leppäkoski, H., and Nurmi, J. (2011). Relative Positioning of Mass Market Devices in Ad-hoc Networks. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN'11)*, pages 1–7 (online publication, no pagination). IEEE.

[Roth, 2005] Roth, J. (2005). *Mobile Computing: Grundlagen, Technik, Konzepte*. Heidelberg: dpunkt, second edition.

[Runge et al., 2011] Runge, A., Baunach, M., and Kolla, R. (2011). Precise Self-Calibration of Ultrasound Based Indoor Localization Systems. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN'11)*, pages 1–8 (online publication, no pagination). IEEE.

[Ruotsalainen et al., 2011] Ruotsalainen, L., Kuusniemi, H., and Chen, R. (2011). Heading Change Detection for Indoor Navigation with a Smartphone Camera. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN'11)*, pages 1–7 (online publication, no pagination). IEEE.

[Russel and Norvig, 1995] Russel, S. J. and Norvig, P. (1995). *Artificial Intelligence, A Modern Approach*. New Jersey: Prentice Hall International.

[Russell and Norvig, 2003] Russell, S. J. and Norvig, P. (2003). *Artificial Intelligence, A Modern Approach*, chapter 14 and 15, pages 492–580. New Jersey: Pearson Education, second edition.

[Sadalla et al., 1980] Sadalla, E. K., Burroughs, W. J., and Staplin, L. J. (1980). Reference Points in Spatial Cognition. *Journal of Experimental Psychology: Human Learning and Memory*, 6(5):516–528.

[Sakamoto et al., 2011] Sakamoto, Y., Arie, H., Ebinuma, T., Fujii, K., and Sugano, S. (2011). High-accuracy IMES Localization Using a Movable Reveiver Antenna. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN'11)*, pages 1–6 (online publication, no pagination). IEEE.

[Saliba, 2011] Saliba, B. (2011). An Evaluation Method For Indoor Positioning Systems On the Example of LORIOT. Master's thesis, Saarland University.

[Scarth, 1999] Scarth, R. N. (1999). *Echoes from the Sky: A Story of Acoustic Defence*. Hythe: Civic Society.

[Schilit et al., 1994] Schilit, B. N., Adams, N., and Want, R. (1994). Context-Aware Computing Applications. In *First International Workshop on Mobile Computing Systems and Applications*, pages 85–90. IEEE.

[Schmitz, 2010] Schmitz, M. (2010). *Tangible Interaction with Anthropomorphic Smart Objects in Instrumented Environments*. PhD thesis, Saarland University.

[Schmitz et al., 2008] Schmitz, M., Endres, C., and Butz, A. (2008). A Survey of Human-Computer Interaction Design in Science Fiction Movies. In *Proceedings of the International Conference on Intelligent Technologies for Interactive Entertainment (ICST INTETAIN' 08)*, pages 1–10 (online publication, no pagination). ICST.

[Schöttle, 2006] Schöttle, F. (2006). IPlay Blue: Interaktive Displays mit Bluetooth-basierter Benutzererkennung. Fortgeschrittenen praktikum, Saarland University.

[Schwartz, 2010] Schwartz, T. (2010). Shopping Assistance. In May, M. and Saitta, L., editors, *Ubiquitous Knowledge Discovery*, volume 6202 of *LNAI*, chapter A Blueprint for Ubiquitous Knowledge Discovery – Application Challenges for Ubiquitous Knowledge Discovery, pages 120–122. Springer, Berlin, Heidelberg, New York.

[Schwartz et al., 2005] Schwartz, T., Brandherm, B., and Heckmann, D. (2005). Calculation of the User-Direction in an Always Best Positioned Mobile Localization System. In *Proceedings of the International Workshop on Artificial Intelligence in Mobile Systems (AIMS 2005)*, pages 1–4 (online publication, no pagination), Salzburg, Austria. AIMS.

[Schwartz et al., 2006] Schwartz, T., Heckmann, D., and Baus, J. (2006). Sharing Sensor Data in Instrumented Environments. In *Proceedings of the Workshop on Ubiquitous Knowledge Discovery for Users (UKDU'06)*, pages 81–88. Humboldt University Berlin.

[Schwartz and Jung, 2006] Schwartz, T. and Jung, R. (2006). A Development Cycle for Ambient System Design. In Pederson, T., Pinto, H., Schmitz, M., Stahl, C., and Terrenghi, L., editors, *Proceedings of Modelling and Designing User Assistance in Intelligent Environments (MODIE 2006)*, pages 13–17. Saarland University.

[Schwartz et al., 2010a] Schwartz, T., Stahl, C., Baus, J., and Wahlster, W. (2010a). Seamless Resource-Adaptive Navigation. In Crocker, M. W. and Siekmann, J., editors, *Resource-Adaptive Cognitive Processes*, Cognitive Technologies, pages 239 – 265. Berlin, Heidelberg, New York: Springer.

[Schwartz et al., 2010b] Schwartz, T., Stahl, C., Müller, C., Dimitrov, V., and Ji, H. (2010b). UbiSpot – A User-Trained Always Best Positioned Engine for Mobile Phones. In *Proceedings of Ubiquitous Positioning Indoor Navigation and Location Based Services (UPINLBS'10)*, pages 1–8 (online publication, no pagination). IEEE.

[Scientific American, 1880] Scientific American (1880). Navigation in Fogs. *Scientific American*, XLIII(1):8.

[Silventoinen and Rantalainen, 1996] Silventoinen, M. and Rantalainen, T. (1996). Mobile Station Locating in GSM. In *Proceedings of the International Conference on Personal Wireless Communications (ICPWC'96)*, pages 53–59. IEEE.

[Silventoinen et al., 1996] Silventoinen, M. I., Kuusela, M., and Ranta, P. A. (1996). Analysis of a new Channel Access Method for Home Base Station. In *Proceedings* *of the 5th International Universal Personal Communications*, volume 2, pages 930–935. IEEE.

[Singh and Ismail, 2005] Singh, K. and Ismail, M. (2005). OTDOA Location Determining Technology for Universal Intelligent Positioning System (UIPS) Implementation in Malaysia. In *Proceedings of the International Conference on Communication*, pages 1057–1061. IEEE.

[Spassova, 2011] Spassova, L. (2011). *Interactive Ubiquitous Displays Based on Steerable Projection*. PhD thesis, Saarland University.

[Spassova et al., 2009] Spassova, L., Schöning, J., Kahl, G., and Krüger, A. (2009). Innovative Retail Laboratory. In *Adjunct Proceedings of the European Conference on Ambient Intelligence (AmI'09)*, pages 1–4 (online publication, no pagination). AmI.

[Stahl, 2009] Stahl, C. (2009). *Spatial Modeling of Activity and User Assistance in Instrumented Environments*. PhD thesis, Saarland University.

[Stahl et al., 2005] Stahl, C., Brandherm, B., Schmitz, M., and Schwartz, T. (2005). Navigational and Shopping Assistance on the Basis of User Interactions in Intelligent Environments. In *Proceedings of the International Workshop on Intelligent Environments*, pages 182–191. IEEE.

[Stahl and Haupert, 2006] Stahl, C. and Haupert, J. (2006). Taking Location Modelling to new Levels: A Map Modelling Toolkit for Intelligent Envirionments. In Hazas, M., Krumm, J., and Strang, T., editors, *Proceedings of the International Workshop on Location- and Context-Awareness (LoCA'06)*, volume 3987 of *LNCS*, pages 74–85. Berlin, Heidelberg, New York: Springer.

[Stahl et al., 2007] Stahl, C., Heckmann, D., Schwartz, T., and Fickert, O. (2007). Here and Now: A User-Adaptive and Location-Aware Task Planner. In *Proceedings of the International Workshop on Ubiquitous and Decentralized User Modeling (UbiDeUM'2007)*, pages 52–63. UBIDEUM.

[Stahl and Schwartz, 2010] Stahl, C. and Schwartz, T. (2010). Modeling and Simulating Assisitive Environments in 3-D with the YAMAMOTO Toolkit. In Mautz, R., Kunz, M., and Ingensand, H., editors, *Proceedings of the 2010 International Conference on Indoor Positioning and Indoor Navigation (IPIN'10)*, pages 1–6 (online publication, no pagination). IEEE.

[Steggles and Gschwind, 2005] Steggles, P. and Gschwind, S. (2005). The Ubisense Smart Space Platform. In *Adjunct Proceedings of the International Conference on Pervasive Computing (Percom'05)*, pages 73–76. Austrian Computer Society.

[Stephan et al., 2009] Stephan, P., Heck, I., Kraus, P., and Frey, G. (2009). Evaluation of Indoor Positioning Technologies under industrial application conditions in the SmartFactoryKL based on EN ISO 9283. In *Proceedings of the IFAC Symposium on Information Control Problems in Manufacturing (INCOM'09)*, pages 870–875. IFAC.

[Stockman, 1948] Stockman, H. (1948). Communication by Means of Reflected Power. *Proceedings of the IRE*, 36(10):1196–1204.

[Storms et al., 2010] Storms, W., Shockley, J., and Raquet, J. (2010). Magnetic Field Navigation in an Indoor Environment. In *Proceedings of Ubiquitous Positioning Indoor Navigation and Location Based Services (UPINLBS'10)*, pages 1–10 (online publication, no pagination). IEEE.

[Takaragi et al., 2002] Takaragi, K., Usami, M., Imura, R., Itsuki, R., and Satoh, T. (2002). An Ultra Small Individual Recognition Security Chip. *IEEE Micro*, 21(6):4349.

[Tam and Lee, 2009] Tam, S. and Lee, H. (2009). MSF Whitepaper on Location Services in LTE Networks. White paper, MultiService Forum.

[Teunissen and Kleusberg, 1996] Teunissen, P. and Kleusberg, A. (1996). GPS Observation Equations and Positioning Concepts. In Kleusberg, A. and Teunissen, P., editors, *GPS for Geodesy*, volume 60 of *Lecture Notes in Earth Sciences*, pages 175–217. Berlin, Heidelberg: Springer.

[Thomas and Ros, 2005] Thomas, F. and Ros, L. (2005). Revisiting Trilateration for Robot Localization. *IEEE Transactions on Robotics*, 21(1):93–101.

[Vaitl, 1996] Vaitl, D. (1996). Interoception. *Biological Psychology*, 42:1–27.

[van Diggelen, 1997] van Diggelen, F. (1997). GPS and GPS+GLONASS RTK. In *Proceedings of the International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 1997)*, pages 139–144. ION (Institute of Navigation).

[van Waterschoot van der Gracht et al., 1928] van Waterschoot van der Gracht, W., Willis, B., Chamberlin, R. T., Joly, J., Molengraaff, G., Gregory, J., Wegener, A., Schuchert, C., Longwell, C. R., Taylor, F. B., Bowie, W., White, D., Singewald, J. T., and Berry, E. W., editors (1928). *Theory of Continental Drift: A Symposium on the Origin and Movement of Land Masses, both Inter-Continal and Itra-Continal, as proposed by Alfred Wegener*. The American Association of Petroleum Geologists.

[Vesely, 2010] Vesely, J. (2010). Differential Doppler Target Position Fix Computing Model. In *Proceedings of the International Conference on Circuits, Systems, Signals*, pages 284–287. WSEAS Press.

[Virrantaus et al., 2001] Virrantaus, K., Markkula, J., and Garmash, A. (2001). Developing GIS-Supported Location-Based Services. In *Proceedings of the International Workshop on Web Geographical Information Systems (WHIS' 2001)*, pages 423–432. IEEE.

[Wagner et al., 2006] Wagner, M., Wehner, R., and Wolf, H. (2006). The Ant Odometer: Stepping on Stilts and Stumps. *Science*, 312(5782):1965–1967.

[Wahlster et al., 2001] Wahlster, W., Baus, J., Kray, C., and Krüger, A. (2001). REAL: Ein ressourcenadaptierendes mobiles Navigationssystem. *Informatik Forschung und Entwicklung*, 16(4):233–241.

[Wahlster et al., 2006] Wahlster, W., Schwarzkopf, E., Sauermann, L., Roth-Berghofer, T., Pfalzgraf, A., Kiesel, M., Heckmann, D., Dengler, D., Dengel, A., and Sintek, M. (2006). Web 3.0: Convergence of Web 2.0 and the Semantic Web. *Technology Radar*, II:1–23.

[Wahlster and Tack, 1997] Wahlster, W. and Tack, W. (1997). Ressourcenadaptive kognitive Prozesse. In Jarke, M., Pasedach, K., and Pohl, K., editors, *Informatik 97 – Informatik als Innovationsmotor*, pages 51–57. Berlin, Heidelberg, New York: Springer.

[Want and Hopper, 1992] Want, R. and Hopper, A. (1992). Active Badges and Personal Interactive Computing Objects. *IEEE Transactions on Consumer Electronics*, 38(1):10–20.

[Want et al., 1992] Want, R., Hopper, A., Falcao, V., and Gibbons, J. (1992). The Active Badge Location System. *ACM Transactions on Information Systems*, 10(1):91–102.

[Ward, 1998] Ward, A. (1998). *Sensor-driven Computing*. PhD thesis, University of Cambridge.

[Ward et al., 1997] Ward, A., Jones, A., and Hopper, A. (1997). A New Location Technique for the Active Office. *IEEE Personal Communications*, 4(5):42–47.

[Waßmuth, 2006] Waßmuth, S. (2006). Wegsuche in polygonbasierten Umgebungsmodellen. Fortgeschrittenen Praktikum, Saarland University.

[Waßmuth, 2008] Waßmuth, S. (2008). Wegsuche mit dem A*-Algorithmus in hierarchischen polygonbasierten Umgebungsmodellen. Master's thesis, Saarland University.

[Waters et al., 2011] Waters, D. W., Xhafa, A. E., Vedantham, R., and Ramasubramanian, K. (2011). Fine-Time Assistance for GNSS using Wi-Fi. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN'11)*, pages 1–4 (online publication, no pagination). IEEE.

[Wehner and Lanfranconi, 1981] Wehner, R. and Lanfranconi, B. (1981). What do the Ants Know about the Rotation of the Sky? *Nature*, 293(29):731–733.

[Weidmann, 1993] Weidmann, U. (1993). Transporttechnik der Fussgänger. Technical Report 90, IVT – Institut für Verkehrsplanung, Transporttechnik, Strassen- und Eisenbahnbau, ETH Zürich. second edition, 109 pages.

[Welch and Bishop, 2006] Welch, G. and Bishop, G. (2006). An Introduction to the Kalman Filter. Technical report, University of North Carolina at Chapel Hill.

[Wolpert and Ghahramani, 2000] Wolpert, D. M. and Ghahramani, Z. (2000). Computational Principles of Movement Neuroscience. *Nature Neuroscience*, 3:1212–1217.

[Xiao et al., 2011] Xiao, W., Ni, W., and Toh, Y. K. (2011). Integrated Wi-Fi Fingerprinting and Inertial Sensing for Indoor Positioning. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN'11)*, pages 1–6 (online publication, no pagination). IEEE.