Institut für Informatik
der Technischen Universität München

Parasitic Tracking for Ubiquitous Augmented Reality

Manuel Huber

Institut für Informatik
der Technischen Universität München

Parasitic Tracking for Ubiquitous Augmented Reality

Manuel J. Huber

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. H. Seidl

Prüfer der Dissertation:

1. Univ.-Prof. G. J. Klinker, Ph.D.
2. Prof. A. Hopper, University of Cambridge, Großbritannien

Die Dissertation wurde am 24.05.2011 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 28.10.2011 angenommen.

Abstract The determination of the spatial parameters (also called the "tracking problem") is one of most important and challenging parts of any Augmented Reality application. This is especially true considering the increasing convergence of Augmented Reality (AR) and Ubiquitous Computing, resulting in the field of "Ubiquitous Augmented Reality" (UAR). Traditionally AR tracking focuses on tightly controlled, high-accuracy tracking setups that aim to provide high degrees of immersion. Contrary to that, Ubiquitous Computing in general has lower demands on localization accuracy, but considers large-area or ubiquitous scenarios. The greatest challenge associated with such scenarios is that no dedicated tracking infrastructure, as it is commonly encountered in AR setups, may be assumed to be available.

Thus, to address the unique characteristics of such ubiquitous tracking scenarios and to make location data available in circumstances where no dedicated infrastructure is available, the notion of "Parasitic Tracking" is introduced. A natural strategy in such cases is to derive as much location information as possible from observing other aspects of the environment, such as infrastructure used for information technology. Thus, the aim of Parasitic Tracking is to identify suitable sensing technologies based on non-tracking related infrastructure and to dynamically combine these in order to derive a robust and transparent estimate of the UAR user's spatial parameters.

This approach was evaluated using different technologies. Radio-based communication methods are prime examples as infrastructures suitable for Parasitic Tracking and therefore IEEE 802.11 WiFi-based and passive, long-range RFID-based localization methods were implemented. By further using Software Defined Radio (SDR) technology, the availability and accessibility of radio parameters with physically meaningful interpretations was improved. Finally a robust, relative tracking system on the basis of low-cost, optical, odometric sensors was constructed and evaluated.

A crucial requirement to implement such Parasitic Tracking setups was the development of a framework suitable to handle the various aspects of ubiquitous tracking setups. The "Ubitrack" framework is based on the notions of spatial relationship graphs (SRGs) and graph transformations or graph patterns. Special emphasis was placed on dynamic sensor fusion and dynamic adaption to changing tracking situations. This includes a generic method to calibrate the relative lag of unknown sensors.

Zusammenfassung Die Bestimmung der räumlichen Zusammenhängen zwischen Benutzer und realen sowie virtuellen Objekten (auch als "Tracking Problem" bezeichnet) ist eines der wichtigsten und schwierigsten Probleme bei der Umsetzung von Augmented Reality (AR) Anwendungen. Dies wird insbesondere deutlich wenn man die Problemstellung des Bereichs "Ubiquitous Augmented Reality" (UAR) betrachtet, die durch die Konvergenz zwischen Augmented Reality und Ubiquitous Computing entsteht. Tracking in herkömmlichen AR Anwendungen ist meistens auf kleine, gut kontrollierbare Bereiche und auf hohe Tracking-Genauigkeit fokussiert. Szenarien im Bereich Ubiquitous Computing hingegen, kommen im allgemeinen mit geringerer Genauigkeit aus, setzten jedoch allgegenwärtige oder wenigstens weitläufige Verfügbarkeit vorraus. Die gröHerausforderung in solchen Szenarien ist die Tatsache, dass hier keine dedizierte Tracking-Infrastruktur, wie sie sonst für AR Anwendungen üblich ist, erwartet werden kann. Um den speziellen Eigenschaften solcher Tracking-Szenarien für UAR Rechnung zu tragen und Positionsbestimmung in Situationen in denen keine gesonderte Tracking-Infrastruktur verfügbar ist wird der Begriff des "Parasitären Trackings" eingeführt. Eine naheliegende Strategie ist so viel Information über die Position des UAR Benutzers durch Ausnutzen von sonstigen Eigenschaften der Umgebung, wie zum Beispiel Infrastrukturen zur Datenkommunikation, abzuleiten. Deshalb ist das Ziel von parasitärem Tracking, geeignete Sensor-Technologien zu identifizieren, die auf Infrastrukturen ohne Bezug zu Tracking beruhen und diese zu einer robusten und transparenten Schätzung der räumlichen Parameter eines UAR Benutzers zu kombinieren. Verschiedene Technologien wurden bezüglich dieses Ansatzes hin evaluiert. Da Ansätze basierend auf Datenfunk-Infrastrukturen ein wichtiges Beispiel für geeignete, parasitäre Tracking-Methoden darstellen, wurden IEEE 802.11 WLAN-Positionierung, sowie Positionsbestimmung mittels passiver RFID-Tags mit hoher Reichweite implementiert. Mittels der weiteren Verwendung von Software-Defined-Radio (SDR) Technologie wurde die Verfügbarkeit und Zugänglichkeit von Funkparametern mit physikalisch aussagekräftigen Interpretationen verbessert. Abschlie, wurde ein relativ messendes, robustes Tracking-System basierend auf optischen, weggebenden Sensoren entwickelt und evaluiert. Eine wesentliche Vorrausetzung zur Implementierung von parasitärem Tracking war die Entwicklung eines Frameworks um die verschiedenen Aspekte ubiquitären Trackings zu unterstützen. Die "Ubitrack"-Bibliothek basiert auf den Prinzipien des "Spatial-Relationship Graphs" (SRG) sowie entsprechender Graphtransformationen ("SRG Pattern"). Besonderer Wert wurde hier weiterhin auf die dynamische Fusion von Sensoren und der dynamischen Anpassung an sich verändernde Tracking-Situationen gelegt, was unter anderem ein allgemeines Verfahren zur zeitlichen Kalibrierung von unbekannten Sensoren voraussetzt.

My resarch and this thesis would never have happend without the support of a number of great people.

Thanks to Gudrun for giving me the chance and for supervising my PhD. Thanks for the countless amounts of good advice and for the fruitful discussions.

Thanks to Andy for being my second supervisor and for reviewing my thesis.

Thanks to my close collaborators in the various research endeavors, especially Miki, Florian, Daniel, Marcus, Pete, Benni and Christian.

Thanks to all of my colleagues at the "Fachbereich Augmented Reality" for the one of the best work climates imaginable and the great amounts fun.

Thanks to Christian for being my best friend and to his family Corinna, Lara and Alexander.

Thanks to my parents Sieglinde and Hans and to my sister Claudia for caring for me and for being a great family alltogether.

Contents

1 Introduction

Augmented Reality is a relatively new approach to seamlessly combine virtual and real objects in order to provide natural user interfaces to experience and interact with information in various contexts. Inside the field of Augmented Reality, *Ubiquitous Augmented Reality* (UAR) is one of the most active areas of research. Ubiquitous Augmented Reality can be seen as the convergence of the initially diverse domains of Ubiquitous Computing (UbiComp) and Augmented Reality (AR). While Ubiquitous Computing anticipates the emergence of large-scale, interactive computing in ubiquitous environments, it also predicts that at the same time the interface and the computing device themselves will fade into the background. The explicit interaction with computing devices should rather be replaced by seamless interaction with ubiquitous information itself. In this context, Augmented Reality can provide a natural interface to these kinds of information using real-time, three-dimensionally registered interfaces. The goal of this work is to further develop the notion of ubiquitous augmented reality and especially one of the main technologies needed to implement such setups, in order to attain the goal of realizing true ubiquitous augmented reality applications.

A crucial component of any AR framework or application is the determination of the spatial parameters of the user. This is generally called the *tracking problem* in augmented reality and is indeed one of the most fundamental preconditions to construct further AR concepts. While excellent solutions exist for small, static setups, Ubiquitous Augmented Reality further increases the complexity due to the unrestricted and non-homogeneous nature inherent to ubiquitous scenarios.

For ubiquitous tracking scenarios is it mandatory to dynamically adapt to changing environments and to deal with unknown and unmanageable environments. More specifically, contrary to controlled laboratory environments, the existence of dedicated tracking infrastructure cannot be assumed in a general UAR case. On the other hand, the explosive growth of mobile information technologies also implicate the almost assured existence of some kind of communications infrastructure at almost any time. This circumstance suggests a strategy to employ these kinds of infrastructure in a non-standard way to determine the UAR user's location, by purely observing the infrastructure's physical parameters.

To overcome the accompanying complexity, a formal and computer-readable way to describe tracking scenarios and methods is required that is able to capture the specific circumstances of every setup and each situation of the UAR user. Based on such a description automatic and computer-assisted reasoning can be performed, which en-

ables dynamic reconfiguration of the tracking solution in order to match the unstable conditions encountered in UAR scenarios. Due to the scale implied by a general ubiquitous tracking scenario, the solutions to such tracking tasks furthermore need to take distributed acquisition and distributed computation into consideration.

A crucial task to successfully realize ubiquitous tracking strategies is also to identify suitable infrastructure technologies and parameters which lend themselves to use in localization approaches. An immediate area, which seems to hold promise in this context, are the numerous kinds of radio-communication technologies, which offer various kinds of exploitable physical parameters. A drawback of such radio based infrastructures is the relatively high level of required technical understanding and technology to access these parameters. Examples of suitable approaches in this regard include Wi-Fi or RFID based infrastructure.

A final requirement for an ubiquitous tracking approach is the ability to bridge transient gaps tracking availability. In most current wide-area tracking scenarios, the area is usually divided into islands where high-fidelity tracking is available and surrounding areas where no localization is possible. Ubiquitous tracking solutions therefore need to dynamically fill these gaps to provide uninterrupted tracking data.

Based on these arguments, the corresponding problems and possible solutions are explored in this document. The structure of the following chapters is as follows:

Chapter 2 In this chapter the concepts of *Augmented Reality* and *Ubiquitous Computing* as well as related areas are reviewed. Based on these notions, two previous definitions of *Ubiquitous Augmented Reality* are shortly discussed. A further, novel approach to define UAR in the context of social realities is presented.

Chapter 3 For Ubiquitous Augmented Reality it is necessary to address the unique characteristics of uncontrollable environments. The aim of the *Parasitic Tracking* approach presented in this chapter is to provide best-effort localization or tracking, without relying on dedicated infrastructure and rather leveraging already existing infrastructure that is unrelated to the tracking task of the UAR user. This approach needs to deal with the problem that unrelated infrastructure can be characterized as indifferent, volatile and non-robust concerning this use-case, which in turn leads to inherently unreliable sensing methods. Thus it is necessary to identify suitable sensing technologies and to dynamically combine these in order to derive a robust and transparent estimate of the UAR user's spatial parameters.

Chapter 4 This chapter presents an approach to formalizing and automatic processing of ubiquitous tracking setups. The *Ubitrack* framework is based on the ideas of the Spatial Relationship Graph (SRG) and the Spatial Relationship Patterns. The SRG captures and abstracts the geometric foundation of a concrete tracking situation. Similarly, a Spatial Relationship Pattern abstracts the requirements and effects of tracking algorithms by graph transformations on an SRG. This enables

computer-aided and automatic reasoning about AR tracking setups. The framework also contains necessary means to instantiate derived solutions as a data flow network for real-time processing of sensor data. Special emphasis was placed on the ability to dynamically reconfigure each component. By further embedding these concepts in a distributed, centrally coordinated peer-to-peer architecture, a complete system for context dependent reconfiguration of tracking setups can be achieved.

Chapter 5 Part of the dynamic tracking reconfiguration aimed for by the Ubitrack framework is the dynamic fusion of sensor data. Due to the general, heterogeneous nature of ubiquitous tracking setups, generally unknown sensors need to be combined without a priori knowledge of their parameters. The correct aggregation of data from different sensors into correspondences can only be performed if the exact temporal relationship between two sensors is known. While the Ubitrack framework deals with timestamping and synchronization of different data sources by suitable interpolation, the relative temporal offset between timestamps generated or assigned to different sources still needs to be calibrated. In this chapter a generic method to determine this relative lag for arbitrary sensor combinations, as may be encountered in dynamic sensor fusion setups is presented.

Chapter 6 Since radio communication infrastructure is among the most promising approaches for parasitic tracking, basic radio terminology is presented in this chapter. Further fundamental approaches to radio localization based on different physical properties of radio wave propagation are reviewed. The use of Software Defined Radio (SDR) as a means to improve the availability and accessibility of relevant radio parameters is also discussed. By moving the interface of the radio antenna as close to the computer as possible, more parameters with physically meaningful interpretations can be accessed. This presents certain advantages to various localization approaches.

Chapter 7 While IEEE 802.11 Wi-Fi-tracking has already seen some amount of interest in location based computing, it is nevertheless still a prime candidate for the Parasitic Tracking approach. In this chapter an implementation based on an available SDR framework shows that by tightly controlling the reception parameters, improved range estimates to fixed WiFi-beacons can be computed, as compared to common off-the-shelf Wi-Fi devices. This enables the implementation of localization based on fixed Wi-Fi-anchors and physical interpretations of the signal measurements. The concept and setup of this implementation and experimental data are presented.

Chapter 8 In this chapter the localization based on passive, long-range RFID tags is examined. These tags are assumed to be installed at various parts of complex

assemblies or at prominent points in the environment, thus presenting another kind of infrastructure. By detecting a cloud of individually identifiable tags the location of the reader device is deduced. Based on this approach, two scenarios are presented. In the first scenario the localization inside an aircraft fuselage for guided maintenance workers is derived by fitting the characteristic pattern of a directional RFID antenna into the currently visible cloud. To further improve the optimization and to provide a complete set of spatial parameters, this position is combined with the orientation from a gyroscope. The second scenario evaluated the localization based on range estimates to the individual tags based on an radio propagation model. This is again possible by using SDR technology that allows quick reconfiguration of the tag detection process.

Chapter 9 To bridge gaps in tracking setups, the localization method presented in this chapter aims to provide robust, relative tracking in cases where no other methods are available. The basic approach is to estimate the relative movement of the user by measuring the motion on the floor using optical odometric sensors. A prototype based on low-cost sensors and its evaluation are described. Finally a specialized high-end optical odometric sensor system is presented in order to analyze the tracking errors produced by the low-cost system and to evaluate its future potential.

Chapter 10 The last chapter concludes the discussion by reviewing the results and by highlighting intended future work.

2 Ubiquitous Augmented Reality

As previously mentioned the aim of this document is to further the development of the ubiquitous augmented reality research endeavor. While this comparatively young research vision has by now been generally accepted by the community at large, it still lacks concise definition that goes beyond its implicit declaration based on postulates.

To start off with the discussion in this area, this chapter presents short introductions into the definitions of augmented reality and ubiquitous computing. The two commonly recognized definitions of ubiquitous augmented reality are further discussed and contrasted by a novel definition proposed by the author.

2.1 Augmented Reality

The Ultimate Display The history of Augmented Reality can be traced back at least to the influential article "The Ultimate Display" by Ivan Sutherland published in 1965 [Sut65]. In this article the vision of a type of display is formulated that extends or replaces all the perceptible aspects of the user's physical world in a computer controlled fashion.

The continuation of this work resulted in the development of the first head mounted display and head-tracking system, the "Sword of Damocles" in 1968, also by Ivan Sutherland's group. This system was the first to demonstrate virtual and augmented reality features.

Following this first demonstration a number of different research issues were raised and ultimately new research areas in the context of virtual environments emerged. While the concept of virtual reality was predominant in the beginning, augmented reality aspects continued to be researched in this context.

The Milgram-Continuum Especially since the beginning of the last decade of the 20th century, the importance of augmented reality became apparent which resulted in a more widespread use of the term, albeit with varying implicit definitions. This motivated the rigorous classification of the various different types of virtual or augmented reality interaction modes. Milgram et al. ([MK94], [MTUK94]) define a three dimensional design space in which each virtual environment display can be located according to its properties of "World Knowledge", "Representation Fidelity" and "Presence Metaphor". By defining different regions in this space different areas of applications can be identified.

The simplified and more widely known one dimensional representation of this classification scheme is also known as the *Reality-Virtuality-Continuum* or *Milgram-Continuum*. Here virtual environment applications are ordered according to their respective degrees of immersion into reality or virtuality, with pure reality on one end, complete virtual reality at the other and various types of mixed reality in between. It is shown in figure 2.1. This method places Augmented Reality towards the real side of its spectrum since AR only augments the pure reality by few virtual objects.

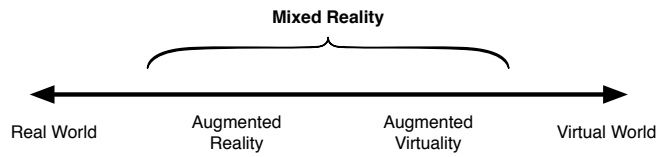


Figure 2.1: Simplified Migram-Continuum; based on [MK94]

Azuma's Criteria A further characterization of AR was given by Ron Azuma, as published in "A Survey of Augmented Reality" in 1997 [Azu97]. To identify relevant applications while simultaneously not being overly dependant on technology based distinctions the following criteria try to capture the essence of the interaction with an AR system.

An augmented reality system thus has to have the following properties:

- Needs to combine real and virtual environments
- Needs to be interactive in real-time
- Has to be registered in 3D

These criteria still serve as the primary guideline and selection criterion in the AR community.

Some successful applications of augmented reality in various contexts have since emerged. Example scenarios can be found in industrial, automotive and medical AR, as well as in augmented reality for entertainment. Industrial augmented reality often supports the worker in an industrial context such as welding (see [ESK+03]) or by visualizing debug information for prototyping (see [KNU+07]). A similar visualization approach for sensor related debug information, albeit in an automotive context, was presented in [TLWK07]. Other use of AR in cars include navigation, night-vision or assistance at intersections. Similarly, medical augmented reality can help planning and diagnosing medical conditions. An overview of different medical AR approaches can be found in [SFN08]. Finally, as an example for the use of augmented reality for entertainment, the well known "ARQuake" ([TCD+00]) demonstration should be mentioned. For a further introduction to Augmented Reality and its applications, see for example [Tön10].

All these examples share the common trait that they operate best in tightly controlled environments, such as the operating theater in medical AR or a specially devised workshop in industrial AR. One of the current challenges is the deployment of robust AR applications outside of such lab-like environments.

2.2 Ubiquitous Computing

Another area of research, which is of fundamental importance, is Ubiquitous Computing. This section will briefly summarize important points about this area.

The underlying vision of Ubiquitous Computing and the related fields were first formulated by Mark Weiser in 1991 ([Wei91], [Wei93]). He derives the main position of this vision as a consequence of the clash between the simultaneous trends that the amount of information that surrounds us and with which we constantly interact steadily increases, while at the same time the computer interfaces which we use become more and more unobtrusive, even self-effacing. We are interacting with an abundance of data, which increasingly is digitized, while the interfaces themselves move into the background.

The vision of ubiquitous computing can thus be formulated as the computer which disappears into our everyday environment and enables ubiquitous interaction with all kinds of information. At the same time the computer and the direct interaction with it retreats into the background.

A number of different disciplines have since emerged, such as "Ubiquitous Computing", "Pervasive Computing" and "Ambient Intelligence" amongst others. The following short characterization aims to help distinguish these disciplines, but makes no claim to be complete or to capture the true essence of any discipline.

Out of these the field of "Ubiquitous Computing" most closely follows the original vision and is thus the most direct descendant. Here emphasis is on the integration of computer services into everyday objects and their interconnectedness. The actual computer interfaces aim to be unobtrusive but may nevertheless be perceived as such. Examples are the computer and information displays added to walls or desks as envisioned by Weiser. Services and objects can also be user centered in the sense that they are in the possession of the users and travel with them.

The aim of "Pervasive Computing" is to permeate all aspects of the daily life of a user. The emphasis on the disappearance of the computer and its interface is here less pronounced than in other disciplines. The focus lies on the actions that the user performs and how these can be supported. As a consequence pervasive computing also includes the use of mobile computing devices such as smartphones.

Contrary to these two disciplines "Ambient Intelligence" aims to move the computing power away from the user completely and into the environment itself. It is thus the room which determines the user's context and reacts intelligently to the perceived user's needs and goals, without depending on any computing device on the user. Further research

aims to completely hide the computer and uses "Ambient Interfaces" to interact with the user.

A different, but also related area of Ubiquitous Computing is the field of *Wireless Sensor Networks*, which is more concerned with the technical aspects of distributing autonomous sensor nodes throughout the environment. For an overview see for example [ASSC02]. Topics of research include, but are not limited to, sensing technologies, energy efficiency, wireless communications or sensor localization.

2.3 Ubiquitous Augmented Reality

Building upon these definitions of Augmented Reality and Ubiquitous Computing, the field of "Ubiquitous Augmented Reality" (UAR) emerged.

In the following sections three independent characterizations of this field are given. The aim is to stimulate the discussion about the concrete goals and applications of Ubiquitous Augmented Reality.

Convergence of AR and UbiComp While the first connections between Ubiquitous Computing and Augmented Reality were first established by Newman in 2001 ([NIH01]), the first definition of the term "Ubiquitous Augmented Reality" was given by MacWilliams in 2003 ([MSW+03]).

The definition of MacWilliams essentially combines the formalistic definitions of AR and Ubiquitous Computing, as given by Azuma and Weiser respectively. Accordingly MacWilliams defines Ubiquitous Augmented Reality (UAR) ([Mac04], [Mac05]):

- augments the real world with virtual information,
- is interactive in real time,
- is spatially registered,
- is available throughout a large physical environment,
- and allows both immersive interaction and unobtrusive assistance.

In this definition, both the notions of Augmented Reality and Ubiquitous Computing can be identified. This definition thus performs a fusion of these, at first glance quite incompatible, concepts and postulates Ubiquitous Augmented Reality as a new notion. This definition had a significant impact on the field of Augmented Reality. Nevertheless it can also be limiting on the emergent vision behind Ubiquitous Augmented Reality.

While the function of the individual technical components of Ubiquitous Augmented Reality can be derived from this definition, its actual significance remains peripheral. This illustrates the fact that the definition is mostly technologically driven. Furthermore the possible characterization of the derived components is coarse and as their individual development advances, the need of a coherent narrative is crucial.

The Milgram-Weiser-Continuum In analogy to the three dimensional Milgram-Continuum discussed earlier, Newman([NBP+07]) defined the two dimensional *Milgram-Weiser-Continuum*. By placing the dimensions "Reality – Virtuality" of the simplified Milgram continuum and "Monolithic – Ubiquitous Computing" (in analogy to Weiser) on perpendicular axes, a plane emerges on which different areas of mixed/virtual reality applications can be located. The continuum, as it was presented by Newman, is shown in figure 2.2.

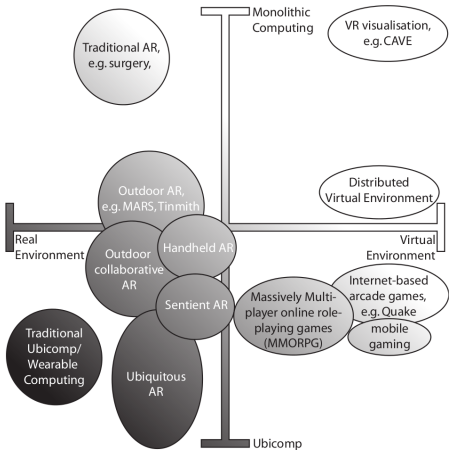


Figure 2.2: Migram-Weiser-Continuum; from [NBP+07]

This also leads to a further, implicit definition of Ubiquitous Augmented Reality, as the area where certain computing aspects of Augmented Reality and Ubiquitous Computing overlap in a distinct fashion.

Thus as the previous definition already suggested, Ubiquitous Augmented Reality can be seen as the concepts of Augmented Reality, applied on a Ubiquitous Computing scale. This implicit definition, or rather characterization, is illuminating regarding the relationship between Ubiquitous Augmented Reality and other disciplines in the field of Augmented Reality. The explicit vision on how this application of the ubiquitous environment to the AR approach could be realized is still vague.

Social Construction of UAR In order to further stimulate the discussion about the current and future role of Ubiquitous Augmented Reality, the author would like to present a definition of UAR based on sociological observations.

While the definitions of both AR and UbiComp are based on their respective vision, the concrete vision of UAR so far has been constructed after the fact. It is thus necessary to develop a strong vision before formalistic definitions can be applied. While implicit definitions are valuable for showing the relative positions and differences to other fields, they lack in-depth discussions about the actual ramifications of these differences.

The technologically oriented or implicitly given definitions so far show some limitations as they do not directly deliver a suitable, underlying narrative. Thus the development of further visions or scenarios on the extended application of Ubiquitous Augmented Reality is made unnecessarily hard. This furthermore disperses research and slows down the emergence of a common foundation.

It is the belief of the author that the vision defining the primary concern of Ubiquitous Augmented Reality should be stated as a social vision.

In the same vein in which Ubiquitous Computing represents a step away from confined laboratory setups of computing interaction and moves towards general, everyday interaction, Ubiquitous Augmented Reality should aim towards the support of daily life. Thus the support of everyday social human interaction is the topic of Ubiquitous Augmented Reality.

By going away from being tools used for specialized tasks in specialized environments, Ubiquitous Augmented Reality becomes a method of communication in everyday social realities. Since this phenomenon is of social nature, it is necessary to look at the social ramifications of using Augmented Reality in a Ubiquitous, everyday sense. The transformation of AR from confined laboratory setups into the world at large, simultaneously represents a transformation of the social context in which AR exists and operates.

To understand the position argued below, it is first necessary to realize that everyday human life already incorporates various degrees of malleable realities or "virtual" realities.

A prime and rather often used example of such a reality is the concept of "money". The reality of money is purely arbitrary since nothing in nature mandates its existence but is rather specific to the social interactions conducted on its basis. On the other hand in our current, western-oriented culture almost nobody would disagree on the apparent reality of money and its impact on our everyday lives. While the reality of money, although it may appear rigid in its daily use, exhibits significant differences between different social circles and different periods in history, these differences can manifest themselves both in appearance and perception of the concept. Thus the existence of "money" is a flexible construct, negotiated by and between the participants of a given social circle at a specific period in time.

In the field of social sciences and psychology the discipline of *Social Constructionism* explores these connections in more detail. While Social Constructionism is mostly concerned with the ramifications of social constructions of knowledge on the methods used in psychology, the author believes that an awareness of the nature of socially constructed reality is also beneficial to the discussion of Augmented Reality.

Social Constructionism The essence of the ideas defining social constructionism or at least underlying most of its representations, can shortly be summarized as follows. This should not represent a comprehensive discussion on the actual characteristics of social constructionism or the specific affiliation of certain practitioners. Instead, an overview

of the ideas on which most directions of social constructions appear to agree upon will be given. More details and discussions of this topic can be found for example in [BL67], [Bur03] or [LS10].

It should also again be noted that the statements below concerning the shared knowledge of the world refer to the social world of human everyday life. The existence of an objective, physical reality is not subject of social constructionism, at least in this context.

In this respect, a broad characterization of social constructionism can be given as follows.

Language is social action Of central interest to social constructionism are the everyday interactions between people inside various social contexts. Social constructionism does not see these interactions and especially the language used as a passive mode of transportation for information or thoughts from one person to another. Instead, language and social interaction actively shape the knowledge about the world in the social context and thus the social world itself. When people talk to each other, the everyday world is being constructed in the understanding of the people involved in these social interactions. This is a form of social action in itself.

Language as pre-condition of thought Language in this context is not limited to the spoken or written word, but includes any type of foundation on which inter-personal communication is carried out. As stated above, the means by which humans understand the social world is not derived from any objective reality, but is carried out through social interaction with other people. Thus language forms the fundamental framework by which the perception and comprehension of the world is shaped.

Sustenance of knowledge Social constructionism further states that the primary method through which knowledge is sustained in a social circle is by social processes. The everyday social interactions inside a social circle are the main method by which the shared knowledge about the world is created. In this context the shared knowledge about the world can be equated with social reality itself. Thus social reality is constructed and sustained by social interaction.

Cultural and historical specificity of knowledge Since language is a social process and the knowledge about the world is sustained using language, it follows that the knowledge about the world is also defined by social processes. As a consequence the ways to understand the world depend on cultural and historical circumstances. More specifically the shared knowledge and thus the social reality is specific to the individual social circles and the particular time. Thus at any given time, different social realities may simultaneously exist in parallel or a single social reality can evolve or be transformed over time.

Knowledge and social action go together Due to the different social realities existing for different historical or social circumstances, different actions emerge similarly. Each social construction of shared knowledge mandates or sustains certain actions, whereas others may be excluded. Thus the differences in social realities also have consequences in the practical sense.

Non-Essentialism A further aspect of social constructionism is the anti-essentialist view of human nature. Since the social world is a product of social process and dependent on the particular circumstances, there can be no fixed, determined nature of the social world of people. This leads to the further conclusion that there are no essences associated with or deducible from individual human beings. This goes beyond the idea that the categories of human behavior are formed by empirical, external influences, but rather states that there is no intrinsic nature of human beings in the first place. This forms an explicit contrast to the essentialistic view held by other disciplines of psychology, for example psychoanalysis.

Questioning Realism As a final note it should be mentioned that, as stated above, social constructionism deals primarily with the social realities of humans, especially with quotidian realities. There are however attempts to apply this reasoning in a more general sense to physical reality and to question the legitimacy of such an objective reality. This discussion inevitably involves the distinction between epistemological and ontological reasoning and will not receive further discussion here. The question about the nature of objective reality is beyond the scope of this document.

Thus the idea of social constructionism can be stated that parts of our everyday realities are constructed via the interaction inside the various groups and social circles that define our social identity. As mentioned above, the main attention of social constructionism as a discipline is directed to consequences for psychological treatment.Nevertheless, the application of this idea on the setting of augmented reality and ubiquitous augmented reality leads to interesting discussions.

The resulting vision of Ubiquitous Augmented Reality based on the social constructedness of quotidian reality could be formulated as such:

Apart from the material, primary reality, there exist a number of different, socially constructed realites. The existence of these realities is generated by the communication about these constructs inside social circles. Thus the concrete manifestations of these realities are dependent on the contingent aspects of the social group in question.

Due to the current trend to increasingly include telecommunications technologies and ubiquitous computing into our social life, the number and immediate presence of these realities become more substantial. An increasing amount of information pertaining to our social and material environment is present and immediately available at all times. The ubiquity of computation and information technology make the surrounding data available and also reinforce the existence of this sea of information.

As a consequence, supplementary information can be associated with almost any entity of interaction in our daily life and additionally, purely constructed information may be present. Some of this information may even have direct spatial relationships to the material world. Nevertheless these supplementary, virtual information layers can be regarded as additional layers of reality, as long as socially agreed upon. This social construction only very rarely happens explicitly, but typically is emergent from the social communication about the constructed facts.

In this context Augmented Reality is transformed from a purely technical aspect into a method to perceive and interact with different layers of constructed, social realities. Augmented Reality provides a natural interface to make these constructed realities directly perceivable and to interact with them. This may also be expressed in the shift of talking about augmented reality as "augmenting some objective reality by virtual elements", but rather that augmented reality provides access to combine the material reality with various layers of socially constructed realities. Especially in ubiquitous, quotidian settings this distinction becomes relevant, since there the multitude of different layers of constructions becomes apparent. In contrast, in an industrial or laboratory setting this distinction is redundant.

The degree with which these constructed reality layers can be spatially registered with the real world varies depending on the nature of the information. This spatial relationship is anchored to a specific entity and thus is usually also mobile, whereas in other cases constructed reality may also be fixed in the world. Based on this, different ways of augmentation and interaction may be suitable.

In a similar way Augmented Reality has always changed the perception of constructed realities, as was also discussed for example by [Dav10]. By visualizing and communicating the existence of virtual objects, an additional layer of reality is created, albeit usually in rather confined contexts.

A principle distinction can be made between augmented reality which highlights and or annotates already existing, material objects and augmented reality which adds additional, purely virtual objects into our space.

2.4 Anatomy of a UAR application

From these descriptions, the basic architecture of a ubiquitous augmented reality application can be derived. In the following a short outline of the various necessary elements is given. The basic idea is that a UAR application can be split into three main components which interact and complement each other. How to define and implement each of these components is a separate research endeavor in itself and many open questions in these areas are currently being investigated.

UbiTrack *Ubiquitous tracking* aims to provide solutions to the physical tracking problem in the context of ubiquitous scenarios. A fundamental requirement of any

augmented reality application is that the system has to keep track of the spatial parameters of the user and the real objects with which the user interacts. For classical AR augmentations, this mostly concerns the complete position and orientation in three dimensions. This spatial characterization with six degrees of freedom (6DoF) is also called the *pose* and the capturing or tracking of these parameters is referred to as the *tracking problem* in augmented reality. For an overview of different tracking technologies commonly used in virtual or augmented reality setups see [RDB01] or [WF02].

For the ubiquitous tracking problem, this is extended to provide the spatial parameters of the user and all mobile objects at any time in a ubiquitous augmented reality scenario. Compared to the other modules, this area has thus seen the most attention and thus its concepts are the most developed so far. This module, the further development of concepts for ubiquitous tracking and the adaption to ubiquitous tracking scenarios will be the topic of the following chapters.

UbiDisp *Ubiquitous displays* deal with the corresponding problem on how to bring the relevant augmentation into the world in a ubiquitous scenario. In order to provide an augmented reality interface to the individual data, an application has to present some kind of visualization or other augmentation of reality. The methods by which this can be performed depend on the equipment available, the required display fidelity or degree of immersion as well as the concrete situation of the user. Similar to the ubiquitous tracking problem, this question is further complicated when taking the nature of ubiquitous augmented reality scenarios into consideration, rendering a number of classical AR display approaches impractical.

While the topic of "ubiquitously distributed presentation surfaces" is related, the focus of *ubiquitous displays* is different. The former is a technical solution to a specific circumstance and may be included in ubiquitous display solutions, whereas the focus of UbiDisp in general is rather the exploration and the classification of different display strategies and how they relate to different situations occurring in the context of a UAR application. For example mobile devices, interactive 3D displays and projective augmented reality may well be integrated in a single ubiquitous augmented reality application, albeit for different aspects. Currently active research on ubiquitous displays is emerging and is mostly concerned with classification of different augmentation strategies (see for example [TPK09] and [Ple11]).

UbiInteract *Ubiquitous interaction* finally deals with the question on how the user can interact in a ubiquitous environment with such ubiquitous augmentations. As this largely depends on both the concepts of ubiquitous tracking and ubiquitous displays, this topic has so far seen the least attention, and its scope and its concrete questions are still emerging.

3 Parasitic Tracking

Parts of the description below are loosely based on the presentation [Hub09] given at PerCom 2009.

The Ubitrack tracking approach, is the most fundamental requirement to enable ubiquitous tracking in dynamic and distributed environments. In general the tracking situation of a mixed reality user with all its associated parameters is always a balance between technical requirements, economic factors and the desires of the user.

The usual approach for augmented reality applications so far mostly assumes that the tracking methods are controlled by and dedicated to the application at hand. Usually this is achieved by deploying application-specific tracking infrastructure. This infrastructure can have many different forms, ranging from outside-in tracking cameras to artificial optical landmarks. The common characteristic is that the operating environment of the AR user is deliberately modified to enable the user's positioning and tracking. Dedicated tracking infrastructure also offers the benefit of being tailored to the application at hand in terms of accuracy and interaction fidelity. Especially high-precision augmentations (for example medical AR) cannot operate without such installations.

Ubiquitous augmented reality on the other hand envisions the pervasive use of AR for private, public and commercial social use in everyday circumstances. Example scenarios include AR based interactions in public offices, shopping centers or during recreational activities, or the assistance offered to a business person in a foreign town who wants to combine business tasks with sightseeing. All these examples require tracking information in large, unknown and inherently uncontrollable environments.

Nevertheless, in today's everyday world the permeation with suitable, dedicated tracking setups still is rather low. Thus this direct approach may not (yet) be applicable to the tracking problem in the case of ubiquitous augmented reality. Mostly due to the volatile nature and obvious vastness of a ubiquitous tracking scenario, no presumed single tracking system can be relied upon ubiquitously. Thus the UAR user is forced to rely on solutions that do not depend on particular tracking-infrastructure, such as inertial or inside-out tracking. Furthermore, due to the wide range of different Ubiquitous Augmented Reality applications, a similarly large range of different tracking requirements exist. While for some tasks the availability of high accuracy tracking is required, other tasks (such as navigation) can be supported with lower accuracy or fewer degrees of freedom.

In the light of these constraints, this thesis introduces and motivates the concept of "Parasitic Tracking" which aims to enable best-effort infrastructure based tracking

without the requirement of dedicated infrastructure. By using localization-unrelated technologies in atypical ways location information can be derived. This transforms for example privately owned Wi-Fi routers or RFID tags for logistics purposes into infrastructure useable for position estimation.

3.1 Characteristics Ubiquitous Tracking Environments

In order to motivate the need for Parasitic Tracking and to derive its concrete aspects, it is first necessary to review the general situation in which a user of ubiquitous augmented reality is expected to operate.

3.1.1 Infeasibility of Dedicated Tracking

In general, a ubiquitous augmented reality scenario may include situations where dedicated tracking infrastructure is indeed available. In such exceptional cases the tracking data provided is usually superior to any localization derived otherwise and thus naturally should be utilized. However, in most cases, the ubiquitous augmented reality application also needs to be able to handle situations where no such infrastructure is installed. The lack of infrastructure in such situations can have a variety of reasons, of which the major ones will be discussed in the following.

Cost The main and most obvious obstacle to providing a homogeneous, ubiquitous tracking solution is the extremely high cost associated with deployment. Usually, the scope of ubiquitous augmented reality scenarios ranges from rather local wide-area scenarios to urban scale scenarios or even global scenarios. Thus the required covered area alone is in most cases a prohibitive factor for comprehensive coverage with dedicated deployment of tracking infrastructure, especially when considering that the standard, single installation of an augmented reality tracking solution alone may range in the thousands of euros. Apart from that, the range of traditional high-precision tracking systems in general is rather limited, this additionally multiplies the number of required deployed units. Thus, from a purely economic point of view, no homogeneous tracking method encompassing ubiquitous augmented reality needs is foreseeable.

Maintainability Apart from initial costs, tracking infrastructure also generates high maintenance cost. Especially high precision tracking systems require regular recalibration. Such tasks usually have to be performed by trained measurement engineers, this again increases the associated cost. Furthermore the scopes of responsibility in such deployments may not be clear at all. This leads to the next problem.

Pervasion While the use of homogeneous sensors for wide-area coverage might in principle still be feasible for a single organization such possibilities vanish as soon as a complete permeation in all situations is desired. First the desired tracking environments may encompass the scope of multiple entities and their authority. For example, theoretically, a location service in the open street could be provided by the municipality. But as soon as offices, industrial workshops or even private homes are to be included, the scope of responsibilities becomes unclear. This affects deployment and maintenance as well as privacy issues. No single tracking technology can be assumed to be appropriate for all environments. For example, specific scenarios may have different challenges associated with certain tracking areas. Thus again, no homogeneous, single tracking infrastructure will be able to cover the many different environments that a general ubiquitous augmented reality scenario aims to cover.

Undesired / Cosmetic Another criterion could be the obtrusiveness of such installations. For example in AR scenarios aimed at design evaluation, tracking cameras can be perceived as a distraction and can, in the worst case, render the application itself futile. In private homes as well, the installation of any kind of dedicated infrastructure may simply be undesired.

The result of this discussion is that, in general, a UAR system or a UAR application may not depend on the existence of any specific infrastructure for tracking. Thus, in order to enable UAR applications, it is necessary to enable tracking or positioning in real-world sensing scenarios.

3.1.2 Information Society

These problems show that no single tracking technology is available that could cover the entire range of different ubiquitous tracking setups. Dedicated tracking infrastructure in general cannot be expected to be available, either. So, in order to enable ubiquitous localization despite these difficulties, a different view of the environment must be taken.

This section looks at some examples and reasons. Apart from dedicated tracking infrastructure, different kinds of other infrastructure is available in various contexts. One prime example is the success of pervasive computing in our everyday lives.

The technological advances in mobile computing and the growing diffusion of personal information displays have led to the emergence of an information oriented society. A rapidly increasing number of social interactions is performed using various information technology based services. The applications basically cover the entire spectrum of daily life, ranging from business to private communications.

Inherent with the rise of these social tools is the rise of the corresponding information infrastructure, in order to be able to exchange the relevant data. The main technologies used in this context are Wi-Fi and cellular networks.

Other kinds of infrastructure which might be available include other types of wireless communications, ranging from radio broadcasts to personal wireless networks such as bluetooth. RFID technology used for logistics and inventory management also serves as a different kind of information technology infrastructure.

This leads to the conclusion that while dedicated localization or tracking infrastructure may not be available in the foreseeable future, environments relevant to ubiquitous augmented reality exhibit other kinds of available infrastructure.

3.1.3 Characteristics of UAR Environments

As has just been established there are different kinds of infrastructure available. Thus, the various challenges posed by the environments in which ubiquitous augmented reality applications may operate, can be summarized as follows:

Indifferent Environment Since no dedicated location or tracking infrastructure is generally available, the environment can be described as generally indifferent towards the tracking needs of the ubiquitous augmented reality user. This is because, on the one hand, no effort can be expected to explicitly provide location or tracking data. On the other hand, the environment provides other types of infrastructure, in which the user can participate. Usually non-typical or non-intended use of those other kinds of infrastructure is tolerated, as long as no disruption of the service is the consequence. But this indifference also means that any subtle environmental characteristics, which are not directly related to the correct functionality of the infrastructure may change at any time.

Volatile Environment Second, the environment and thus the characteristics of the available infrastructures are generally not under the control of the augmented reality user. The environment and its concrete characteristics may be greatly varying over time. Furthermore, due to the indifference of the environment towards the tracking needs of the ubiquitous augmented reality user, changes usually happen without prior notice. One example where the characteristics of the environment change without affecting the basic intent of the corresponding infrastructure would be the addition of a wireless router to an existing domain. In order to improve reception in certain areas, the operator may decide to add additional access points. While this does not change the existence of the infrastructure itself, and indeed is transparent to the typical users, it nevertheless changes the underlying characteristics of the environment.

No guaranteed availability A further aspect of such non-tracking related kinds of infrastructure is that there are usually no guarantees of availability. This is partly due to the indifference of the environment and partly due to the fact that this kind of infrastructure usually is not in the authorative domain of the user. While major, public infrastructure such as cellular networks usually have very good availability and may even

offer guarantees, the same is not the case for smaller or privately owned infrastructure. In the extreme case of infrastructure which is not directly intended for public use, but which is publically visible, disruptions may occur at any time and may possibly not even be restored.

Unreliable Sensing Methods The consequence of these observations is that in a ubiquitous augmented reality scenario any sensing method relying on any single infrastructure is assumned to be unreliable. As the environment and any of its characteristics may change without notice or become unavailable, any associated sensing method may produce incorrect data based on false assumptions or be unavailable. This leads to the requirement that a successful and robust sensing strategy has to include multiple methods of observing its environment. Combining multiple sensing methods on the other hand enables the user to accommodate to single changes and to update the user's view of the world accordingly. This assumes that the rate of change of the environment is in general slow enough for the user to be able to react sufficiently fast.

In combination, the requirements of ubiquitous augmented reality and the characteristics of the environments in which it should operate lead to the idea of *Parasitic Tracking*.

3.2 Parasitic Tracking Approach

Relating to all these challenges and to the requirements of tracking in ubiquitous augmented reality scenarios, this thesis introduces the "Parasitic Tracking" approach. In short the idea of Parasitic Tracking is to derive the location or pose of the user from any clues the environment may offer. The resulting estimate is always a best-effort guess obtained by extracting as much information as possible from the user's current situation. This approach relies on two basic strategies to transform unrelated environmental characteristics into a robust tracking infrastructure.

The first and main idea is to exploit any observable characteristics of not tracking-related infrastructures that happen to be available in the user's environment at any specific time. By sensing these features and comparing them to known attributes of the environment, it is possible to gain knowledge about the position of the user in relation to this infrastructure. Thus, this facilitates location awareness or even pose tracking without dedicated infrastructure.

Note that the "exploitation" in the parasitic tracking sense is always limited to observations of features the particular infrastructure exhibits. While in some cases the observation first requires the initiation of communication from the user's side, in no case should the performance of the primary function of the affected infrastructure be degraded. That is, any sensing for parasitic tracking must not disrupt the host infrastructure.

Second, as it was also discussed above, any single tracking infrastructure cannot be

relied upon indiscriminately. The infrastructure may change or become unavailable at any time, leading to wrong estimates or tracking failures. To account for these circumstances a natural strategy is to diversify the methods used to estimate the user's position. In a ubiquitous tracking environment, there can usually be multiple, independent kinds of infrastructure assumed to coexist. Thus different parameters from different tracking-unrelated infrastructures can be combined into a single, reliable tracking infrastructure. This can be achieved using dynamic sensor fusion methods.

Note that these sensor fusion techniques can also help in updating the localization models used with the various infrastructures. For example, suppose the user encounters a changed environment which no longer conforms to the user's knowledge. In this case the actually sensed parameters together with a position estimate obtained from other available infrastructure can be used to update this knowledge.

By this mechanism, the parasitic tracking framework can adapt to changes in the environment with the help of redundant position estimation methods. This is supported by the assumption that the individual infrastructures are independently operated and thus generally do not change simultaneously.

The main task is to identify the type of available infrastructure, together with the exploitable parameters and to devise methods to determine localization from the observed behavior of the infrastructure.

3.2.1 Infrastructure Mediated Sensing

The class of human activity monitoring approaches called *Infrastructure Mediated Sensing* was introduced by [PTA06], [PRA08] and [Pat08]. This class shares certain similarities to the parasitic tracking approach and a brief comparison will be presented in this section.

Infrastructure Mediated Sensing Approach Infrastructure mediated sensing represents a general approach to implement systems for monitoring human activity in a home environment. Such systems are important for various pervasive or ubiquitous computing scenarios, such as intelligent home automation or health care applications. The information obtained may include, but is not limited to, the location of the user as well as the classification of tasks performed by the user. The difference between Infrastructure mediated sensing and more direct sensing approaches is that the former avoids the installation of additional, dedicated infrastructure. Instead, the already existing infrastructure in the home is used and its characteristics are monitored, with the assumption that actions of the user are observable as changes in the infrastructure.

In the scenario presented in [PRA08] for example, the air-circulation system of an home heating, ventilation and air conditioning (HVAC) installation is examined. High-resolution pressure sensors were installed in the central air handler of the system. By instrumenting the air circulation infrastructure in such a way, small perturbations in the

pressure in various areas of the home can be measured. These pressure changes can be translated to events, such as passage of a human in front of an air duct or the opening or closing of a door. The infrastructure thus forwards actions performed by the user as observable changes in its own parameters; it acts as a proxy for the actions of the user.

Other scenarios which were examined include monitoring of electrical noise in the line power or acoustical monitoring of the plumbing systems.

Comparison to parasitic tracking As mentioned, the infrastructure mediated sensing and the parasitic tracking approaches share some similar elements.

The most prominent aspect shared by both approaches is the desire to leave the user's environment untouched. Instead various observable facts of the already existing kinds of unrelated infrastructure are exploited to obtain as much information as possible, while still being unobtrusive. In this sense [PRA08] also refers to infrastructure mediated sensing as "home bus snooping".

Also both infrastructure mediated sensing and parasitic tracking share the same general goal. Contextual information of the user should be determined, but in the presence of the limitations imposed by the reluctance or overall infeasibility of dedicated infrastructure.

At the same time there are also differences between the two approaches. While parasitic tracking aims to purely observe the infrastructure in question from outside, infrastructure mediated sensing is mostly integrated into the infrastructure, although in an unobtrusive way. Central points for instrumentation are identified, such that changes effected by the user can be sensed, while keeping the changes required as minimal as possible. It thus represents an outside-in approach which observes the actions of the user inside the environment.

On the other hand, parasitic tracking represents an inside-out approach and assumes that the infrastructure itself is static and independent of the user. Instead, the user himself determines his location by observing the contingent aspects of the infrastructure in question at his specific location.

3.3 Localization approaches

The topics of location awareness and position estimation have already seen considerable interest and continue to be active topics of research for many different disciplines. Accordingly there is a number of both different observed infrastructures (such as *GSM*, *WiFi*, *DECT*, custom hardware, etc.) as well as different approaches to derive the position estimate from these observations.

To harness these technologies in the context of ubiquitous computing applications, a number of different frameworks to handle the user's context or location information have

been established. Examples for such systems include the *Nexus* platform for spatial-aware applications ([HKL+99]). The *QoSDREAM* project ([Cou02]) aims to implement an architecture to for reconfigurable and distributed multimedia applications and also handles unified location information obtained from various tracking sources. The *context toolkit* ([DSA01] treats location at a more abstract level as a part of the user's environment and context. Other frameworks which more tightly focus on localization and sensor fusion are the *location stack* ([HBB02]) and its successor, the *PlaceLab* (see for example [HLS06]). In [Hub08] the extension of a framework for personal context sensitive processes by a combined location sensing technology is presented. An overview of the large number of different approaches can also be gained from [HB01a], [HB01b], which also introduces a taxonomy for localization approaches, or from [KH06]. A different survey on frameworks for ubiquitous computing, which also has a pronounced focus on AR related systems is [EBM05].

Similar to the number of different frameworks, various different localization technologies have been proposed. Some of the approaches, not otherwise discussed in this document include GPS pseudolites (see [Cob97] or [NKS+08]), Bluetooth-based infrastructure (see [OBSR08]) or even the signal strength of regular FM radio broadcast stations [KCH03]. Indoor localization using GSM base station infrastructure also seems to be especially promising (see for example [OVLDL05] and [VCL+06, VLH+07]).

3.3.1 Overview of different approaches

Of further interest in the context of parasitic tracking are especially the individual localization methods, as parasitic tracking methods have to rely on multiple approaches simultaneously. In the following a number of different approaches are summarized and the decision to mostly employ range-based localization for the further experiments is discussed.

The fundamental setting of these approaches can be described as follows: Based on observations of the user's environment the location of the user is to be estimated. These observations consist of detected anchors (sometimes called beacons) in the user's vicinity and each observation can, depending on the concrete technology, either be associated with a quality value or it can be a binary indication only. The anchors themselves are assumed to be fixed in the environment, to be distinguishable and individually identifiable. It is furthermore assumed that their observed behavior does not change over time. A further common assumption, which some approaches require, is that the locations of the anchors are actually known.

Furthermore note that the relationship of the observation between anchor and user may also be inverted. In some cases the anchors are actually sensors which sense signals transmitted by the user (for example [WHFG92]). The localization is performed on the observations of the user from multiple sensors. While these approaches do not fit the parasitic tracking idea (since instrumentation of the environment with sensors is

required), the localization methods are still comparable to other cases.

Based on these assumptions, most localization approaches can be classified as one of the following fundamental methods:

Centroid methods The most straightforward way to derive a position estimate from multiple observed anchors with known locations is the centroid method. In its basic form the location of the user is computed as the mean of the locations of the anchors which are currently detected. Thus the user is estimated to be in the center amongst the observed anchors. Note that locations estimated by centroid computation always lie inside or at most on the hull of the volume spanned by the locations of the anchors.

A direct improvement of this method is the *weighted centroid method*, which considers a further weighting of the individual anchor locations, usually depending on some estimate of distance or "nearness" of the respective anchors. This includes approaches based on signal strength (for example [BRT05]) or likelihood estimation of the weights (for example [NLLP04]).

A special case is *Point based localization* which adopts the location of the beacon as the user's position in the case that only a single beacon is observed by the user. This is also related to *cell based localization* in a cell-oriented network where the current cell in which the user resides is identified as the user's location.

Fingerprinting A different class of approaches which receives much attention are the *fingerprint* or *snapshot* based localization schemes. These methods have in common that the environment in which the user will be located afterwards, has to be sampled in a preprocessing step. This produces a set of reference measurements at known locations (either absolute in the environment or at least relative to each other), which are used as a (high-dimensional) "map" of the environment. From this map the current location of the user can be deduced. This is similar to a machine-learning approach, in that the tracking system has no underlying model of the environment or the behavior of the anchors, but learns how to determine the user's position based on the reference set. In most of these approaches even the locations of the anchors themselves are irrelevant. On the other hand, both the position and orientation of the user may be relevant for the reference measurements, which increases the dimensionality of the map.

An interesting property of this approach in the context of parasitic tracking is that it can operate with a minimal number of assumptions about the environment. The only requirements are that the anchors are individually identifiable and that their behavior stays constant over time. On the other hand no assumptions on the sensing model (for example propagation model) or the location of the anchors are made. While this property seems desirable for parasitic tracking approaches, the

required setup phase of the working volume seems unsuitable for use in ubiquitous tracking scenarios.

Fingerprinting methods can further be differentiated by the required density of reference data and the way the location estimate is obtained from the reference data.

K-Nearest Neighbors A common approach is to compute the Euclidian distance between the current observation and all reference data, thus ordering the reference points by distance of their corresponding observations. The k nearest references (thus k-nearest neighbors) are selected and the position estimate is computed as the mean of the corresponding reference positions. Note that these are the positions where reference measurements were taken and which in general are not the locations of the anchors. Thus this approach is not contained in the class of centroid algorithms.

Examples where this method was used include [CSC+06], [OVLDL05], [SVL+06], [VLH+07], [WLS+07] or [Zho06].

An interesting variation of this approach was also used by [BP00] by combining k-nearest neighbors localization with pre-computed (rather than empirically determined) reference data based on a radio propagation model.

Probabilistic positioning Instead of directly operating on the reference measurements and comparing them with the currently observed values, the class of *probabilistic positioning methods* extracts an empirical model of the conditional probability distribution of $Pr[O|L]$ of observing O at location L. By assuming a prior distribution of $Pr[L]$ and by using Bayes' rule the distribution of $Pr[L|O]$ can be derived as

$$Pr[L|O] = \frac{Pr[O|L] \cdot Pr[L]}{Pr[Z]} = \frac{Pr[O|L] \cdot Pr[L]}{\sum_{L_i \in \mathcal{L}} Pr[L_i] \cdot Pr[O|L_i]},$$

where $Pr[Z]$ is defined by summing over all possible locations $\mathcal{L}$. Further note that $Pr[Z]$ does not depend on L and thus can be seen as a scaling factor for $Pr[L|O]$. More specifically, when solving for extrema of $Pr[L|O]$, the factor $Pr[Z]$ can be neglected, provided that $Pr[Z]$ exists under the individual circumstances.

Using the distribution of $Pr[L|O]$ the location estimate of the user can be derived in a maximum likelihood fashion, by determining the most likely location L for a given observation O, thus maximizing $Pr[L|O]$.

This approach has several advantages compared to the k-nearest neighbor approach. By incorporating contradicting or uncertain reference measurements for a single location into a probability function, the approach handles noisy

or error-prone sensor measurements better. Another important aspect is that the required density of the reference measurements is reduced and thus a similar level of accuracy can be achieved using less reference data and thus requires less setup-time. Furthermore the probability distribution can include an informed prior distribution of the user's position. This enables tracking incorporating a motion model of the user or the exclusion of highly improbable areas, such as obstacles. If no such information is available, a simple uniform distribution can be used as an uninformed prior.

This approach was introduced by [RMT+02b, RMT02a] and applications include [EM06], [HBF+04], [SVZ07] or [Zho06].

The most common approach to empirically derive the conditional distribution of $Pr[O|L]$ from the reference data is to compute the quotient of the number that any individual observation was made over the total number of observations made at a specific location. Still, more advanced models may also be used such as for example *Gaussian Process modeling* which models the measured value of each anchor and at each location as a Gaussian process (see [SGTH04] and [CSC+06]).

Another interesting method is used by [KPV10, Kus08], who incorporated the probabilistic positioning in a general position filter (in the concrete case a "nonparametric information-filter").

Range based Another classical method to estimate the position using observations of anchors with known locations is to first estimate the individual distances (or ranges) to each anchor. Using these ranges and the locations of the anchors, the position of the user can be estimated, for example using *trilateration*.

[JZU+10] describes such a method. It takes cheating anchors into consideration. Further examples of similar localization schemes include [HWB00] who uses trilateration based on an empirical signal strength model for custom hardware and subsequent localization by signal strength minimization or [LCC+05] who estimates ranges to anchors and formulates the localization as an associated Euclidian graph embedding problem such that the graph edge lengths best fit to the estimated distances. The cricket system (see [PCB00]) also uses range estimates to various fixed nodes to determine the estimated location of the user. In this system, the distance is estimated by comparing the time-of-flight of a radio and an acoustic, ultrasound wave which are simultaneously transmitted. A detailed analysis of such a localization scheme for custom hardware used in a wireless sensor network can be found in [WKC07].

The localization methods used in the later chapters mostly belong to this class of range based approaches.

Range-free localization In the area of Wireless Sensor Networks (WSN) a number of *range-free localization* methods have emerged. Here no direct sensing of the anchors is performed, instead the only way of communication for each node is passing messages via the sensor network. The nodes representing the anchors know their respective location and usually broadcast this at regular intervals into the network. The location of the remaining nodes is then deduced using connectivity properties of the network.

For example an approximation of the distance to an anchor node can be derived by counting the number of hops the message required to reach a certain node. There are also more advanced schemes, such as determining triangles of triplets of anchors and locating a node via intersections of such triangles([HHB+03]).

Symbolic localization As a final class of localization methods *symbolic localization* methods should be mentioned. Instead of computing a Cartesian position or area of probable presence, these methods derive a symbolic description of the user's current location, such as distinguishing rooms. The aforementioned *cell based localization* can be interpreted as a symbolic localization method. Such information can be sufficient for a number of location-aware applications.

Semantically relevant places can also be automatically discovered from traces of sensor observations([KHGE09]).

3.3.2 Localization methods used in this thesis

The localization method used in the following chapters is based on range estimation and determines likely areas of presence. The reason this method was chosen as the first method to investigate is based on two points.

First, because of the relative ease of implementation, this approach was more readily available and promised to produce immediate results. Furthermore the transparency of the approach made the individual performance and problems in each case more accessible.

Second, these localization methods stress the approach to directly exploit the physical parameters of the observed infrastructure and to directly gain meaningful information, as opposed to machine-learning methods. These physical parameters are expected to be better manageable, especially in multi-sensor setups and to offer better insight on how to manage contradicting observations or how to handle evolving environmental conditions (such as addition, removal or relocation of a Wi-Fi beacon). The parameters describing the exploitable infrastructure are smaller and thus better distributable between different sensing methods or different participants. Similarly the setup time is minimal for these range based approaches, since only the locations of the anchors are required to be known.

In summary, trilateration methods based on distance estimates were most easily available and also seemed most fitting to the parasitic tracking approach. Nevertheless future

work should explore the applicability of more advanced localization procedures. Especially probabilistic methods promise great potential to provide both sufficient accuracy and improved robustness against contradicting or uncertain observations, albeit in a different sense.

Furthermore note that motion estimation or filtering of the localization results and fusion with other sensors does not have to be integrated into each of the localization methods itself, but can be achieved on a higher level, as will be explained in the next chapter.

4 Ubiquitous Tracking

The parasitic tracking approach is based on the general notion of ubiquitous tracking, and in particular, the "Ubitrack" concept and system at the "Fachgebiet Augmented Reality" at TU München. Establishing Ubitrack, under the considerations of the requirements of parasitic tracking, has been a major part of this thesis. This chapter provides an overview.

The work described in this chapter was funded by the Trackframe[1] project and was conducted in close cooperation with Daniel Pustka, Peter Keitler, Michael Schlegel and Florian Echter. Parts of the description below have already been published in the papers [PHBK06] and [HPK+07] presented at ISMAR 2006 and ISMAR 2007 respectively.

4.1 Ubiquitous Tracking - Abstract notation and reasoning

In the previous chapter a short overview of the different components required in a ubiquitous augmented reality application was given. The determination of the spatial parameters of the user and the surrounding environment, which is also called the tracking task, is one of the most challenging tasks involved. Concerning ubiquitous augmented reality, this component, at this point in time, is also the one to which most attention has been paid. This development appears to be understandable, as especially ubiquitous interaction requires sufficient progress in the other areas to be viable for scientific investigation.

To enable systematic and rigorous examination and development of ubiquitous tracking methods, a suitable research framework was developed. This framework, called the "Ubitrack library" or just "Ubitrack", serves both as a platform for ubiquitous tracking research and as a deployment platform for augmented reality applications. Apart from the primary use as a ubiquitous augmented reality research platform this library has been used so far in a number of industrial augmented reality applications and has facilitated interesting research in the general field of tracking for augmented reality.

In the remainder of this chapter the core ideas underlying the framework will be discussed and the resulting architecture will be introduced. This framework is the basic foundation of the further localization techniques presented in the later parts.

Ubiquitous tracking situations share some common attributes which make them interesting or even difficult tracking scenarios. First, ubiquitous tracking scenarios in general

[1] `http://trackframe.de`

include wide area trackers with unknown or anonymous bodies. The tracking setup can also be assumed to be highly heterogeneous and to incorporate many different tracking technologies. On the other hand UAR applications should be able to operate without knowledge of the concrete tracking situation. This is required to guarantee smooth transition throughout a UAR environment, as different sensors become available or go away. Usually the user's main concern is to find a method to compute the tracking data the user is interested in from the data the various tracking sensors deliver. The geometric arrangement of the sensors and objects can be seen as the fundamental underlying structure of the problem, and the spatial relationships the user is interested in as the actual question to be answered.

Thus in general, UAR tracking situations are highly dynamic and heterogeneous and thus require dynamic runtime adaptation and dynamic combination of sensors (sensor fusion). These goals should be achieved while minimizing user interaction with the tracking system, rendering the tracking situation effectively invisible for the user.

To attain these goals, a suitable abstraction and formulation has to be found which has the ability to capture the essential attributes of the current tracking situation and enable computers to automatically adjust to it.

4.2 Related Work

In the tracking for AR and VR field, there are already other software and conceptual frameworks, aimed at various stages of the tracking problem. Two of the most important tracking frameworks in this context are the OpenTracker [RS01] system and the VRPN [THS+01] system. These systems however mostly target the actual processing of the sensor data and in general do not focus on the automatic derivation of tracking solutions or the dynamic configuration or reconfiguration of tracking setups.

Another well known concept for description of spatial setups is the scene graph ([SC92]). Some approaches for tracking solutions based on this abstraction are for example [CMJ04] or [SSFG97], although the rigid tree-structure imposed by the scene graph is too restrictive for general multi-sensor setups.

The here presented Ubitrack library is based on the previous work and experience of the various Ubiquitous Augmented Reality endeavors so far. In [NWB04] a Ubitrack system was presented, which was able to automatically derive pose information from different tracking sources and transparently present them to the user. A drawback of this approach was the reduced performance of the system, especially in larger setups. This system was based on the *Dwarf* framework, and thus improvements in the performance of such a system were among the main motivations for this research. The performance of the automatically configured system should be no worse than that of a hand-tailored approach, with focus on low latency data processing and distribution and quick reaction and reconfiguration times.

The aforementioned Dwarf framework, which is the conceptual predecessor of the here presented system, was presented in [BBK+01], [MSW+03], [Wag04], [WHK04], [Wag05] and [Mac05]. While the spatial relationship graph concept, on which the "Ubitrack" system is heavily based, was already introduced by [NWP+03] and [NWB04], the exact notion which is employed in the remainder of this document differs in some significant aspects.

Dynamic fusion of gyroscope data based on the "Ubitrack" system was presented in [PK08],[PHK08]. An interesting application of the spatial relationship graph concept to graphics rendering instead of tracking data is discussed in [EHP+08].

As part of the Ubitrack approach, also a graphical management tool for spatial relationship graphs and data flow networks was developed. An overview of this tool and its advantages can be found in [KPH+10].

4.3 Spatial Relations and Spatial Relationship Graphs

At its most fundamental level, any tracking situation can be described as a set of local coordinate frames or points in space associated with real or virtual objects together with the relevant relationships between these entities. Such a setup can be represented as a graph where a node is associated with each of these entities and the edges describe either tracked or inferred transformations between two nodes. We call this graph is called the "Spatial Relationship Graph" (SRG). See figure 4.2b for an example.

In the most common case, the two entities forming a spatial relationship are complete coordinate frames, and the corresponding spatial relationship is the respective coordinate frame transformation consisting of a translation and rotation. These are generally called *poses* or 6 degrees-of-freedom (6DoF) measurements. In more rare cases, where points in either 3- or 2-space are involved (such as positional sensors with subsets of these parameters, inertial orientation sensors or points on an image-plane), different types of relationships may be required.

Any such relationship is represented as a directed edge in the graph. To be able to completely capture all relevant aspects of a tracking setup further attributes have to be associated with each edge.

4.3.1 Edge annotations

Generally there are various kinds of edge attributes and the usage of some attributes may be situation specific. Nevertheless the following attributes are the most common ones and are always present. Also note that edge attributes may have arbitrary codomain. While most attributes will be binary, different kinds of annotations require different associated data types.

Data type The most important attribute is the aforementioned data type of the relationship. Usual types encountered in usual tracking or registration scenarios include 6DoF coordinate transformations, 3D positions, 3DoF orientations or 2D positions. Projective transformations such as 3D $\rightarrow$ 2D used for image plane projections are also common. But also more exotic types (such as a 3DoF pose on a surface for example) are possible. In diagrams the data type is usually annotated to the graphical arrow representing the directed edge.

Static/Dynamic Another important annotation specifies whether the relationship is assumed to be static, that is constant in the scope of the SRG, or dynamic. Static edges include fixed or rigid relationships between two objects such as offsets or rigid connections. Such edges are usually determined by registration. Dynamic edges, on the other hand, are flexible relationships which can and usually do change at runtime. Such edges either have to be tracked or have to be inferred from sensor data. Many tracking algorithms utilize the fact that some part of the setup cannot change to derive information on the dynamic parts. In diagrams static edges are further "static" whereas dynamic edges have no extra label.

Measured/Inferred The third core attribute of an edge is whether the transformation described by the relationship is directly observable by the technology used in the concrete tracking setup. Any transformation which is known a priori or can be explicitly sensed by some device is called measured. This includes pre-registered static edges. Inferred edges on the other hand have to be computed from the sensor data of measured edges. The directly measured edges form the basic, underlying spatial relationship graph to which inferred edges can be added using tracking algorithms. Note that the addition of inferred edges does not add new information to the tracking setup.

Generally the number of edge associations is not limited and new kinds may be introduced at any time. The later discussions for example will also require edge attributes "push/pull", "synchronization domain" or "information sources" for more technical reasons.

4.3.2 Spatial Relationship Graph formalism

Formally, a spatial relationship graph can be defined as follows. Let the SRG $G = (V, E)$ be a multigraph over the set of nodes V of all entities relevant to the tracking setup with edge set $E \subseteq V \times V$ of directed spatial relationships. Note that E may be a multiset since multiple edges between two nodes are allowed, as well as loops.

Let a function $A : E \mapsto X$ be an edge annotation with codomain X which maps each edge $e \in E$ to its associated value $A(e) \in X$. Concrete instances include $A_{\text{type}} : E \mapsto \{\text{6DoF, 3D Position, 3DoF Orientation, 2D Position}, \ldots\}$, $A_{\text{static}} : E \mapsto \{\text{static, dynamic}\}$ or $A_{\text{inferred}} : E \mapsto \{\text{measured, inferred}\}$. We call the family of such

functions $\mathcal{A} = \{A_{\text{type}}, A_{\text{static}}, A_{\text{inferred}}, \ldots\}$ the annotations on the edge set of G or more informally the annotations on G.

4.4 Patterns for Tracking and Registration

To describe the transformations used to introduce new inferred edges into a basic SRG, the notion of spatial relationship patterns will be introduced. Like the introduction of spatial relationship graphs enabled the formal discussion about tracking setups, this abstract concept extends these discussions to include tracking algorithms.

A spatial relationship pattern is used to identify parts of the overall spatial relationship graph for which a known algorithm exists. Associated to each pattern is a set of algorithms in the from of data flow components, which, based on certain input edges, compute a result which again can be inserted back into the graph. This procedure can be applied recursively until a solution for a requested edge is found.

In an abstract sense such patterns define the 'signature' of an algorithm to solve a specific problem in a tracking setup. The input of these algorithms is a set of measurements as defined by the problem and they return a set of measurements that is part of the solution to the overall problem. Contrary to function signatures in programming languages, spatial relationship patterns do not merely define the type of arguments and return values, but also impose restrictions on the geometric relationship between them.

The graphical representation of spatial relationship patterns derives naturally from the representations of spatial relationship graphs, as patterns are in fact sub-graphs of an SRG. However, there are two types of edges in a pattern: The *input edges* are the edges that are required in the spatial relationship graph before a pattern can be applied. These edges represent the required inputs of the tracking algorithm. Analogous, the second type of edges represent the computed outputs of the algorithm and are called the *output edges*. This distinction is so fundamental that rather than using another edge attribute, this fact is signified by a partition of the edge set.

4.4.1 Formalism

Similar to the formal definition of the SRG, a formal definition of spatial relationship patterns can be derived.

Formal definition A spatial relationship pattern P consists of five parts:

- a graph $G_P = (V_P, E_P)$ describing the basic spatial relationship represented by the pattern,
- an edge partition $E_P = I_P \dot{\cup} O_P$ which designates each edge either as input or output,

- a family $\mathcal{A}_P$ of further edge attributes which either have to be met or which describe the properties of the output edges,
- a set $\mathcal{C}_P$ of correspondences between certain parts of the input. This will be explained later.
- a set $\mathcal{A}_P$ of algorithms implementing the specific pattern.

Graph Part of the pattern P is the graph $G_P = (V_P, E_P)$ on the set of nodes V_P which are virtual objects or roles (such as camera, object or fiducial) and which eventually have to be assigned to nodes in the spatial relationship graph. On these nodes the edge set $E_P \subseteq V_P \times V_P$ of directed edges is defined. These represent involved spatial relationships on which the algorithms operate.

Furthermore these edges are partioned into distinct sets $E_P = I_P \dot{\cup} O_P$. We call the set I_P the set of input edges and O_P the set of output edges. The input edges represent the data necessary as input to the algorithm in order to produce the output edges. When applying a pattern to a spatial relationship graph the input edges have to be present while the output edges are inferred by the pattern.

In the graphical representation we draw input edges I_P as normal and output edges O_P as dashed arrows.

Edge attributes Similar to the edges of a spatial relationship graph, the edges in pattern are attributed by a familiy $\mathcal{A}_\mathcal{P}$ of functions $A_P \in \mathcal{A}_\mathcal{P} : E \mapsto X$ on the edge set of the pattern graph. The semantics of this mapping differs severely for input and output edges. For input edges these attributes represent requirements on the attributes of the edges in the spatial relationship graph. In order for an edge in the SRG to be suitable as an input for a specific pattern all attributes on the edges in the pattern have to agree with the corresponding attributes on the edges in the SRG.

For output edges the semantics is similar to the edge attributes of an SRG. The attributes are simply the attributes of the newly inferred edges which are to be added to the SRG.

Note that every edge contains the partial mapping $A_{P,inferred}(e \in O_P) =$ inferred, which designates all output edges as inferred. Also note that without loss of generality the reverse, i.e. $A_{P,inferred}(e \in I_P) =$ measured is not true, since an input edge may either be directly measured or may also be inferred by a different algorithm.

In the graphical representation edge attributes are represented as edge labels, just as for edges in the spatial relationship graph.

4.4.2 Application of a pattern on an SRG

Given a spatial relationship graph $G = (V, E)$ and a specific pattern P, an instance of this pattern is located in the SRG by an edge matching $\alpha : E_P \mapsto E \cup E^+$ with

$E^+ \subseteq V \times V$ which assigns every input edge in I_P of the pattern to a suitable edge contained in the SRG and every output edge in O_P of the pattern to an edge in a set E^+ of derived edges which may or may not be already present in G.

Such an instance is denoted by $G[\alpha(P)]$, its node set by $V[\alpha(V_P)] \subseteq V$ and its edge set by $E[\alpha(E_P)] \subseteq E \cup E^+$.

For an instance $G[\alpha(P)]$ it can finally be decided whether to add the output edges to the SRG or not. This is called an application of the pattern instance $G[\alpha(P)]$ on G and results in a new spatial relationship graph $G' = (V, E')$ with $E' = E \cup E^+$.

4.4.3 Correspondences

Many tracking algorithms rely on the existence of corresponding measurements from two or more frames of reference, in order to calculate the requested output.

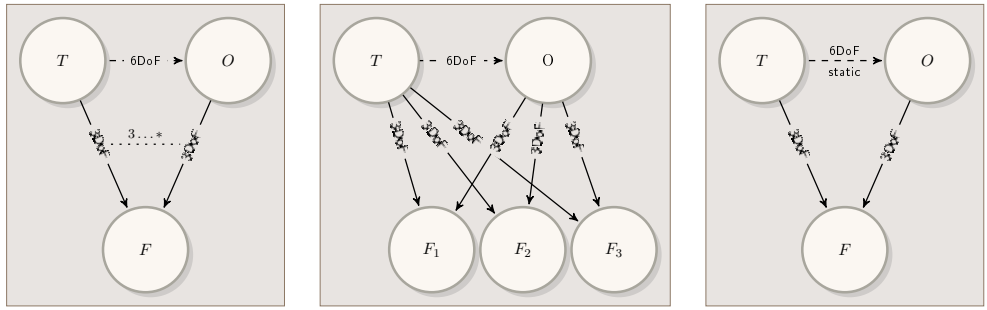
One common example is Horn's ([Hor87]) method to determine the transformation between two different coodinate systems, also known as the 3D-3D pose estimation or absolute orientation problem. Based on the measurement of a common set of at least three points from both references, a 6DoF pose can be computed, which transforms one coordinate frame into the other. The requirement of the algorithm is such that for a set of three points in both the coordinates in the first and the corresponding coordinates in the second coordinate frame are known.

This is a general observation which is applicable to many tracking algorithms and has led to the development and inclusion of the edge correspondence concept for spatial relationship patterns.

Also note that in the example above, the three measurements could be acquired simultaneously by mapping the correspondences to three distinct corresponding spatial relationship pairs in an SRG. Alternatively the measurements could be acquired sequentially by observing a single and dynamic corresponding spatial relationship pair at three different times where the pair assumes three different measurements.

It is also a general observation that for most algorithms edge correspondences can either be resolved in parallel or resolved serialized. The first concept is called expansion of the correspondence in space and whereas the latter is called expansion in time. See figure 4.1 for an example of an space and time expansion of the absolute orientation pattern.

Formal definition Every time an algorithm of pattern P needs to relate k different measurements in potentially different coordinate systems, the corresponding edges $e_i \in I_P$ for $1 \leq i \leq k$ are identified as a correspondence set C_P. Furthermore, with each correspondence set C_P a set of integers M_P of acceptable numbers of corresponding measurements needed by the algorithm is associated. Most of the time this set will be characterized by some lower bound, but it is also thinkable that some algorithm only



(a) Pattern notation (b) Expansion in space (c) Expansion in time

Figure 4.1: Expanding the 3D-3D pose estimation pattern in time and in space

operates for example on multiples of four measurements. Finally the correspondence of a pattern is defined as $\mathcal{C}_P = (C_P, M_P)$.

Note that these measurements in general have to be distinct from each other and have to supply significantly different data. Furthermore there may be additional restrictions which are not directly representable by a spatial relationship graph or pattern, such as non-coplanarity of measured points. These restrictions are beyond the scope of the pattern formalism, since they apply to the semanic data of the measurements and not to the geometric setup of the tracking environment.

While patterns with two corresponding edges ($k = 2$) are the most common case, it is also possible to have one or more than two edges in one correspondence. Thus correspondences form generalized undirected hyperedges on the node set of the pattern. Also note that an easy generalization to more than one correspondence per pattern is possible.

Notation In the graphical notation, the correspondence sets C_P are denoted by dotted lines connecting all edges $e_i \in C_P$. Furthermore these lines are annotated with restrictions on the number of acceptable measurements M_P. A label "3...*" for example signifies that at least three measurements are required with no upper limit or $M_P = \{3, 4, \ldots\}$ in this case. An example for this notation is given in figure 4.1a.

In order for a subgraph of a spatial relationship graph to fulfill these requirements and thus to contain a pattern, it is necessary to find an expansion of the correspondence of the pattern, leading to an expanded version $\bar{P}$ of pattern P. As mentioned above, there are two fundamental possibilities for expansion: expansion in space and expansion in time.

Expansion in space For a space expansion of a correspondence set C_P with restrictions M_P in a spatial relationship pattern P all edges $e_i \in C_P$ contained in the correspondence

set are replaced in $\bar{P}$ by distinct instances $e_{i,1}, \ldots, e_{i,l}$ of these edges for some acceptable multiplicity $l \in M_P$. The different edges thus represent similar, but may not represent identical spatial relationships.

For example, for the spatial expansion of the 3D-3D pose estimation example from above, it would be necessary to have at least three copies of the node representing the point F which is measured in both coordinate systems. The space expansion of this pattern is depicted in figure 4.1b. Spatial expansions of a pattern calculate the output edges of the pattern for each time step in which the input edges are measured. Thus spatial expansion may have both static or dynamic output edges.

Expansion in time The other possibility to expand a correspondence (C_P, M_P) is in time. This way the edges $e_i \in C_P$ remain single edges in the expansion $\bar{P}$ but have to be measured at different times during the running time of the application. Furthermore all edges $f_j \in E_P \setminus C_P$ not contained in the correspondence in general have to be static, while the edges $e_i \in C_P$ must not be static and have to display m distinct values for an $m \in M_P$ during the running time of the system.

This is the case for example when calibrating a static relationship by sequentially placing a calibration tool at different points in space and making measurements each time. See figure 4.1c for an example of the time expanded 3D-3D pose estimation pattern. It should be noted that expansions in time are only applicable if all output relationships are static.

Application Note that a pattern P using correspondences is never directly applied to an SR graph, but rather in one of the two expanded forms. We denote the expanded form of a pattern P as $\bar{P}$. Such an expanded pattern $\bar{P}$ can be applied as outlined above.

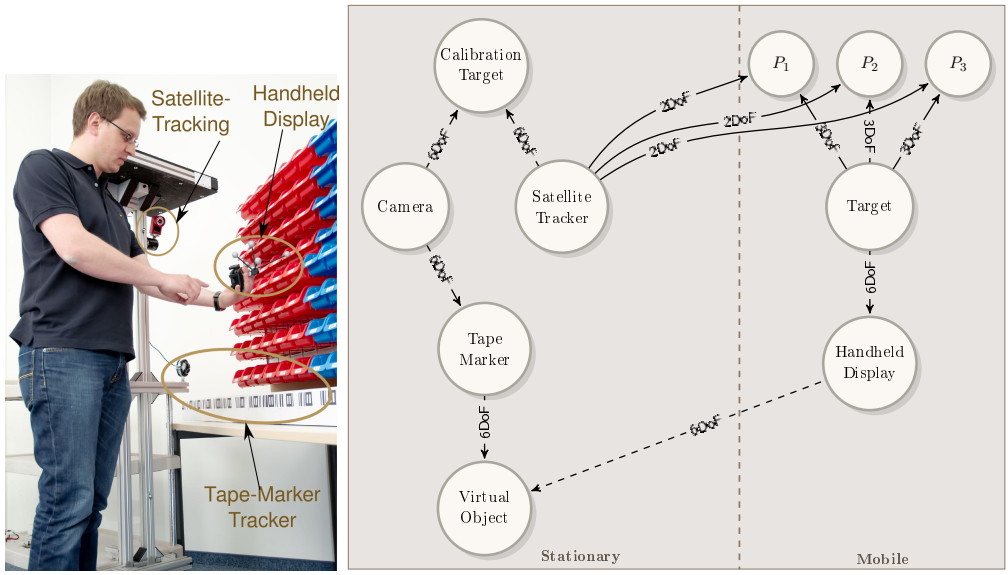
4.5 Example

The following example demonstrates the use of spatial relationship graphs to model a complex, heterogeneous tracking setup and to derive a correct solution to a user query.

The example is motivated by a logistics and picking scenario derived from the recent AVILUS[2] (BMBF) and FORLOG[3] (BFS) projects, which involved guiding a storage worker to the next picking task and to help confirm the correctness of the picked item. The setup of this scenario consists of an industrial picking cart, which can be maneuvered in the narrow corridors of a warehouse. This cart simultaneously serves as the support for various kinds of wide-area and relative tracking systems, as well as different augmentations methods. The applicability of these technologies was evaluated in different prototypical setups. Figure 4.2a shows one of these prototypes, consisting of

[2] `http://www.avilus.de`
[3] `http://www.forlog.de/`

a tape-marker wide-area tracking system, a handheld display for augmentation and a satellite tracking systems for determining the pose of the handheld relative to the cart. The corresponding spatial relationship graph which describes this tracking setup can be seen in figure 4.2b.



(a) AVILUS picking cart (b) Spatial relationship graph

Figure 4.2: Example scenario with tape-marker tracker and handheld visualization.

All relevant parts of the tracking setup are represented in the spatial relationship graph.

Handheld display The mobile display is a handheld smartphone mainly running the visualization needed to indicate the next picking task. Tracking tasks of low complexity can be performed on the device.

Handheld display marker target A marker target consisting of retroreflective marker balls (serving as fiducials) is mounted to the mobile display. The coordinate frame of this marker target is fixed, and the 3D positions of each ball relative to this coordinate frame are known. These two parts, the handheld device and its marker, form the handheld visualization satellite of the system.

Satellite tracking camera A tracking camera with infrared flash is mounted on the picking platform. The flash illuminates the retroreflective balls of the handheld target such that the marker is easily discernible in the camera image. Using this image the $6DoF$ pose of the target can be computed using monocular 2D-3D pose estimation.

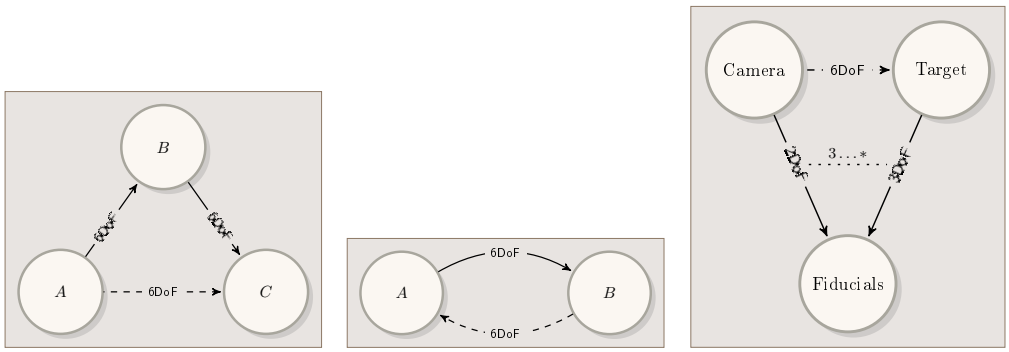
Tape-marker tracker For localization of the picking cart inside the warehouse environment, each of the shelf corridors is instrumented with part of an endless marker tape. The tape contains sections which enable full 6DoF pose tracking at regular intervals and further encodes the current position along the tape as a machine-readable barcode. This tape is tracked by a dedicated camera, which is suitably mounted on the picking cart.

Calibration target To compute the offset and the relative orientation of the two cart-mounted cameras, a special calibration target has been inserted into the viewing area of both cameras. The calibration target is designed to be detectable by both tracking systems, so that both poses can be computed simultaneously. This target is not actually tracked or even present during the regular runtime of the system, but rather previously recorded corresponding poses are loaded and replayed.

Virtual object Fixed in the room a virtual object is positioned at the shelf where the next item should be picked. The coordinates of this virtual object are given relative to the tape marker.

The spatial relationship necessary to display the augmentation of the virtual object is the 6DoF pose between the handheld display and the virtual object. This relationship is indicated by the dashed edge in figure 4.2b.

In order to derive this edge from the given tracked or static relationships, a number of spatial relationship patterns are necessary. Figure 4.3 shows three common patterns: the pose concatenation or multiplication, the pose inversion and the 2D-3D pose estimation.



(a) Pose multiplication (b) Pose inversion (c) 2D-3D-Pose estimation

Figure 4.3: Basic spatial relationship patterns

Using these patterns, it is possible to derive additional edges in the SRG and eventually satisfy the query of the handheld device. The following steps are needed to derive this edge. The derived SRG with all intermediate steps is shown in figure 4.4.

1. The static calibration between mobile display and mobile marker target coordinate frame is inverted.
2. Using 3D-2D-pose estimation the $6DoF$ pose of the mobile marker target is computed from the 2D positions of the marker balls in the satellite tracking camera and the known, corresponding 3D positions in the target coordinate frame.
3. This computed pose is inverted.
4. By multiplying the inverted calibration by the inverted tracked relationship, the transformation from the mobile display into the coordinate frame of the satellite tracking camera can be computed. By performing this multiplication via patterns, the correct multiplication order is automatically ensured.
5. The transformation from the tape-marker camera to the calibration target is inverted.
6. Multiplying the spatial transformations from the satellite tracking camera to the calibration target and continuing to the tape-marker camera, the calibration between the two cameras can be computed.
7. Further multiplying the transformation from the mobile display to the satellite tracking camera with the calibration computed in the previous step results in the relationship from the mobile display to the tape-tracking camera.
8. This transformation can furthermore be multiplied by the tracked relationship to the marker tape.
9. As the virtual object is defined in the coordinate frame of the marker tape, the transformation originating from the handheld device is multiplied by the transformation into the coordinate frame of the virtual object. This transformation is finally the relationship which has originally been requested.

4.6 Automatic Pattern Detection in Spatial Relationship Graphs

There are some advantages in using the spatial relationship graph and pattern abstractions to describe tracking setups. On the one hand an SRG description can serve as an essential tool to systematically explore a concrete tracking task, communicate different approaches with collaborators and accurately document solutions. A different benefit from this abstraction, which is of even greater importance to the application to UAR, is the possibility to capture the essence of the tracking setup in a computer-readable

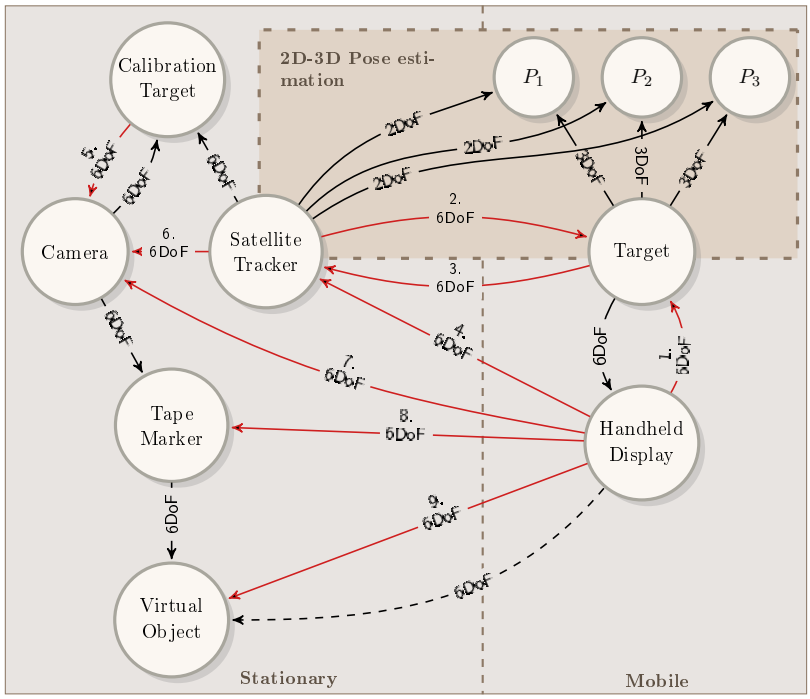


Figure 4.4: Example scenario SRG with patterns applied

manner. This particularly enables automatic reasoning about tracking setups, resulting in an adaptive system and full- or semi-automatic SRG and dataflow generation. Note that especially in the context of ubiquitous augmented reality this is a fundamental requirement, since in such a context no handcrafted tracking solution can be applied. Due to the inherent heterogeneity and complexity of the tracking environment, adaption and dynamic derivation of tracking solutions are necessary.

Thus it is vital to have some means to automatically process spatial relationship graphs and patterns. The approach to automatically derive spatial relationship graphs through pattern application can be split into two distinct tasks:

Single pattern detection Given a spatial relationship graph G and a pattern P, systematically apply the pattern to the graph as described above. This step first locates all instances $G[\alpha_i(\bar{P})]$ of a suitable expansion $\bar{P}$ of the selected pattern P as subgraphs of the SRG. This is related to the well known subgraph isomorphism problem (see for example [Ull76]). Formally, given graphs $G = (V_1, E_1)$ and $H = (V_2, E_2)$, the problem is: Determine whether G contains a subgraph that is isomorphic to H. Unfortunately this problem is well known to be NP-complete[GJ79] and thus is unlikely to have an efficient solution in the general case. On the other hand, in this special case there are

some factors which work in our favor, which will be discussed later.

Control mechanism In order to reach a specific goal as specified by the requirements of the user, a control mechanism is necessary that determines a suitable sequence of patterns $(P_1, P_2, \ldots, P_n)$ to apply to the SRG in question in order to reach the desired goal. For this, the controller could exhibit some kind of reasoning about rules associated with the patterns. The current implementation uses a simpler strategy, which effectively performs an explorative search over the tree formed by all possible pattern combinations.

As a first step, this approach assumes that a spatial relationship graph and the pattern to apply are given in its correspondence notation as earlier described.

The following algorithm implements the first step of this strategy. Given a graph G and a pattern P, determine the application of P on G. Note that instead of first choosing an expansion $\bar{P}$ and locating this expansion in G, all unexpanded copies of P in G are located and unified to form expanded pattern instances. The basic steps of the algorithm are:

1. Find all occurrences $G[\alpha'_i(P)]$ of the unexpanded pattern P in the graph G using a subgraph isomorphism algorithm.

2. Find inclusion maximal sets of patterns that represent expansions of the pattern and respect all relevant constraints.

3. If the pattern satisfies the correspondence requirements, we have located an instance $G[\alpha_i(\bar{P})]$ of an expansion $\bar{P}$ of P.

4. If the inferred edge contributes towards the overall strategy as determined by the controller, it is inserted into the graph.

4.6.1 Unifying pattern instances

In this first step, the algorithm determines a list $(G[\alpha'_1(P)], \ldots, G[\alpha'_k(P))$ of all instances of the unexpanded pattern P. In order to successfully detect suitable expansions of the pattern compatible instances need to be collected and unified so that a resulting instance $G[\alpha_i(\bar{P})]$ contains as many edges as possible which contribute to the correspondence (C_P, M_P) of P. Two non-expanded pattern instances $G[\alpha'_1(P)]$ and $G[\alpha'_2(P)]$ are compatible if all edges of P not contained in the correspondence set C_P are mapped to equal edges in G.

An important additional constraint is that for any expansion of a pattern, a set of edges contributes to the correspondence only if the information expressed by these edges is not already contained in the collection of the pattern so far. Denote such a located collection by $G[\alpha_i(\bar{P})]$ and check if it actually represents an instance of a time or space expansion of P.

If every edge $e_j \in C_P$ in the correspondence set of P is represented by m edges in the collected instance and $m \in M_P$ is an acceptable number of measurements, then an instance $G[\alpha_i(\bar{P})]$ of a space expansion of P has been found. If, on the other hand, every edge $e_j \in C_P$ in the correspondence set of P is represented by a single edge in the collected instance and all edges in $E_P \setminus C_P$ are static as well as all edges in C_P are dynamic, an instance $G[\alpha_i(\bar{P})]$ of a time expansion of P has been found. In any other case, the collected instance as an incorrect expansion of P will be rejected. This step finally constructs a list of all expanded instances of P found in G.

4.6.2 When to add new edges

After an instance of an expanded spatial relationship pattern $\bar{P}$ was successfully located in the SRG G, its output edges O_P may be inserted into the SRG.

Here another problem arises, as the unconditional insertion of all derivable edges immediately leads to uninteresting results and perpetual insertion of similar edges. As an obvious example, an application of the inversion pattern produces a new edge which can of course be inverted again. This evidently adds no new information to the spatial relationship graph, and with each inversion the accuracy of the edge is reduced which is an undesirable consequence.

To overcome this problem an additional edge attribute is applied to each edge which identified the set of all edges in the original SRG that were used to infer the edge in question. This edge attribute is called the "source set" of the edge.

The controller thus determined a sequence of located pattern instances $(G[\alpha_1(P_1)], \ldots, G[\alpha_k(P_k)])$ and their applications in order to compute a given transformation on an SRG G.

The next section will discuss methods on how this can be used to dynamically create correct tracking engines.

4.7 Component architecture and data flow networks

As noted above, the spatial relationship graph, which has been the main focus of the discussion, does not deal with actual sensor data. As a general rule, the information associated with each edge describes rather the availability of a spatial relationship as well as its characteristics than the concrete measurements of the tracking process. The SRG thus captures the structural and geometric properties of the tracking setup without processing actual sensor data. Note that this differs from the earlier definition by [NWP+03],[NWB04] which also included the measured data as edge attributes in the graph.

At first glance, this may seem to ignore the primary interests of concrete Augmented Reality applications. It is obvious that applications do require realtime processing of and access to tracking and sensor data. The previous section has shown how the SRG

can be used to derive a sequence of algorithm applications necessary to compute the spatial relationships required by the application from the various sources available. Thus, the abstraction provided by the spatial relationship graph indeed frees the application developer for dealing with complex and dynamic tracking setups and allows the user to focus on the desired data.

To further reduce the effort of the transitions between abstract SRG representations, derived algorithm sequences and concrete data manipulation, the Ubitrack tracking framework includes a fully integrated library of common tracking algorithms and a flow based data processing engine.

The collection of algorithms contains a number of commonly required transformations as well as more involved methods, such as Hand-Eye-Calibration or absolute orientation. Additional hardware abstraction drivers for tracking sensors commonly encountered in AR setups are provided by the library. The library was developed with easy extensibility in mind and also offers interfaces to different programming languages on various platforms. All the algorithms and device handlers are both implemented as components and abstractly described by associated patterns. This enables complete reasoning and implementation of dataflow graphs derived from reasoning about SRGs.

The individual tracking algorithm components are represented as nodes in a dataflow graph. Each node may have an arbitrary number of input and output ports which are interconnected between different algorithms, effectively passing sensor data between different stages of computation. This forms a directed graph, where the edges indicate the direction of sent messages.

Each message contains a typed sensor measurement or other datum. Furthermore each message contains a timestamp that indicates the exact time at which the particular measurement was obtained. This is especially important for further fusion of data from different sensors.

All the messages are stored in a centralized message queue and sophisticated methods for message distribution and scheduling are provided.

A major goal of the library is to be able to perform no less efficiently as a standard, static implementation of a tracking setup. This means that the overhead imposed by the dataflow framework has to be negligible, ensuring that SRG derived solutions to be competitive as compared to specialized solutions.

While basic system characteristics such as robustness and efficiency are required for any usage, the ubitrack dataflow implementation further focuses on aspects which become necessary in ubiquitous tracking scenarios.

4.7.1 Data flow construction

Using the subsequent pattern application on a spatial relationship graph the sequence of pattern applications necessary to compute a given query edge can be derived. From this sequence a data flow graph can be created, which describes the connected components

that perform the actual computations.

An example of a data flow graph that performs the computations as explained in section 4.5 is shown in figure 4.5.

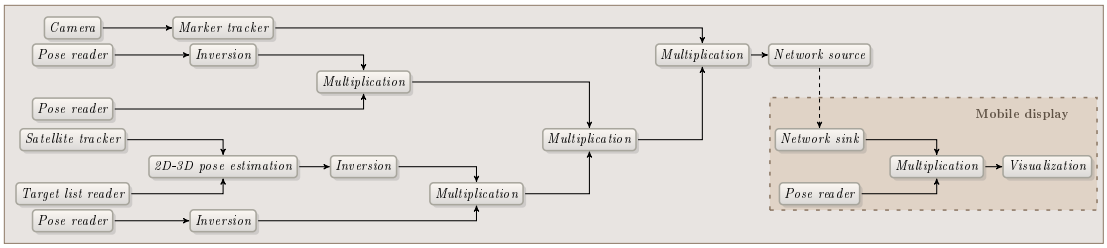


Figure 4.5: Data flow graph for implementing the example scenario from section 4.5.

Constructing the data flow graph from the sequence of located pattern instances $(G[\alpha_1(P_1)], \ldots, G[\alpha_k(P_k)])$ is done in a rather straightforward fashion: All edges in the initial spatial relationship graph are associated with a tracking component or some other service that initially provides the unprocessed measurements. These components serve as the basic inputs to the data flow network.

When an expanded instance of pattern P_i is applied in the SRG, this creates a new component in the data flow graph. This component executes an algorithm from the set of associated algorithms of the pattern P_i and provides new outputs in the data flow graph, which are associated with the new edges $E[\alpha(O_{\bar{P}_i})]$ in the SRG. The inputs of the component are connected to the components associated with the matched input edges $E[\alpha(I_{\bar{P}_i})]$. This is repeated for all pattern applications of the sequence till the desired measurement becomes available at the end of the data flow graph.

4.7.2 Dynamic reconfigurations

Using the mechanisms discussed so far, the SRG and the dataflow generation are able to handle diverse, heterogeneous tracking setups, where only structural information about the tracking setup is available. To fully support ubiquitous augmented reality scenarios such easy adaption is required, but may not be sufficient in the general case. In an ubiquitous tracking scenario it may be necessary to cope with the dynamically changing availability of certain tracking modalities while disrupting the operation of the application as little as possible. Thus a more active role of the tracking system is imperative, rather than static adjustment of different tracking solution to different environments. In a volatile environment it is necessary for the tracking solution itself to be dynamic and to react to changes in the environment.

In consequence, the library must be able to dynamically reconfigure the dataflow based on structural changes as reflected in changing SRGs. This has to be done while

minimizing the impact on the application. More specifically this implies that the dataflow has to keep processing data, while being reconfigured or rearranged. This requires that any dataflow component can be stopped and the dataflow itself be rearranged at any moment.

First the difference between two dataflow graphs has to be determined by comparing the contained components, their individual configurations and the connections. Any component that differs in at least one aspect is affected by the reconfiguration and is assigned to the change set. The remainder of the already running graph is left untouched. All components in the change set are temporarily stopped and reconfigured as necessary. Messages in the central queue which are to be processed by already stopped or deleted components are detected and accordingly discarded. If new components are required or existing components are no longer needed, these nodes will be created or destroyed respectively. As a final step the required connections in the graph are created and the components are restarted. Note that any part of the network that is not affected by the reconfiguration, continued to operate normally.

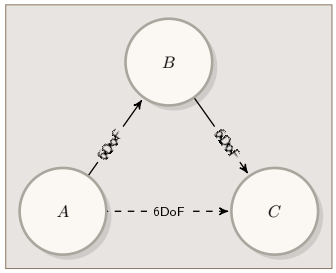
4.7.3 Synchronization

Another crucial problem that needs to be overcome for automatic construction of data flow networks is the consistent treatment of measurement time.

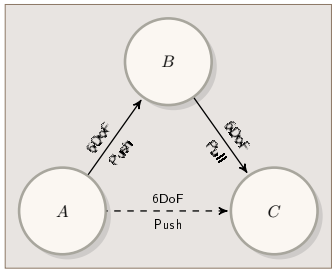
Consider, for example, the concatenation pattern applied to measurements of two independent trackers. In this case, the multiplication only yields correct results when the measurements were made at exactly the same time. In general, this is not the case and interpolation or extrapolation components need to be inserted into the data flow network unless the two sensors are synchronized in hardware. To amend this situation, the concept of pull- and push-style interfaces was introduced in [Pus06]. A push-style interface represents the classic way of sending tracking data events as soon as the measurement is available, while a pull-style interface allows the receiving end to request tracking data for any desired timestamp.

By additionally attributing each edge in both the SRG and the patterns with the respective interface style, this mechanism makes it possible to also automatically integrate interpolation or sampling components in the data flow network where necessary. Figure 4.6 shows the different possibilities to annotate the pose multiplication pattern. To convert between push and pull interfaces, additional patterns are introduces such as the interpolation or sampler pattern (see figure 4.7).

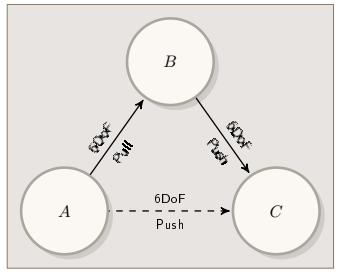
Note that a concatenation with two push-style inputs is in general not desirable because of the argument given above, but may be useful if the additional requirement that both inputs stem from the same tracking source (e.g. two markers detected in the same camera frame) is met or the sensors are synchronized in hardware. This requirement is expressed by the additional "sync" attributes in figure 4.6 d), which has to be equal for both inputs and is hence inherited by the output.



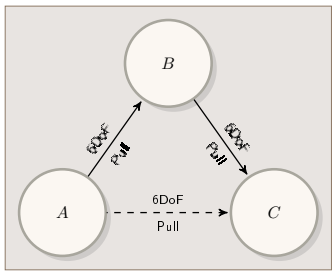
(a) Multiplication pattern



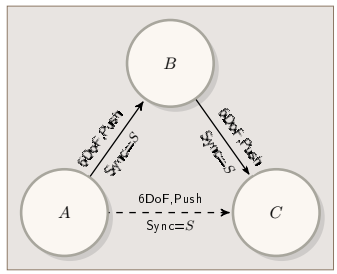
(b) Push-Pull synchronization



(c) Pull-Push synchronization

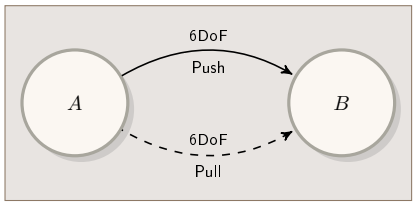


(d) Pull synchronization

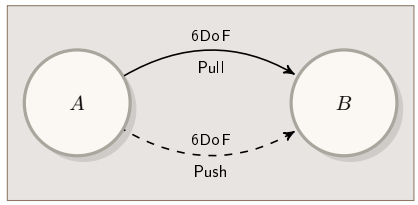


(e) Synchronized push synchronization

Figure 4.6: Synchronization possibilities for pose multiplication



(a) Interpolation pattern



(b) Sampler pattern

Figure 4.7: Examples for Push/Pull conversion

The additional blowup in patterns produced due to this circumstance can be handled by a generic implementation of tracking algorithms and dynamic configuration of its input and output logic.

4.8 Distributed System Architecture

As mentioned above, in a ubiquitous tracking scenario, the tracking system has to take on an active role and has to dynamically react to changes in the environment or in the intentions or desires of the user. While the concepts presented so far suffice to implement an adaptive and dynamic reconfigurable tracking solution on a single machine, ubiquitous augmented reality scenarios are expected to involve a number of different computing entities. For example various clients may exist in the area of a dedicated tracking system, which is hosted on a different machine. In order to coordinate the needs and abilities of these different a suitable distributed tracking middleware is necessary, which also pays attention not only to the requirements of handling real-time data but also is able to handle clients in a ubiquitous tracking scenario.

In this section, the Ubitrack system architecture is outlined, which aims to implement such a distributed tracking reasoning system.

Efficient communication of tracking data at runtime As already stated above, interposing a tracking middleware must not lead to a degradation in tracking quality or performance. This especially means that sensor measurements have to be transported and processed as fast as possible so that there is no delay for the AR-application that would not also be existent if the middleware was not used.

Queries for involved objects and transformations Applications have to be able to query for available objects and their spatial relationships. This is a direct consequence of separating the tracking and application domains. If applications are developed without knowing the concrete layout of the underlying tracking environment, they have to express their needs regarding available objects and the spatial relationship(s) between them in some way.

Varying capabilities of the participating application platforms Especially for mobile AR applications running on handheld devices, the available interfaces and also the computing power are often limited. In order to provide them with high-quality tracking resulting from different sensors and computationally expensive sensor fusion, the only solution may be to attach the sensors to another machine and also to source out some or all of the computations. The final results can then be transmitted via network and directly consumed by the visualization unit on the mobile device.

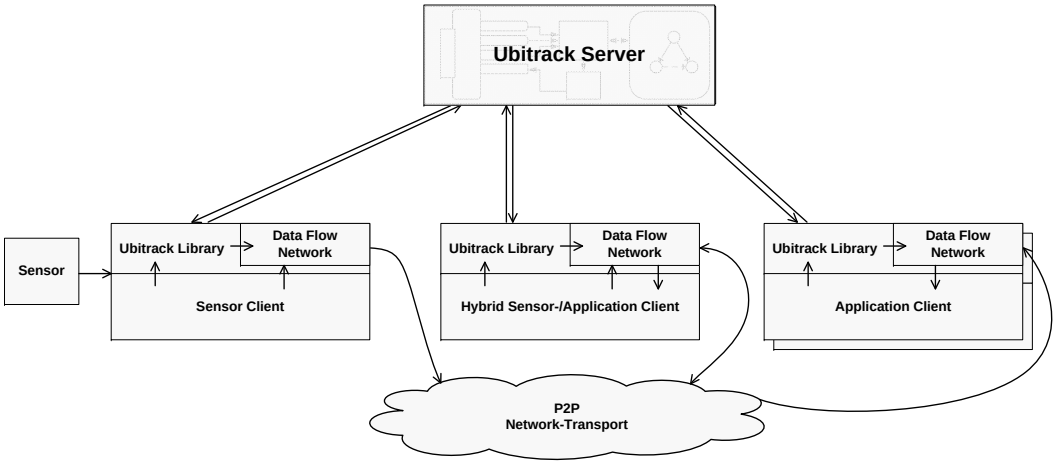


Figure 4.8: Ubitrack Architecture overview

Bearing this in mind, the Ubitrack middleware implements a *centrally coordinated peer-to-peer architecture*, combining the advantages of both a pure client-server and a peer-to-peer architecture. Figure 4.8 shows an overview of the Ubitrack distributed environment.

The client application registers its needs and abilities with the Ubitrack library which in turn registers itself with the Ubitrack server. Thereupon, the server generates suitable data flow networks for each client and also coordinates the peer-to-peer network between the clients. The server itself is not involved in the runtime flow of tracking data. Depending on the type of client, tracking data is flowing into and/or out of the client's data flow network.

Compared to the pure peer-to-peer approach described in [NWB04], the coordination and maintenance of a tracking environment is much easier using the client-server-approach. For the highly time-critical propagation of tracking data, however, it is advisable to use the shortest communication paths between suppliers and consumers, avoiding latency due to the indirection via a server and, as a consequence, avoiding also the corresponding processing load for the server.

4.8.1 Entities

A *client* in the Ubitrack environment is an entity that is involved in the flow of tracking data. It can run on a dedicated hardware platform, though this is not a requirement. For all functionality related to tracking, the *application* software running on the client relies on the *Ubitrack library*. The application uses the API of the library in order to register its base SRG(s) describing the spatial relationships between locally connected *sensors* and other real or virtual objects at the *Ubitrack server*. Furthermore, using the

same mechanism, it also registers its requirements towards the tracking systems in terms of application queries. The server coordinates the requests of all clients and assigns a data flow network to each of them.

For the purpose of clarity, two archetypical kinds of clients are outlined (see also figure 4.8), although very often, the client will be a combination of both.

Sensor client A sensor client provides tracking data to the rest of the system. For this purpose, it instantiates a data flow network which uses the measurements of the sensors it owns, potentially filters and/or fuses them and finally feeds the results into the peer-to-peer network so that they can be further processed in data flow networks running on other clients.

Application client The application client is the exact counterpart of the sensor client. It is a pure consumer of tracking data and relies on the data flow networks of other clients to provide it with the data it needs. The most simple data flow network of an application client consists just of two data flow components, one receiving data from the peer-to-peer network and another handing it over to the application. An examples for an almost pure application client is the handheld visualization device of section 4.5.

In the following the individual components that make up a Ubitrack client will be further detailed.

Application The application software relies on the API of the Ubitrack library for all purposes related to tracking. Furthermore, it may perform various other more or less complex tasks. A typical AR application for example would register a query for the transformation of some virtual objects to the HMD coordinate frame and use the tracking results to render them in the HMD.

Ubitrack Library The Ubitrack library encapsulates all communication between the client and the Ubitrack server and provides means to instantiate a data flow network according to the description constructed by the server. It contains all the algorithms and concepts introduced in the previous sections.

Before the server can compute descriptions of suitable data flow networks, the client applications first have to announce their tracking abilities and needs via the API of the library. A pure sensor client on the one hand registers only SRG parts describing sensors, tracked objects and known spatial relationships between them. An application running on a pure application client on the other hand registers only queries describing the application's need(s) in terms of objects and spatial relationships satisfying certain predicates. Of course, the application can also change or delete SRG parts or queries previously registered at any time without restarting the system. SRG updates and queries

are registered by the application via the API in the *Ubiquitous Tracking Query Language (UTQL)* format and are handed over to the server.

The Ubitrack library of an individual client also uses UTQL to register all the data flow components it is able to instantiate as part of its data flow network. This is done by describing their spatial relationship patterns. Those SR patterns directly depend on the data flow algorithms that are implemented within the library and are therefore generated by the library itself. The list of available SR patterns may differ from client to client due to differing library versions or computing power.

Whenever the server is able to resolve an application's query, it sends back the description of a data flow network in the UTQL format that is in turn instantiated by the Ubitrack library. It may also happen that the server reacts to dynamic changes in the tracking environment initiated by other clients by sending an updated description asynchronously. The described data flow network is suitable for deriving tracking data for those spatial relationships that have been specified in the application's query before.

In the standard case, the application does not have to bother with UTQL directly. Requests can be generated by a configuration tool and the corresponding server responses are interpreted for data flow generation directly by the Ubitrack library. Therefore, all an application needs to know for obtaining tracking data for a certain edge is the type of transformation (e.g. 6DoF) and the synchronization style (push/pull).

Sensors A *sensor* in the context of Ubitrack is considered to be a black box that provides tracking data at a high level of abstraction. It may supply for example 6DoF pose, 3DoF translational or 3DoF rotational information. A typical sensor provides its measurements via a push-style interface. In the simplest case, a sensor is integrated into the data flow network by instantiating the corresponding sensor data flow component which mainly consists of a driver and maybe also some algorithms that provide the necessary level of abstraction.

More complex sensors, such as a markerless optical tracking system requiring other sensors for initialization, may need to have more extensive access to the Ubitrack system. Such sensors would therefore be realized as dedicated sensor clients (see Section 4.8.1) at the application level using the Ubitrack API.

Ubitrack Server The Ubitrack server is the central coordination unit of the system. It administrates a list of all base SRGs, application queries and SR patterns that may be registered and deregistered by the participating clients at runtime. All currently registered base SRGs together make up the *world SRG*. It represents the general knowledge about the tracking environment provided by the clients and changes whenever the list of registered base SRGs changes.

The Ubitrack server always tries to fulfill all registered queries. For this purpose it applies the registered SR patterns to the world SRG with the purpose of somehow

deriving the SRG edge denoted by the query. As soon as the server succeeds in finding a sequence of pattern applications in order to derive a queried edge, it communicates this knowledge to the client the query came from. In case tracking data has to be provided by other clients via the peer-to-peer network in order to compute edges contained in the query, the server also updates the data flow description of the other clients. Thereby, it also considers the varying client capabilities.

It is important to state that the Ubitrack server itself does not know anything about the underlying data flow algorithms. Its derivations merely depend on the question about which constellation of edges in the SRG is necessary so that a certain SR pattern can be applied in order to provide a new edge.

Runtime While the tracking environment is in operation, an administration tool is able to provide insights into the internals of the Ubitrack server, namely the current list of registered base SRGs, SR patterns and application queries per client as well as the current world SRG and the data flow networks derived from it.

4.8.2 Ubiquitous Tracking Query Language (UTQL)

The communication between the Ubitrack server and its clients requires a dedicated protocol to support the following operations:

Client Registration To support clients with different processing capabilities, or simply running different versions of the Ubitrack framework, the clients need to tell the server which data flow operations they support.

Base SRG Registration As the server's world SRG is initially empty, clients need to be able to update the global base SRG with the nodes and edges they provide.

Query Registration Clients need to be able to express their interest in arbitrary parts of the global SRG, including previously unknown nodes that are only characterized by their attributes.

Removal of Clients, Base Subgraphs and Queries For truly dynamic scenarios, the protocol also needs to support the removal of clients, subgraphs and queries.

Data Flow Construction As a response to a query, the server must be able to tell one or more clients which data flow components to instantiate and how to connect them.

The first part of the solution to this problem is to model all of these operations as spatial relationship patterns. Then, the whole UTQL conversation between the clients and the server consists of lists of patterns. A detailed description of the syntax and semantics of the UTQL language can be found in [PHEK07].

To register at the server, a client sends the list of all supported data flow components, represented by their SR patterns. An SRG registration can be seen as a special case of an SR pattern which has no inputs, and therefore can be inserted unconditionally into the global SRG. Similarly, a query is a pattern that only contains input nodes and edges, but does not produce new outputs. By giving all patterns a unique name (per client), a pattern, query or SRG registration can be revoked at the server by sending an empty pattern of the same name with no inputs and outputs.

To instantiate a data flow network, the server sends back concrete instances of the previously specified patterns to the clients, each representing a data flow component, together with instructions on how to connect them.

4.8.3 Server Implementation Details

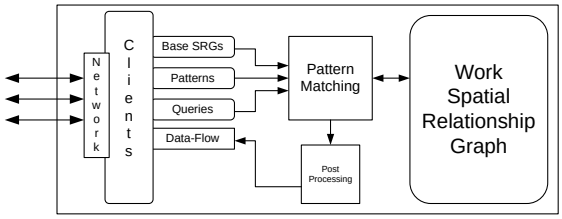


Figure 4.9: Server structure

The remainder of this section will give a more detailed look on the necessary components and inner workings of the Ubitrack server component. Figure 4.9 provides a general overview over its different parts.

4.8.3.1 Data Structures

The main task of the server is to provide responses for spatial relationship queries and to coordinate the communication between clients. As it initially starts in a completely empty state, the server can only act on information provided by the clients that register their respective knowledge and capabilities. This helps to avoid additional assumptions about the structure of the tracking system and adds to its overall flexibility.

Network Interface For bi-directional communication with the clients, the server needs to have some sort of network interface. For the lower layers, a simple TCP socket network protocol with keep-alive packets is used to allow stream orientated data transport and to be able to detect disappearing or defunct clients. The data itself is communicated across these network links as UTQL documents.

Client Management To manage and distinguish all clients currently connected, the server assigns each connection a unique client ID. These IDs are stored in the client list which serves as the main reference point for attributing all information that is kept in other data structures to uniquely identify the originating source. Furthermore it is also important to keep references from the client entry to all the contributed information, as these need to be removed if the client exits or becomes otherwise unavailable.

Base SRG, Pattern and Query Management When processing the client announcements, the server gathers different types of data for each of the different types of announcements. To store these for further processing and for reference for later response generation a separate list is used for each type individually. When processing the client announcements, the server may gain different types of data and for each of these types also a separate list is used to store the data of the announcement for further processing. As described above each of these entries are linked to the client via its respective ID and the client list.

For efficiency, equal patterns from different clients are not stored as individual items but rather are condensed into one pattern which may be instantiated on a set of different clients.

If a client deletes information (by sending an empty pattern), the server can remove the corresponding entry and act upon the new situation. Similarly if a client becomes unavailable, the server deletes all information which was associated with the client.

Work Spatial Relationship Graph The main central data structure which not only captures the raw data provided by the clients but also represents the server's main working ground is the work SRG. It is the representation of the state of the overall tracking systems as viewed from the server's perspective, integrating all the different bits of knowledge and capabilities of the clients. In this sense it is not only a description of the current running state, but also of potential future states since this is the working space for the pattern matching system which stores every interesting edge derived, even if not immediately needed by any query.

4.8.3.2 Work Flow

This process of acquiring data and computing responses is also reflected in the basic work flow steps performed by the server and the corresponding data structures. Note that in a real world application of the server, the following basic work flow steps can occur in arbitrary order, especially if the server is handling more than one client.

1. Reception of announcements from client

2. Assembly of base spatial relationship graph

3. Application of pattern matching to infer new relationships
4. Matching of queries
5. Construction of data flow network responses
6. Evaluation of multiple solutions for identical relationships
7. Distribution across multiple clients
8. Notification of clients of new responses

Client Announcements The document sent by a client consists of a series of patterns as described in Section 4.8.2, which in this context are also called announcements. For every received announcement the first action performed by the server is to determine its type by classifying it according to its number of input and output edges.

Every announcement thus may be one of the following types and triggers a corresponding action in the server:

Base SRG Registration A base SRG fragment is registered for inclusion into the server's work SRG. This represents an announcement of infrastructure which is available to and can be used through the announcing client.

Pattern Registration A client can announce to the server its computational capabilities by specifying the patterns it is able to instantiate. The patterns available to a client may depend on computational power or installed third party hardware or software.

Query Registration A client can express its interest in a certain spatial relationship expressed as a query. The server will try to construct a data flow satisfying this need which is updated every time the tracking situation changes either by registering new base SRGs, new patterns or by deletion thereof.

Deletion Each of the former patterns can also be deleted by the client at a later point in time, which may also have effects on the tracking situation of other clients.

Base Spatial Relationship Graph Assembly When receiving a base SRG registration announcement, the server merges the SRG part from the client into its work SRG as new base graph edges. This is done by identifying nodes with matching IDs as equal and thus fixating the SRG fragment in the global view. At the same time distinct node attributes specified by different SRG fragments are unified in the work SRG. Finally, a reference of the SRG announcement is stored in order to keep additional information (for example dataflow configuration related parameters) intact.

Pattern Registration and Detection By sending a spatial relationship pattern notification the client tells the server about new computational capabilities available to the client, which can be used to answer queries posed by this or other clients.

Using these registered patterns the server may perform an explorative search on the possible pattern instantiations in order to fulfill any registered queries repeatedly applying all registered patterns.

Up to now a detected pattern is represented as a node- and edge-matching between the pattern graph and the work SRG. In order to be able to finally merge the detected pattern into the work SRG the matching has to be applied to the pattern graph thus fixing the matched nodes and edges with respect to the work SRG. While input nodes are fixed by setting the input nodes IDs, for the input edges an edge reference is created and stored as an additional edge annotation.

Query Detection The query matching step of the server work flow functions much like the pattern detection step as queries are special cases of patterns with no output section. Thus for each query all matches are detected and for each of these matches a data flow description is generated by collecting all relevant pattern instantiations. The collection process works in a depth first search-manner starting form the query and descending every edge reference stored in an input section and terminating at the leaves formed by the base SRG registrations. If a pattern is referenced more than once it is nevertheless collected only once.

Data Flow Evaluation and Network Distribution The computed data flows pass a post processing stage before they are finally sent as a response to the client.

First, in certain configurations, it may be desirable for a client to only receive one data flow configuration per spatial relationship (i.e. a pair of SRG nodes) while in others it may want to be informed about all possible ways of computation. In the first case the generated data flow configurations have to be evaluated before producing the responses for the clients. Only the one that is evaluated to be "best" is kept as a response whereas in the second case all are kept.

In the last post processing step, the constructed and evaluated data flow descriptions are distributed across the clients according to their individual capabilities. A query is always locked to the client which registered the query, a tracker is quite possibly dependent on some hardware and thus may also be locked, whereas basic patterns like inversion can be computed by most clients. Starting from the query, the server iterates over all the pattern instantiations which constitute the data flow in the order of edge references and assigns a client ID from the set of capable clients for the pattern in question to each subgraph. The server tries to keep the data flow on the same client as long as possible, but if a needed pattern is not available or locked to a specific client, the computation has to continue elsewhere. To be able to do so, the server inserts network

transport components into the data flow and adjusts the edge references accordingly.

Response Generation Finally all patterns are grouped by clients and are converted back into UTQL representation, which is then sent as response to the clients. Therefore clients may receive asynchronous responses at any time, as an announcement of one client may change the data flow on another client.

5 Temporal Sensor Calibration

The work described in this chapter was partially funded by the Trackframe project and partially by the AVILUS project. It was conducted in close cooperation with Michael Schlegel. Parts of the description below are based on the presentation [HSK09] at ISMAR 2009.

Temporal calibration In order to correctly combine data from two sensors, it is necessary to know the exact temporal relationship between data acquired from different sources. For example a concatenation of poses may only be performed on data that refers to the same moment in time of the physical reality. Such sensors are called "synchronized" or "temporally calibrated".

Synchronization can either be achieved by hardware or by logical means on the sensor data. For hardware synchronization the acquisition of sensor data is triggered by a central hardware clock (trigger signal), connected to all participating sensors. Usually this trigger signal is derived from an oscillator in one of the sensors itself. This approach is the common approach in current sensor-fusion scenarios. Also note that this almost completely prohibits dynamic construction or reconfiguration of sensor setups or the use of common off-the-shelf hardware, which mostly lacks hardware synchronization interfaces.

Synchronization in software, on the other hand depends on correctly attaching timestamps to each sensor measurement. Such timestamps are preferably provided by the sensor itself or have to be generated by the tracking framework as soon as the measurement enters the system. The Ubitrack system described earlier uses this approach by assigning a timestamp to each datum present in the data-flow network.

In either case the timestamps of the different sensors have to be adjusted to refer to a common time base. The previously discussed push/pull mechanisms for sensor synchronization only account for asynchronous sensor updates, while implicitly assuming that the time base of the individual sensors is equal. This chapter presents a method to generally solve this calibration problem of relative lag between arbitrary sensors.

Temporal calibration is also important for parasitic tracking due to the number of imprecise, unsynchronized devices that are involved. While precision in the context of Parasitic Tracking is not as critical as for local or industrial settings, yet errors due to sensor asynchronicity negatively influence sensor fusion results, especially during calibration or registration of sensor parameters.

This method is applicable either as an offline one-time calibration step or as an online recalibration method. Periodically online adjustments to the sensor calibration enable

dynamic adaption for sensor setups that can change at runtime and compensation for changing environmental influences (such as changing CPU load or network delay). Such recalibration is especially relevant in distributed sensor fusion setups which involve computer communication or in long-term applications. Over longer periods of time the relative lag between two sensors may change due to the temporal drift caused by crystal aging or crystal instabilities. This enables calibration of ad-hoc dynamic sensor fusion setups in highly flexible ubiquitous tracking scenarios, where no a priori knowledge about the involved sensors is available.

This method has further interesting applications apart from dynamic sensor fusion. One example is error evaluation of tracking setups where the data measured by a tracking system under test is compared to ground-truth data from a second, more precise system. Unless the data is only recorded when all sensors are completely motionless, a temporal offset between the system under test and the ground truth causes an additional tracking error which disturbs the actual experiment. A scenario where this kind of calibration was used is described in [GGV+10].

5.1 Relative Sensor Lag

The problem of adjusting timestamps between sensors can be considered for an arbitrary numbers of sensors. For simplicity only the case of a single pair of sensors S_1 and S_2 will be considered, assuming that they are rigidly connected. The principle can be extended to n sensors, where each pair is calibrated accordingly, resulting in $\binom{n}{2}$ calibrations.

Considering an event happening in the real world at time t_0, two sensors S_1 and S_2 sense this event as an analog physical input and convert it into digital representations at times t_{S_1} and t_{S_2}. In general, the observation times of the same event differ, since every type of sensor requires a different amount of time for the internal signal processing. Further delay will be caused by the various communication stacks of the operating system or by network transport of the measurement data. The measurements arrive at the tracking framework at times t'_{S_1}, t'_{S_2} and are accordingly tagged with timestamps at that point in time. As soon as a reliable timestamp is attached, all further processing delays can be neglected as far as correctness of sensor fusion is concerned. A schematic of the mentioned points in time can be seen in figure 5.1.

The accumulated lags can be reduced to one single delay for each sensor $\Delta t_{S_1} = t'_{S_1} - t_0$ and $\Delta t_{S_2} = t'_{S_1} - t_0$. For sensor fusion it is only necessary that all sensors are temporally aligned relative to each other; the offset to the unknown true point in time t_0 is not relevant. To align the sensor data, it is sufficient to determine the temporal offset $\Delta t = t'_{S_1} - t'_{S_2}$, which can be achieved using the method described below.

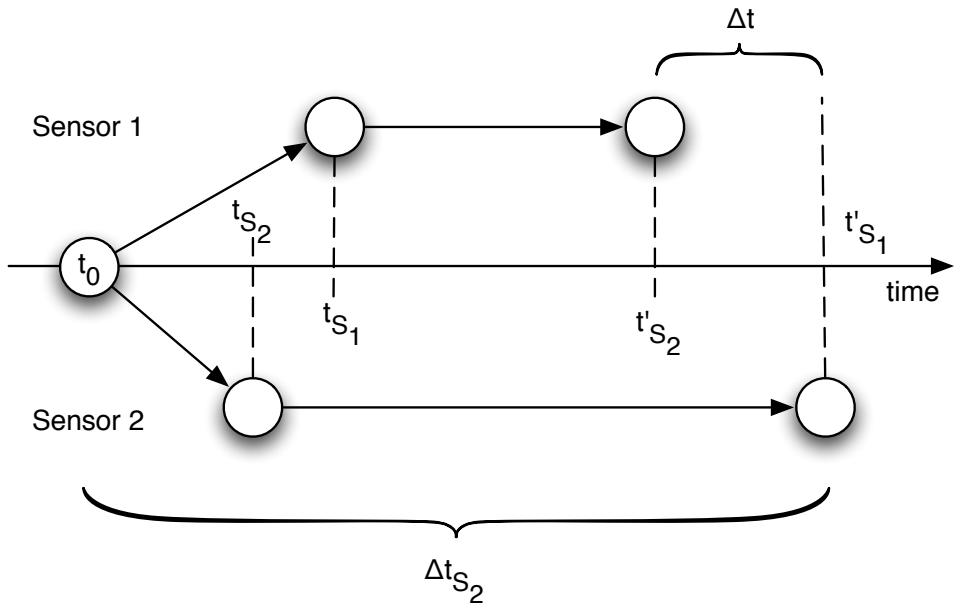


Figure 5.1: Schematic visualization of different points in time which are relevant for a temporal calibration

5.2 Related Work

Several publications deal with the topic of sensor fusion with a wide range of different focuses.

Recent examples of successful applications of sensor fusion for different applications include outdoor augmented reality [PT01] or [RD06] which describes the fusion of separate location and orientation sensors to derive a full pose. Furthermore in the fusion of inertial tracking devices with vision based tracking, interesting results were achieved [BS08]. The temporal calibration problem so far is mostly solved only for particular hardware setups. In most cases components are either hardware synchronized or the lag between different sensors is tuned in software by experimental means.

Moreover, the negative influence of lag on the general usability of AR applications is generally agreed upon (see for example [AB94, AB95] or [WB97, Wel96]).

In [ASB+04] and [ASB07] a sensor synchronization scheme is discussed for the application of calibrating inertial sensors and vision based tracking. Their approach relies on detecting abrupt movements in both the camera image as well as the inertial tracker. In [BS08] it is indicated that the employed camera and inertial tracker are synchronized via a common clock source that triggers both sensors. Such a setup using hardware syn-

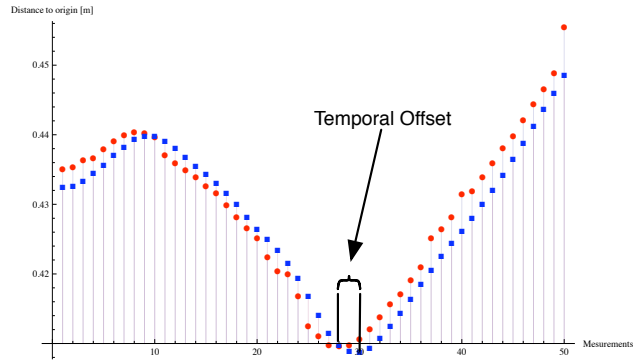


Figure 5.2: Data of two different sensors; the temporal offset is clearly visible.

chronization currently seems to be the most common case, but in general is prohibitive in ubiquitous tracking scenarios.

In [LBMN09, SS04] a static temporal offset was calibrated in a static manner during the spatial calibration process. They shift one signal in time while constantly calculating the geometric residuum. The temporal offset which minimizes the geometrical residuum is then taken as the temporal offset (see section 5.4.3 for a comparison with the approach presented here).

In [JLS97] an in-depth study of latencies occurring in an AR system was conducted. In addition to the measurement of the overall latency they also performed a calibration of the offset of two sensors during setup time. Instead of using mathematical optimization however, they built an AR system and used a manually adjustable prediction to determine the temporal offset between the sensors by visual means.

A different approach to calibrating the overall latency of an augmented reality system including tracking, rendering and augmentation was presented in [SSK+07]. The system used a novel encoding of the current time included in the current visualization, which was in turn fed back into the system.

5.3 Calibration Method

5.3.1 General Strategy

Let T_{S_1}, T_{S_2} be the sets of all timestamps $t \in T_{S_1} \cup T_{S_2}$ where measurements of either S_1 or S_2 respectively were taken. The two data time series $X = \{x_t : t \in T_{S_1}\}$ and $Y = \{y_t : t \in T_{S_2}\}$ can be defined as the actual sensor data x_t, y_t from sensors S_1 and S_2 respectively at the individual timestamps. Note that the timestamps at which S_1 and S_2 acquire data, are in general not the same and thus T_{S_1} and T_{S_2} are distinct.

For each pair of signals a suitable similarity measure $\rho_{X,Y}$ can be computed that measures the mutual agreement of the signals. The similarity measure is assumed to be normalized such that $0 \leq \rho_{X,Y} \leq 1$. Signals with $\rho_{X,Y} = 0$ are called *orthogonal*, and analogically signals with $\rho_{X,Y} = 1$ are called *identical*. In general $\rho_{X,Y} \neq 1$ even if the same type of sensors is used since both measurements will be individually affected by noise and other kinds of errors.

The time-offset of the two sensor signals can be determined by consecutively shifting one signal by small offsets against the other signal until a maximum of agreement is reached. The shift value for which this maximum is attained is identified as the temporal offset between the sensors. This is a well-known approach in signal processing [Car81].

This leads to the following formula. The task is to find a Δt which maximizes the similarity of both signals.

$$\Delta t = \text{argmax}_{\delta t}\{\rho_{X,Y^{(\delta t)}}\},$$

where $Y^{(\delta t)}$ is the signal Y shifted in time by δt.

5.3.2 Properties of Various Time Series

Time series have a number of properties which can have different impacts on the performance of the calibration procedure.

Dimensionality Naturally, different sensors can sense different kinds of physical activities. Accordingly, the degrees of freedom (DoF) of the measurements and the number of dimensions required to represent these measurements vary. The most common types of tracking sensors measure 3D position, 3D orientation or both simultaneously (6 DoF), but also less common combinations exist (for example 3DoF position and orientation poses on a 2D surface, which will be used in a later chapter). The calibration procedure has to take these different types of measurements into consideration.

Registration In order for the signals of two different sensors to be comparable it is assumed that the geometric registration between the two sensors is static and known and that the sensor measurements have been transformed into a common coordinate frame. This spatial relationship can usually be determined by spatial calibration. Note that since the accuracy of this spatial calibration also depends on the temporal alignment of the sensors, the spatial and temporal calibration can be used in an iterative or adaptive process. Moreover, as the temporal calibration process is rather robust against spatial registration errors, the requirements on the spatial accuracy is rather low.

Signal-to-noise ratio The Signal-to-noise ratio (SNR) of a signal is defined as the ratio between the power of a signal and the power of the measurement noise, respectively the

square of the individual amplitudes.

$$SNR = \frac{P_{signal}}{P_{noise}} = \left(\frac{A_{signal}}{A_{noise}}\right)^2$$

For tracking sensors this characteristic describes the amount of movement as compared to the measurement noise of the sensor. Time series with large SNR usually feature large movements or fast velocities, whereas a low SNR indicates little activity or very slow movement. Thus the noise characteristics of the sensor also determines the minimum movement required to produce a signal exhibiting sufficient SNR.

Sampling rate Another basic assumption about the tracking data is that each sensor acquires its corresponding measurements of the real world at periodic intervals in time. Each measurement is called a sample point of the sensor and the periodicity of these samples is called the sampling rate. Thus the sampling rate is a basic characteristic of the sensor that also determines the maximum temporal resolution of tracked movements.

While the sample points are usually assumed to be distributed equidistantly in time, in practice this assumption does not always hold true. The implications thereof are further discussed in section 5.3.6.

5.3.3 Overview

As mentioned above, the sensor data that is acquired by spatial sensors usually features high (commonly 3 or 6 DoF) dimensionality. In the course of the proposed calibration method these measurements are reduced to one-dimensional signals. The rationale behind this is that while losing physical representational precision of the sensor signal, the time calibration primarily utilizes the shape of the sensor signals. Similar to the situation in machine learning [Fod02], the reduction in dimensionality can actually enhance the performance of the time calibration by making the characteristics of the sensor movement more prevalent. This results in improved behavior in low SNR settings, as will be discussed later.

Computing the similarity on signals of reduced dimensionality is computationally faster, which enables real-time on-line calibration in the first place. Furthermore, the adaptation of the calibration process to different pairs of sensors is simplified, even in cases where no immediate geometrical comparison of the sensor data is available.

To implement the strategy for determining the relative lag of a sensor pair, the proposed method needs to perform the following steps.

- Segmentation
- Dimensionality reduction
- Interpolation

- Time Delay Estimation
- Aggregation

These steps will be discussed below.

5.3.4 Segmentation

The sensor data is assumed to be an endless stream of measurements. Thus, as a first step, this data is divided into chunks of equal length (duration). This serves two specific purposes. Shorter chunks of data are easier to process, both in terms of speed and complexity. Moreover the significance of the comparison of very large chunks tends to decrease due to the increased existence of tracking outliers.

Second an on-line estimation requires some sort of segmentation in order to produce results during the runtime of the procedure. In particular the stream is not assumed to have any sort of end. Although it is possible to aggregate the calibration results of severals chunks, global optimization strategies working on the complete history of the tracking data are in general not feasible.

The segmentation is performed by accumulating incoming samples in a separate buffer for each sensor. As soon as the required amount of sensor data has been reached, the buffer is copied and processed by the calibration method. Note that if the sampling rates of the sensors differ, this condition will depend on the amount of data produced by the slower sensor.

There are two basic strategies either to produce subsequent disjoint segments or to produce overlapping segments, where each new segment consists of a certain amount of old data with new data appended. In the former case the buffers are completely emptied between segments, whereas in the latter case the buffers are only partially cleared and shifted.

5.3.5 Dimensionality reduction

As previously discussed, the main part of the calibration method will not deal with high dimensional tracking data but rather with reduced one-dimensional signals. The most crucial aspect of this reduction is not to maintain the immediate relationship to any specific geometric interpretation, but rather to maintain the comparability of similar events as registered by different sensors. Furthermore, the presence of spatial events, as described above, which are registered by the sensors, should still be discernible from the reduced signal.

The different projection methods are only discussed for sensor S_1 and it is assumed that the same projection is used for the data of sensor S_2. In general a mapping $f : \mathbb{R}^n \mapsto \mathbb{R}$ is required which reduces the high dimensional measurement data x_t to a one-dimensional signal $\hat{x}_t = f(x_t)$ for all timestamps t.

To achieve this goal a number of different approaches are feasible, which can be classified as follows.

Static computations A simple, yet useful method to reduce the dimensionality is to use a static projection or computation for each measurement. One example of this class of reduction is simply taking a single component (for example the x-component) of a 3D position measurement by the projection $\hat{x}_t = w_x^T x_t$ with projection vector $w_x = (1, 0, 0)^T$. These kinds of projections suffer from a dramatically reduced SNR in cases where the movement of the sensor is mainly perpendicular to the projection vector.

Another kind of static computation is to use the Euclidean norm as a mapping function $\hat{x}_t = \sqrt{x_t^T x_t}$. This is equivalent to computing the distances to the origin for 3D position measurements.

This method also works for incremental rotation measurements (as used for gyroscope integration, again see [PK08]). A static projection can be used if the measurements are represented as incremental rotation measurements. Alternatively, if the measurements are represented as a 3-element rotation velocity, the reduction can be obtained by computing the norm.

Adaptive computations A more advanced method of dimensionality reduction incorporates the influence of all measurements in the current segment and adapts the projection vector accordingly. Similar to the static computation, the projection can be defined as $\hat{x}_t = w_t^T x_t$, where w_t is now dynamically computed for each segment. It is still constant for each segment, and typically the same projection will be used for different sensors.

One way to compute a projection vector for the measurements of the current segment is to transform them using principal component analysis (PCA) [Fod02]. The PCA determines the direction of the major movement for each single segment. These directions can then be normalized and used as projection vectors for the particular segments.

It is also possible to further smooth the projection behavior of this method, by calculating the projection vector as a moving average over a limited history of previous segments. This further increases the robustness against outliers while reducing the speed of adaption.

Reduction incorporating feedback Another improvement of the dimensionality reduction is to also consider the final significance of the relative lag estimation as determined using the particular projection. This is a kind of feedback situation, where the choice of the dimensionality reduction projection is optimized in order to maximize the significance of the final result. It is easy to see that this increases the computational cost of the reduction.

Another similar method, which represents a trade-off between the generality of this approach and the required computational cost, is to consider two different dimensionality

User	Relative lag	Std. deviation
1	32.4 ms	2.7 ms
2	32.1 ms	0.33 ms

Table 5.1: Results of calibration using physiological tremor

reductions in parallel. The time delay estimation is then performed on two different pairs of signals and the final decision between the two projections is delayed until the significance of both is known. This can be seen similar to the "power of two choices" principle in randomized computing.

Pathological sensor data While there are many possible variations, experience shows that in practice the time offset calibration is usually robust against the concrete choice of dimensionality reduction. The various possible pathological cases which render the dimensionality reduction ineffective and thus the temporal calibration useless is discussed shortly.

The simplest pathological case is zero movement, with no events happening in reality. In this case the signals of the two sensors consist only of the sensor-noise. Since the sensors are assumed to be operating independently, the resulting signals are consequently uncorrelated. This obviously leads to useless results and the inability to determine the relative lag. Thus, mechanisms have to be included that recognize the complete absence of movement and do not process the data in this case.

Apart from this trivial instance, each dimensionality reduction method may exhibit specific cases, which can indubitably be constructed. For example the simple projection of the 3D sensor positions onto one of the primary axes obviously produces unsuitable signals for sensor movements orthogonal to that axis.

Yet, these cases are actually rare in practice, since they require very precise movements. To show this, an experiment was conducted with two participants, and the relative lag between an infrared tracking system and a coordinate measurement machine (CMM) was calculated using only these sensor movements. The task for the participants was to try to keep the sensors as steady as possible, without actually resting their arms on the table. This approximates the trivial pathological case of zero movement, yet in a scenario with human interaction, as generally expected in AR applications.

Due to the physiological (normal) tremor of the hands of the participants, the signal-to-noise ratio of these measurements was already sufficient to successfully calculate the relative lag. The frequency of the physiological tremor (about 6 Hz to 12 Hz, see [Lip71]) was well within the temporal resolution of both tracking systems. This can be seen in table 5.1. On the other hand, the signal of the reference experiment, where the sensors were fixed with a vice, produced no suitable results, as was expected.

Thus, while pathological cases for the individual cases do exist, these cases are scarcely

encountered in practice mostly due to the nature of human interaction at the limitations of human stability. In addition the use of adaptive projections or the power of choice between different projections can help to make such cases even more unlikely.

Only in scenarios where the sensors are mounted on computer controlled actuators or robots, special care has to be taken to tune the dimensionality reduction to the data at hand.

This experiment also raises the question on the general influences of human factors on the various parameters of sensor data and the resulting quality of the temporal calibration. Evaluations and results regarding this subject will be researched in future work.

5.3.6 Interpolation

A common assumption is that each signal is represented as measurements which are sampled as equidistant points in time, as mentioned when discussing the sample rate of the sensors. In practice the sample rate of a single sensor, as seen from the tracking framework, may not be constant. In figure 5.3 the sample rate of the Faro CMM is depicted and the variance of the sample rate, also called jitter, is clearly visible.

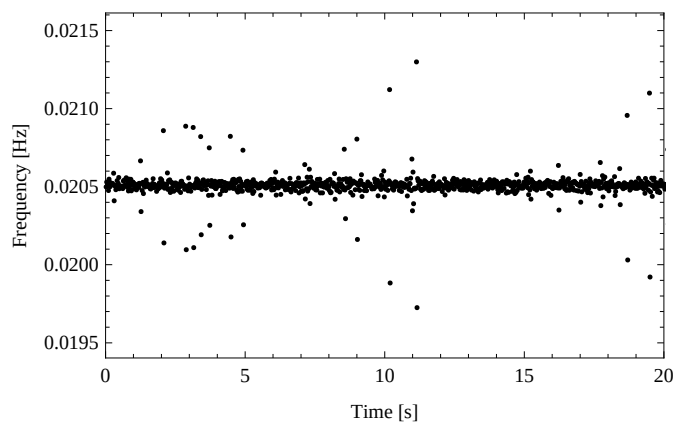


Figure 5.3: Sample frequency of the Faro CMM ($\mu = 48.8\,\mathrm{Hz}$, $\sigma = 0.53\,\mathrm{Hz}$)

Furthermore, the relationship between the sample points of two signals produced independently by two sensors is generally not known. Thus, one sample point of the first signal does not directly correspond to one sample point of the second signal. Besides, while the individual fluctuations in the update rates can usually simply be handled by assigning timestamps, these fluctuations complicate the computation of similarity measures and thus the comparability of the signals in general. It is therefore necessary to create a common basis of sample points for both signals.

Using the originally assigned timestamps for each measurement of either sensor, both

signals are interpolated using linear interpolation between the individual sampling points, resulting in continuous signals.

This results in direct one-to-one correspondences of samples in both signals by sampling from both signals at common timestamps. This is the foundation for the similarity computation (see below).

Note that in practice it suffices to perform the actual interpolation on only one signal. The sampling points for the similarity computation can conveniently be chosen to coincide with the original sampling points of one of the two sensors. If the sampling rates of the sensors differ, it is beneficial to interpolate the signal with the higher sampling rate to minimize interpolation errors.

Also note that it is generally preferable to interpolate the one dimensional projected signal as opposed to the higher dimensional tracking data. First, the interpolation at this stage is more straightforward and more clearly defined. This may be harder for general tracking data (for example there are different interpretations of quaternion interpolation or more generally interpolation of rotations). Second, the computational complexity is also less due to the reduced number of dimensions.

5.3.7 Time Delay Estimation (TDE)

These preprocessing steps result in two one-dimensional signals $\hat{X} = \{\hat{x}_t = f(x_t) : t \in T'\}$ and $\hat{Y} = \{\hat{y}_t = f(y_t) : t \in T'\}$ which have been interpolated and can be assumed to be continuous on the time domain T'.

The actual estimation of the relative lag of these two signals can be performed by a method also known as the time delay estimation (TDE). This method is well known and understood in signal processing and is used for applications such as RADAR or SONAR (for example [Car81]). Generally, the aim of this method is to estimate the temporal offset of a specific pattern contained in a generally noisy signal. The pattern usually is also transmitted by the system and later received as a reflection.

The application of the time delay estimation to tracking data differs from the standard application in that no template signal is actively transmitted as one instance of the pattern searched for in the second channel. Rather the signal of one sensor is treated as the pattern, which will be searched for in the other sensor signal. The offset between these two pattern instances is then the relative lag between the sensors.

The general procedure of the time delay estimation keeps one signal fixed and shifts the second in time relative to the first. For each possible time shift a similarity measure is computed. Finally the timeshift which maximizes this measure is the time delay estimate.

Cross-Correlation One of the earliest and still most important similarity measures used for such setups is the normalized cross-correlation. Especially in application areas which have to deal with reverberation or multipath effects, other methods which compensate

for such channel effects are employed (such as the generalized cross-correlation [KC76]). Since in this application no such "propagation channel" of the signal exists, such effects can be ignored. The computation can also be optimized by methods presented in [JS93] and others.

The normalized correlation [Wei98] coefficient (also called Pearsons' correlation), is defined as

$$\rho_{X,Y} := \frac{E[XY]}{E[X]E[Y]}.$$

Calculating the similarity between the signals while shifting one in time results in a graph as exemplified in figure 5.4.

Resolution of the TDE The resolution is mainly determined by the step-size of the timeshifts performed. Also the determination of the maximum value becomes increasingly less well-defined with decreasing step-size. Furthermore the computation time obviously increases with decreasing step size, making very small steps infeasible.

A common approach to increase the localization of the time delay estimate beyond the resolution of the calculated step-size is to fit a parabola to the similarity measurements and calculate the delay estimation as the vertex of this parabola [BH81].

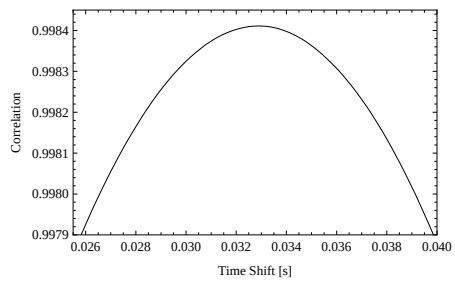


Figure 5.4: Graph of similarity (correlation) vs. timeshift

Both the maximum correlation value or the rate of increase of the fitted parabola can be used as indicators on the significance of the time delay estimate on the individual segment.

Other approaches could adaptively refine the area around the maximum of the TDE, but as already mentioned, the discrimination of the maximum value is reduced with too fine steps. Experiments suggest 1 ms as both a feasible and useful resolution.

5.3.8 Aggregation

After the time delay estimation has determined the relative lag of the two signals on one segment, multiple segments can be aggregated to identify meaningless results, reject outliers or perform smoothing of the lag calibration. Since the aim is to be able to

perform calibration and correction at run-time, the combination of segments needs to be performed adaptively as well.

Suitable approaches are the simple moving average, the weighted moving average using the significance parameters as discussed above as weights or the moving median.

5.4 Evaluation

To validate the method described above a series of experiments were conducted involving different combinations of sensors. An example for such an experiment can be seen in figure 5.5.

In the following, a description of the concrete hardware setup used as well as the individual experiments are given. Finally the effectiveness of this approach is demonstrated by an evaluation of registration errors in both unsynchronized and synchronized sensor fusion. Naturally, the Ubitrack system was used for all evaluations.

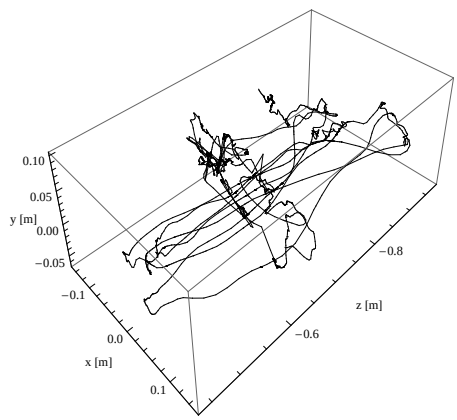


Figure 5.5: Sample movement used for calibration of the relative latency.

5.4.1 Hardware setup

The following list summarizes the available hardware used for the experiments.

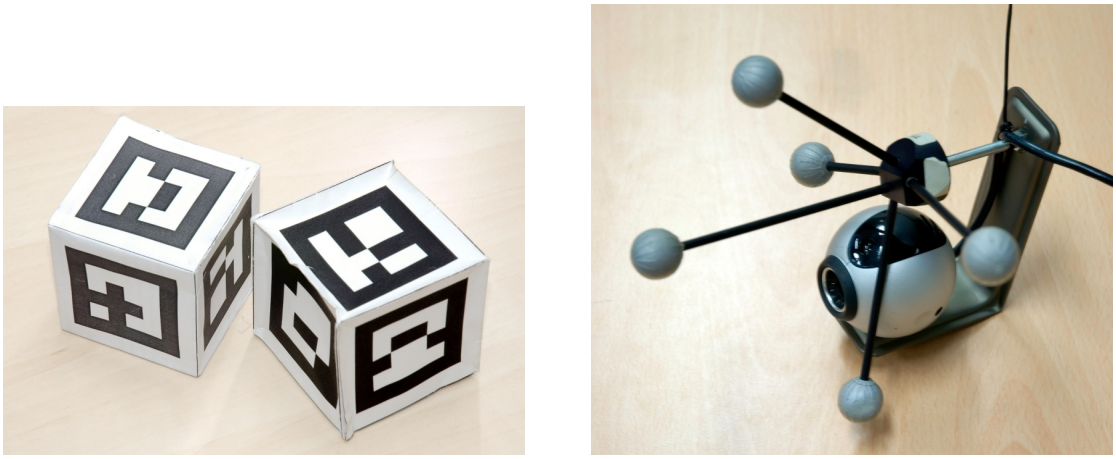
- The *A.R.T. system* is an optical, infrared outside-in tracking system based on retro-reflective ball markers. Either 6DoF poses for rigid marker constellations or 3DoF positions for single balls can be obtained. The sample tracking setup can be seen in figure 5.6a.

- The *Faro Fusion* coordinate measurement machine (CMM) is a high precision measurement device. It produces 6DoF measurements of the position and orientation of the tip of the arm. A picture of the used Faro CMM can be seen in figure 5.6c.

- An inside-out *optical square marker tracker* which can track the 6DoF pose of a printed square-marker pattern using an off-the-shelf webcam. This combination can be seen in figure 5.7.
- The *Xsens MT9* inertial sensor can track its orientation as a 3DoF rotation without reference system. Figure 5.6b shows the sensor used.

(a) (b) (c)

Figure 5.6: Sensors used for evaluation; (a)A.R.T. infrared tracker; (b)Xsens inertial tracker; (c)Faro CMM arm



(a) (b)

Figure 5.7: Multiple tracked square markers and webcam

5.4.2 Cases

This section describes the concrete experimental setups used.

Setup 1 – A.R.T. vs. Square-Marker In the first setup the A.R.T. outside-in tracking system was compared with the inside-out square marker tracker. For this an A.R.T. marker body was mounted on top of a USB camera and a square marker was fixed relative to the A.R.T. cameras. The spatial relationship graph describing this setup can be seen in figure 5.8.

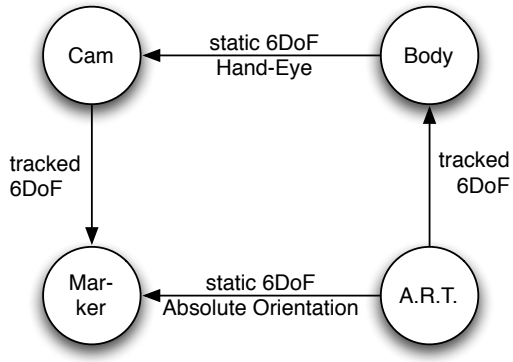


Figure 5.8: SRG describing the spatial relations of A.R.T. and square-marker tracker

The poses describing the relationship between the square marker and the A.R.T. cameras as well as the relationship between the USB camera and the mounted A.R.T. body, are static for the duration of the experiment and have both been calibrated in advance. While the pose of the square marker in the A.R.T. system was calibrated by solving the absolute orientation problem [Hor87], the relationship between the USB camera and the A.R.T. body was calibrated by Hand-Eye-Calibration (e.g. [Dan99]). Using these calibrations it is possible to transform the 6DoF poses of the A.R.T. system into the USB camera system.

Setup 2 – A.R.T. vs. Faro CMM In the second setup the A.R.T. outside-in tracking system was compared with the Faro CMM. The setup consisted of the Faro CMM inside the tracking area of the A.R.T. system, where a single A.R.T. marker ball was mounted as the tip of the Faro arm. The A.R.T. system in this scenario is only used for tracking the 3DoF position of the marker ball (as can be seen in figure 5.6c). The SRG describing this setup is shown in figure 5.9.

The tip of the Far arm was calibrated into the center of the A.R.T. marker ball which is also the point tracked by the A.R.T. system. Thus, this relationship does not show up as an explicit static edge in the SRG. The relationship between the Faro base and the A.R.T. cameras needs to be calibrated in order to transform the A.R.T. data into the Faro system. This calibration was again done by solving the corresponding absolute orientation problem.

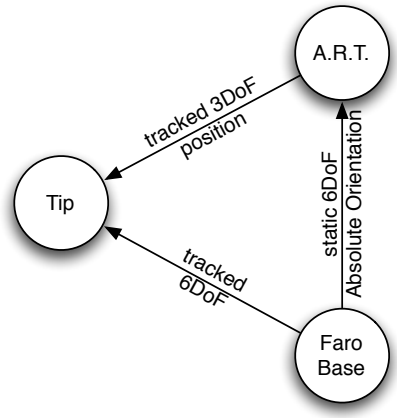


Figure 5.9: SRG describing the spatial relations of A.R.T. and Faro CMM

In this setup one of the trackers only delivers 3DoF position information and thus only the position can be used for aligning the sensor data.

Setup 3 – Faro CMM vs. Square-Marker This setup is similar to the combination of the A.R.T. system with the square marker tracker. In this case the USB camera for marker tracking was mounted rigidly to the Faro head and a square marker was fixed relative to the base of the Faro arm. The spatial relationship graph (SRG) describing this setup is omitted since it is similar to the SRG as seen in figure 5.8, with the Nodes "A.R.T." and "Body" replaced by "Faro Base" and "Tip" respectively. Also the calibration procedure for this setup was similar, featuring an absolute orientation and a Hand-Eye-Calibration.

Setup 4 – Faro CMM vs. Xsens The Faro CMM arm was also compared to the Xsens inertial tracker. The setup consisted of the Xsens rigidly attached to the tip of the Faro arm. Thus, the rotational data was available from both the CMM as well as the inertial tracker. The SRG for this setup is shown in figure 5.10.

The two relationships between the Faro tip and the mounted Xsens and between the Faro base and the Xsens reference frame are static and can be calibrated. Again both relationships were calibrated using Hand-Eye-Calibration.

Setup 5 – Transitivity Another test for the validity of the temporal calibration method is to examine the consistency of setups involving more than two sensors. When combining more than one pair of sensors the relative lag between the individual pairs has to obey

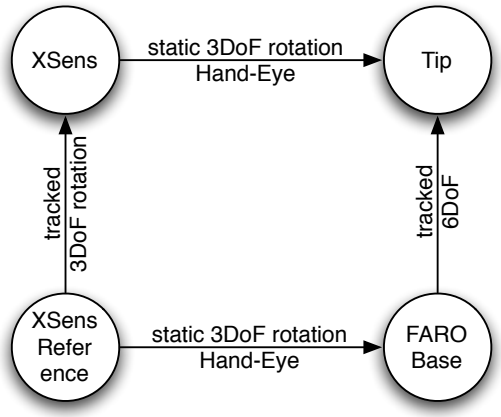


Figure 5.10: SRG describing the spatial relations of Faro CMM and Xsens gyroscope

transitivity. This is visualized in figure 5.11 for the combination of the setups as described above.

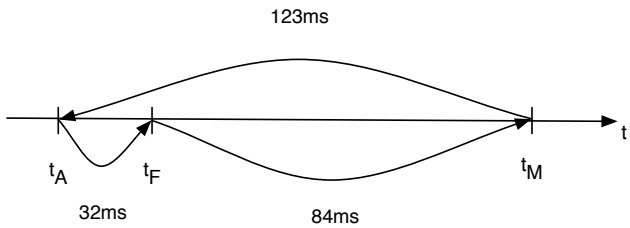


Figure 5.11: Transitivity of pairwise sensor offsets

Evaluation Results The results of the temporal calibrations are summarized in Table 5.2. The relatively large standard deviations in setups 1 and 3 stem mostly from pronounced temporal instabilities of the square marker tracker. This is especially evident when compared to the very precise results obtained for setup 2.

Further results When trying to calibrate an older MT9 Xsens inertial tracker to other tracking devices, no consistent calibration value could be determined. The lag between the inertial tracker and other sensors varied within short time frames, which hints that the Xsens clock has become unstable. Thus no result for Setup 4 is presented in table 5.2, but this odd effect is consistent with the observations in [ASB+04].

Setup	Mean	Std. Dev.
A.R.T. vs. Square Marker	123 ms	65 ms
A.R.T. vs. Faro	32 ms	0.1 ms
Faro vs. Square Marker	84 ms	30 ms

Table 5.2: Results of temporal calibrations

Method	Mean	Std. Dev.
(a) High SNR experiment		
Geometrical	32 ms	0.1 ms
Correlation	32 ms	0.1 ms
(b) Low SNR experiment		
Geometrical	35 ms	39 ms
Correlation	32 ms	0.3 ms

Table 5.3: Comparison with geometric error minimization

In the spirit of Ubiquitous tracking setups also tracking scenarios with sensors connected to distributed machines which communicate via LAN were evaluated. In this case also successful calibrations could be performed, although distribution to multiple hosts caused additional lag of up to 100ms. While the introduction of network synchronization protocols could reduce this problem to the temporal alignment problem on one machine, this approach again does in general not scale to ubiquitous tracking scenarios.

5.4.3 Comparison with geometric error minimization

To further validate the correctness of this approach the resulting relative lag of two sensors was compared with the time offset which minimizes the overall geometric error between the sensors. This is similar to the static calibration approach used in [LBMN09, SS04].

As can be seen in table 5.3(a) both methods yield comparable results. This has been expected since in general the correct relative lag also reduces the error between the registered coordinate frames of the sensors.

What is of further interest is that this experiment was performed using sensor data with significant SNR.

In cases with low SNR, the time delay estimation exhibits more robust behavior than the geometric minimization. The experiment was repeated with the data from the physiological hand tremor experiment (see section 5.3.5). Table 5.3(b) shows the results. This illustrates that under these circumstances, the result of the geometric optimization still falls in the same range as before but with massively increased uncertainty. On the other hand, the correlation-based time delay estimation performs only slightly worse

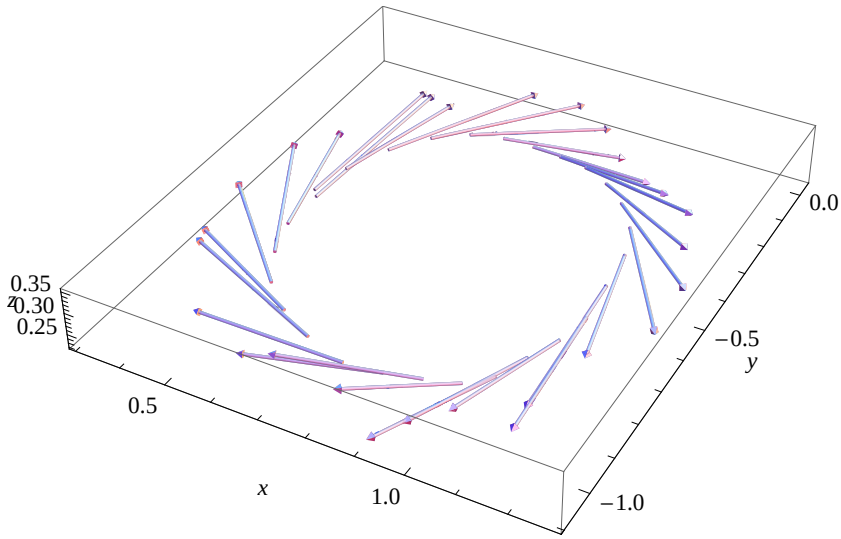
than before.

5.4.4 Error reduction

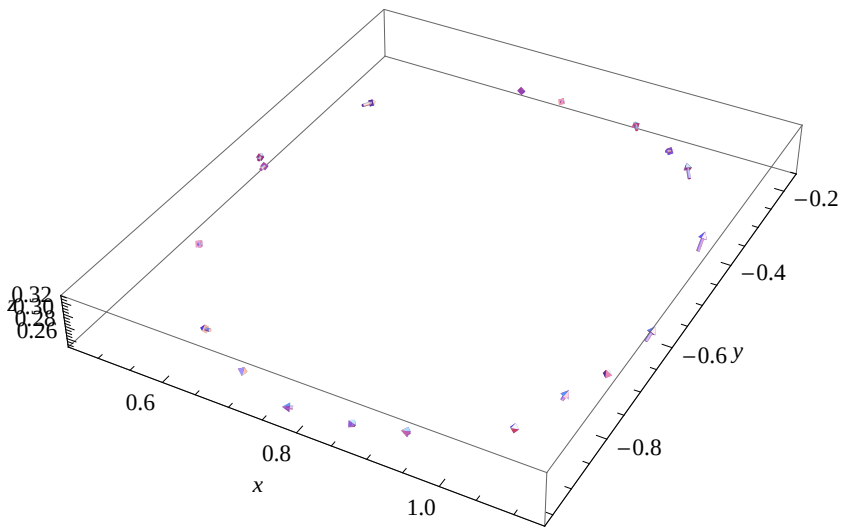
To illustrate the effectiveness of the temporal alignment, the resulting spatial registration error between two different trackers in both the unsynchronized and the synchronized case was analyzed. The same setup from the A.R.T. vs. Faro CMM case is being used for this analysis (Setup 2).

For this data set the tip of the Faro CMM was moved in a simple circle with moderate speed of about $1\,\mathrm{m\,s^{-1}}$ (determined afterwards). The 3DoF position of the tip was recorded both by the Faro system and by the A.R.T. system, which was additionally transformed into the Faro coordinate frame. Figure 5.12a shows the error vector between measurements from the A.R.T. system and corresponding points measured by the Faro system during the movement. The root mean square (RMS) error in this case is 32.1 mm. From the direction of the vectors the movement of the marker ball is clearly visible as a systematic misregistration. This indicates a distinctive lag between the two sensor systems.

The temporal offset in this experiment was determined to be 32 ms with 0.1 ms standard deviation calculated over all segments. Figure 5.12b shows the same error vectors after the timestamps were corrected according to the determined calibration value. In this plot the direction of the error vectors no longer corresponds to the direction of the movement and the RMS has been reduced to 1.6 mm. The remaining errors mostly stem from calibration errors and sensor noise.



(a)



(b)

Figure 5.12: Error vector between measurements from A.R.T. and the Faro system during movement; (a) without temporal alignment; (b) with temporal alignment; (magnification factor 10)

6 Radio Localization

Due to highly dynamic and unpredictable availability of devices used for parasitic tracking, methods need to adapt to the various kinds of infrastructure in the chosen environment.

Especially the methods to access and sense the properties of the environment, which are to be exploited, require sufficient understanding of these properties and suitable technologies.

Wireless communication infrastructure is a prime example of a widely available kind of infrastructure that could be appropriate to parasitic localization methods. At the same time, wireless communication requires understanding to enable somebody to identify and exploit the associated relevant parameters. This chapter will introduce the necessary concepts to give a basic understanding of wireless radio communications.

Since localization-unrelated infrastructure is targeted in parasitic tracking, no assumptions on the content of the radio communication signals can be made. This rules out approaches which require specific content to be transmitted such as for example clock value transmissions, as used in the GPS method. Approaches which use dedicated radio hardware infrastructure for localization are not considered either.

6.1 Introduction to radio terminology

History of wireless communication The history of radio communications can be traced back to James Clerk Maxwell's publication "On Physical Lines of Force" in 1861. Based on the work of Michael Faraday, André-Marie Ampère and Carl-Friedrich Gauß, Maxwell presented a system of equations to the Royal Society in 1864. These equations comprehensively describe the interaction between electrical and magnetic fields as well as the interaction with matter. Furthermore Maxwell predicted that interacting electrical and magnetic fields can travel through empty space at the speed of light. The coincidence of these two velocities lead to the hypothesis that light and electromagnetic fields are a similar type of radiation.

While Maxwell predicted the theoretical existence of electromagnetic radiation, it took more than 20 years until a successful experimental validation was possible. In 1886 Hertz was able to experimentally verify the existence and the properties of electromagnetic radiation as predicted by using a spark-gap transmitter and a wire antenna as sending equipment and a simple dipole antenna as reception equipment. Amongst other exper-

iments, he measured the spatial distribution of both electrical and magnetic fields on a standing wave.

After the experimental validation, applications of radio waves and the ensuing commercialization was pioneered by Guglielmo Marconi. In 1895 he was able to transmit a message via radio waves across a distance of 1.5 km, mostly by combing existing technology. Marconi approached various interested parties and was able to start commercialization in early 1900. One of the first successful applications concerned naval communication between ships and between ships and shore. In 1903 the first transatlantic message was transmitted, also using Marconi's equipment.

Since these days, wireless communication has seen an explosive growth, leading to the many different kinds of simultaneous transmissions and communication modes that are in use today.

Basic introduction Radio waves, as used for digital wireless communications, form a subset of the physical phenomenon of electromagnetic radiation. The most fundamental derivation and the most complete understanding of the principles of electromagnetic (EM) waves or electromagnetic radiation can be obtained by directly studying Maxwell's system of equations. Such an introduction can be found for example in [FLS63], whereas more hands-on information on the construction and operation of radio devices can for example be found in [Wil07]. In the following section, only the most basic properties will be presented without derivation.

As the name suggests, an electromagnetic wave arises from the interaction between an electric and a magnetic field. The electric field is associated with an electric voltage across a certain distance (e.g. capacitor plates or conductor) and describes the force of a charged particle present inside the field. Similarly, a magnetic field is associated with an electric current flowing through a conductor. Around the current-carrying conductor a concentric magnetic eddy field is induced. In a wound coil, the magnetic fields of the neighboring turns overlap in such a way that a homogeneous magnetic field is generated inside the coil.

A further important interaction is that every alternating or changing electrical field induces a related magnetic field and vice versa. Similar to the magnetic field surrounding the current-carrying conductor, the field induced by a changing electrical or magnetic field is a solenoid field (i.e. it contains no sources or sinks) and is perpendicular to the inducing field. This interaction between electrical and magnetic fields enables the existence of electromagnetic waves. EM waves are linked waves of electrical and magnetic energy. The induction of a magnetic field from an electrical field or vice versa is not the property of any matter, but of the field itself. Electromagnetic waves thus can travel through empty space.

Simple examination of wave characteristics and field superposition lead to two further basic properties of EM waves. The oscillation planes of electrical and magnetic fields in

a traveling electromagnetic wave are always perpendicular to each other as well as to the direction of travel. The electrical and the magnetic wave component of an electromagnetic wave are in phase to each other, meaning that the zero-crossings and the extrema of both fields align during propagation. This is visualized in figure 6.1.

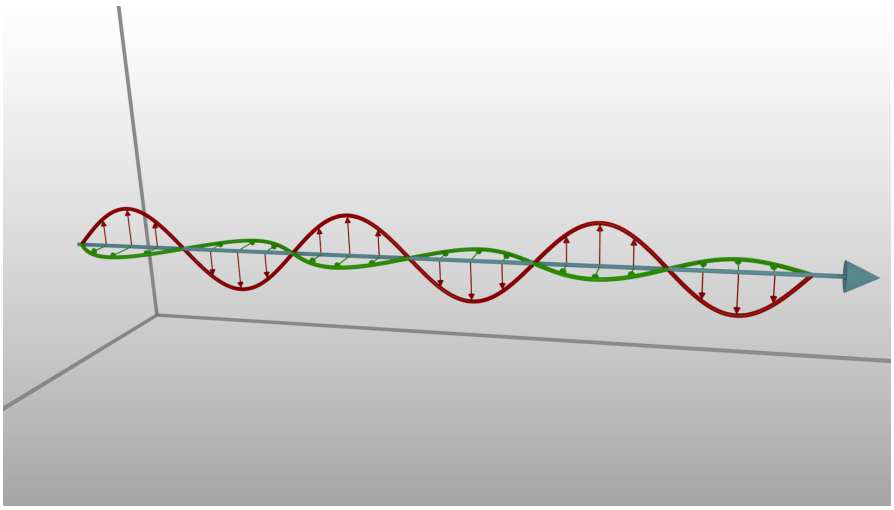


Figure 6.1: Wave propagation

The orientation of the plane in which the electrical field of an electromagnetic wave oscillates is called the polarization of the EM wave. The most common type of polarization is *linear polarization*, meaning that the plane of the electromagnetic field stays constant. Examples are horizontal (the electromagnetic field is parallel to the surface of the earth) or vertical polarization (the electromagnetic field is vertically oriented). The plane of the electrical field does not have to be constant during propagation, but may rotate around the direction of travel. In such a case, the EM wave is called circularly or in the general case elliptically polarized.

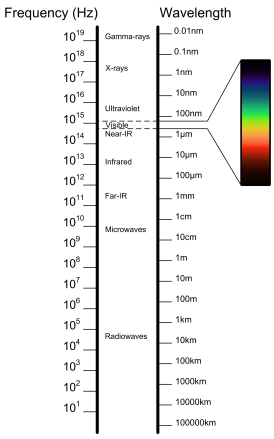
Electromagnetic waves may also interact with matter, in the same way that electrical or magnetic fields interact with matter. An EM wave may, for example, induce current or voltage in a metallic or, more generally, in any conductive material. Such an induced current or voltage may itself result again in radiation of a secondary EM wave. This can be perceived, depending on the observed direction, as reflection or refraction. Furthermore, due to its wave nature, wave effects such as interference of two overlapping EM waves or diffraction are observable. While in free space EM waves always travel in a straight line, the existence of obstacles may deflect the path of the wave or lead to phenomena such as shadowing or multi-path propagation.

A device designed to either stimulate a specific electromagnetic wave or to sense such a wave is called an antenna. In general, the polarization of the EM wave and the receiving

antenna have to agree to maximize reception.

Electromagnetic spectrum Another important characteristic of any electromagnetic wave is the frequency of its oscillation. As mentioned earlier, the electrical and magnetic components of an EM wave are always in phase to each other and thus the frequency of both components is equal. This frequency of the electromagnetic wave is independent of its propagation speed. An equivalent characterization commonly encountered is the specification of the wavelength of the EM wave. This quantity specifies the spatial distance traveled by the wave during one complete oscillation of one of its components. The wavelength λ is related to the frequency f by $\lambda = c/f$, where c is the propagation speed of the wave and in the case of EM waves, the speed of light.

Electromagnetic waves are unique in the sense that the possible frequencies span at least 19 orders of magnitude. Depending on the frequency, the EM radiation exhibits various different characteristics. Thus EM radiation is categorized into different classes according to different ranges of frequencies. This is usually called the electromagnetic spectrum.



Designation	Frequency	Name
EHF	30GHz – 300GHz	Extremely high frequency
SHF	3GHz – 30GHz	Super high frequency
UHF	300MHz – 3GHz	Ultra high frequency
VHF	30MHz – 300MHz	Very high frequency
HF	3MHz – 30MHz	High frequency
MF	300kHz – 3MHz	Medium frequency
LF	30kHz – 300kHz	Low frequency
VLF	3kHz – 30kHz	Very low frequency
ULF	300Hz – 3kHz	Ultra low frequency
SLF	30Hz – 300Hz	Super low frequency
ELF	3Hz – 30Hz	Extremely low frequency

Figure 6.2: Electromagnetic spectrum

The range commonly called radio waves actually ranges from a fraction of one hertz up to 300GHz. It is further divided into different ranges, mostly due to the different propagation characteristics concerning atmospheric influences and due to the corresponding applications.

Due to the usually macroscopic wavelengths of radio waves, wave mechanical phenomena such as interference are more readily observable than with radiation of much higher frequency, such as visible light.

Wavelengths in the range of millimeters to nanometers contain the range of visible and invisible light, which is located in frequency directly above radio waves. The class of

microwaves blends the boundary between radio waves and far-infrared light. In frequency even above the light portion of the spectrum are different types of radiation such as X-rays or Gamma-rays. These types of radiation are usually associated with very good penetration properties hence the medical applications of X-rays, for example.

6.2 Exploitable Parameters

In this section two additional characteristics of electromagnetic radiation will be discussed. These are the most promising parameters as seen from a parasitic tracking point of view. For both of these parameters a distinct relation between measurable physical quantity and the distance between sender and receiver can be established. The general strategy is to estimate the distances between a mobile user and anchor points in the environment, represented by the communication infrastructure.

6.2.1 Time measurements

As established by Maxwell, electromagnetic waves travel at the speed of light. While this speed is extremely fast, it is nevertheless finite. Thus it takes finite, and measurable time for a wireless signal to travel from the transmitting to the receiving equipment. The speed of light in vacuum has been established as $299\,792\,458\,\mathrm{m\,s^{-1}}$ [Boy03] and, subsequently, the definition of the meter was tied to the speed of light. Currently, the best method to estimate this speed is by laser interferometry using lasers of precisely known frequencies.

Since the speed of light in free space is constant, the time required for a wireless signal to travel between sender and receiver is directly proportional to the distance between sender and receiver. To measure this delay, it is important to be able to reference a robust feature of the transmitted signal. As a simplified example, consider an on-off-keying of a carrier wave. By defining the off-to-on transition as the start of the signal, the same portion of the signal can be referenced between sender and receiver and the transmission delay can be estimated. In reality, the start of such a signal is not that well defined, mostly due to the limited bandwidth of any radio transmission, thus other more suitable methods are required. Systematic delays in either transmitter or receiver also have to be calibrated beforehand.

The measurement of the time required for a signal to reach its destination is also called time-of-flight (TOF) measurement (see figure 6.3). In this diagram three transmitters ($TX_{1,\ldots,3}$) send signals to one receiver (RX_1) and three seperate time delay measurements ($t_{1,\ldots,3}$) can be derived. These are necessary to unambiguously derive a three dimensional position esitmate of the receiver.

Although the speed of light is, as mentioned above, very fast, sufficiently fast time measurement hardware exists. To achieve a spatial resolution of about 1 cm, a temporal resolution of better than 3 ps is necessary. Current high-performance time-to-digital

converters (TDC) achieve resolutions of about 10 ps, thus resulting in a maximum spatial resolution of about 3 cm. Such TDCs usually operate with a precise, periodically triggered delay-line with taps, which feed a synchronous counter. Note that to actually measure the time delay between sender and receiver, some kind of synchronization between these two is necessary. Otherwise the start of measurement is not defined at the receiver.

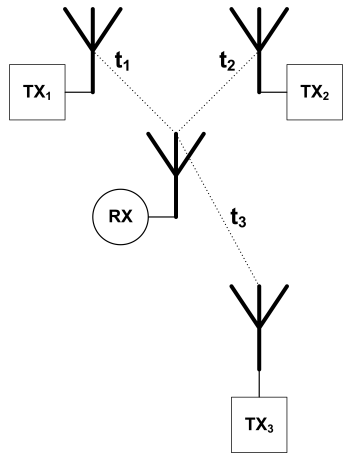


Figure 6.3: Time of flight setup

Instead of directly measuring the delay between sender and receiver, the delay between multiple, synchronized senders and a single receiver or symmetrically between a sender and multiple, synchronized receivers can be measured. This is called the time difference of arrival (TDOA) measurement. By using at least two distinct and synchronized transmitters or receivers with known spatial relationship, the time difference of the arrival of transmitted signals can be estimated. Increasing the number to at least four entities, estimation of a 3D spatial relationship is possible (see figure 6.4). Different schemes, such as using synchronized transmitters or synchronous reception of a single transmitter via multiple antennas are possible.

From this discussion follows that pure TOF can only be applied in setups where a synchronization between sender and receiver is possible. This can be achieved either by hardware (synchronization cables) or by the data payload in the packet. The GPS system, for example, which is also based on estimating the distance of the handset to a number of satellites, includes the current time of the satellite clock in the transmitted packets. The receiver compares the local receiver time to transmission time of the packet and can compute the distance (which still contains errors) to the satellite. As opposed to the satellite clocks, the receiver clocks usually are of moderate quality and stability. Thus, to account for receiver clock errors, the data from an additional satellite is used,

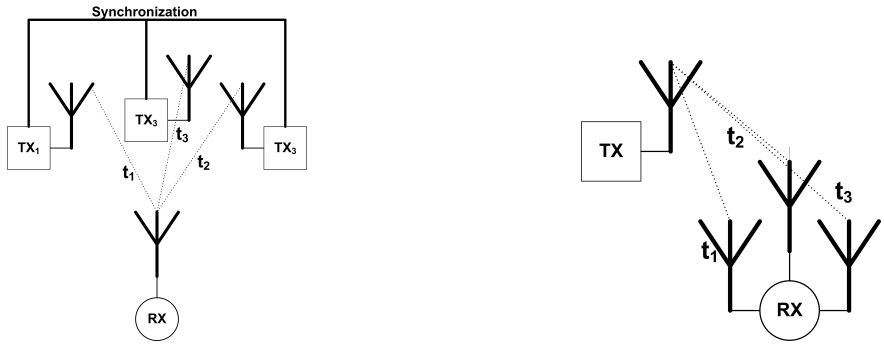


Figure 6.4: Time difference of arrival setup

thus requiring at least 4 satellites to achieve 3D localization.

Such setups do not comply to the assumptions made for parasitic tracking setups. For parasitic tracking, neither a hardware synchronization between user and infrastructure nor specially encoded timing information from the infrastructure can be assumed. Thus the benefit of such setups in parasitic tracking scenarios is limited.

For a similar reason time-difference-of-arrival setups using multiple transmitters do not seem to be applicable to parasitic tracking scenarios. Such a system needs precise information about the spatial setup of the infrastructure as well as coordination mechanisms that synchronize the transmissions from all senders. Again this cannot be guaranteed in a parasitic tracking scenario.

Finally, a setup consisting of a single transmitter and multiple, synchronized receivers might be interesting in the future for parasitic tracking setups. No evaluation has been undertaken so far, mostly due to the relatively high hardware complexity associated with this scenario.

A selection of other setups, as well as other types of waves have been evaluated using a time-digital-converter by [Mac10]. This includes development of a suitable framework and its integration into the Ubitrack library.

6.2.2 Signal power

The second parameter suitable to estimate the distance between sender and receiver is based on the signal power of the electromagnetic wave.

Both the transmitter and the receiver use antennas to convert between electrical and radiated power. This electrical signal is either supplied by the transmitter side or it is measured at the receiver side. The amplitude of this measured signal is proportional to the square root of the signal power that is received by the antenna. The power at the receiver is related to the power of the transmitter, but it is influenced by many factors.

Figure 6.5 shows an overview of the different influences encountered during a single

transmission and reception. Each node represents either an attenuation, reducing the power of the signal or an amplification which increases the power.

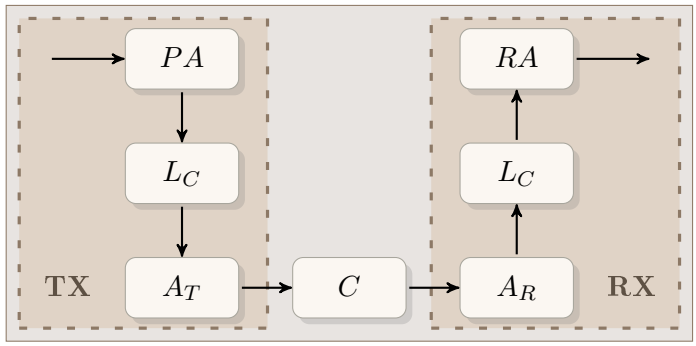


Figure 6.5: Signal power influences

PA The last stage of a transmitting equipment is the power amplifier. The PA delivers the final amount of power that will drive the antenna and that should be converted into radiated power at the antenna. It is therefore modeled as an amplification stage.

LC The transmission lines connecting the power amplifier to the transmitting antenna and respectively the receiving antenna to the receiver usually exhibit some amount of attenuation. The signal power arriving at the remote end of a transmission line is therefore less than the amount provided. These are also sometimes called "cable losses".

A_T/A_R An ideal antenna which exhibits uniform behavior independent of direction is called isotropic and is modeled to have no gain or attenuation in any direction. Any other antenna (and especially real-world antennas) do not exhibit a completely symmetric radiation pattern. This can be seen as deforming the ideal spherical pattern so that certain directions are favored and receive amplification whereas radiation in other directions is attenuated. The amount of amplification (or attenuation) depends on the spatial relationship between sender and receiver. The values considered in this diagram represent the amount of amplification as encountered in the specific path between the depicted sender and transmitter.

C The signal power is attenuated by the propagation between sending and receiving antenna. This is modelled as the channel- or path-loss and will be discussed below. This attenuation can generally be viewed as distance dependant.

RA Similar to the power amplifier (PA) as the last stage of the transmitter, the first stage of the receiver is (usually) also an amplifier. The signal which is picked up

by the receiving antenna is usually too weak to be further processed or measured directly. Thus it is amplified, for example by a low-noise amplifier (LNA).

Note that this discussion is simplified in that it assumes only continuous wave signals and disregards signal reflection at various stages. This scheme is also referred to as the *link budget* of the transmission.

In this diagramm the PA, LC and A_T make up the transmitter. In a parasitic tracking scenario, these parameters are thus assumed to be completely uncontrollable by the user, but can be assumed to be constant, at least during short periods of time. The nodes A_R, LC and RA make up the receiver, which can be assumed to be under control of the user. The channel C in between the sender and the receiver is the element which is influenced the most by its distance. Thus the estimation of the channel-loss can reveal relevant information about the distance between transmitter and receiver.

For a better estimation of this channel-loss, different models for transmission channels are available. The most basic model assumes a spherical distribution of the transmitted power.

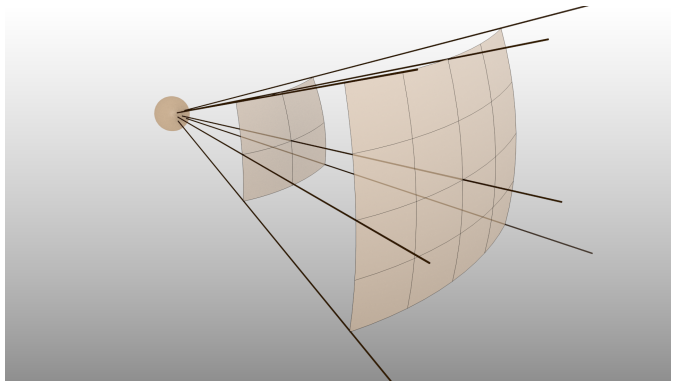


Figure 6.6: Spherical distribution of power density

Assuming that the transmission power, i.e. the power applied to the antenna, is constant, the integral power of the radiated wave at any distance is equal to the transmitted power. Since the surface of a sphere is increasing quadratically with distance, the power of the wave is increasingly spread out and thus the power density of a sphere segment of constant dimensions is decreasing accordingly (see figure 6.6). This is a common occurrence in physics (e.g. gravitational force behaves similarly) and is called the inverse-square-law. For the receiving power this implies that the power received at any antenna of constant size is decreasing quadratically with increasing distance. Note that this applies to the isotropic radiator. For antennas with directional gain, the sphere is accordingly deformed, but the general law nevertheless applies.

This leads to the most basic channel- or path-loss model, called the free-space path-loss or Friis transmission equation. It states that the path-loss is both proportional to the square of the distance and to the square of the frequency of the signal. Apart from the illustrated effect of the distance, also the frequency dependant aperture of the receiving antenna is modeled. The free-space path loss attenuation can be formulated as $FSPL = \left(\frac{4\pi d}{\lambda}\right)^2$. Note that this equation assumes that directional antenna gain is already accounted for.

Apart from this basic model, a number of different channel models which account for different effects have been proposed. By considering not only the direct propagation path but also a number of indirect paths, basic effects of reflections and interference can be accounted for. In the standard case of sender and receiver positioned at different heights above ground the most important indirect path is the reflection off the ground. Accounting for the direct path and the ground reflection leads to the 2-ray model. The attenuation in the 2-ray model is still proportional to the squared distance, but also adds terms based on transmitter and receiver height. Similarly the 10-ray model is used to approximate urban environments including reflections from buildings.

Besides these analytical models a number of empirically established models exist. The basis for a subclass of these models is the Okumura model. The Hata model further refines the Okumura model and adds correctional terms depending on the type of surrounding environment (urban, suburban or rural). Another model is the ITU model which models the channel loss in indoor environments and across office floors. Besides the differences of these models, they share the common fact that attenuation in any case is at least quadratic in the increase in distance (in the ITU model up to an exponent of 3.3). For a more in depth discussion of these various models see for example [Gol94] and [Gol05].

By estimating and modeling the channel loss it is possible to directly estimate the distance of the transmitter to the receiver. Due to the number of unknown factors which could distort these estimates, a more cautious approach only assumes that the received power decreases at least quadratically with increasing distance. Thus the distance between transmitter and receiver can be estimated, or at least a dependency of the received power to the distance can be assumed. The most conservative assumption in this context is that signals with lower received power have originated at least as far or farther away than those with greater power. Thus accurate estimation of the received signal power enables localization with respect to distance estimation using multiple anchors.

A second effect of increased channel loss is the general decrease of the channel fidelity, resulting in less link quality. This could for example be measured by the signal-to-noise ratio at the receiver or the bit error-rate (BER).

6.3 Software Defined Radio

As was established above, radio waves have several parameters which could be exploited in a parasitic tracking scenario. Nevertheless the possibility to access these parameters through a suitable interface for a parasitic tracking system is not guaranteed. This section will highlight a novel wireless communication technology which holds the promise to facilitate such access.

6.3.1 Different approaches

The traditional design for radio equipment usually includes a large portion of analog devices and linear circuits. This analog processing of radio signals using specialized hardware seems natural given the high frequencies and high data rates usually encountered. Nevertheless such a design has a number of drawbacks.

The radio equipment is inherently inflexible since the design of such a device has to incorporate the specific requirements of a concrete application. A traditional radio device is usually limited to a specific radio band and also to a specific application. This includes, for example, frequency generation and tracking (for example using phase-locked-loops) or modulation. Changing an application from one modulation method to another entails redesigning the radio hardware.

As a direct consequence the design and manufacturing of such radio hardware is usually expensive. The design of radio equipment consists of many engineering tasks requiring highly specialized knowledge and skills. Similarly the manufacturing of such hardware requires a comparatively high number of different and also specialized components. This again increases the manufacturing costs and also makes repair of such equipment difficult. Furthermore, the reusability of previous designs is limited and substantial modifications are usually necessary. Finally, as a more complex system is less comprehensible, it is also more difficult to asses correctness and reliability.

An example of a more cost effective digital solution to a problem previously exclusively solved with analog circuits is the generation of a variable frequency. The traditional circuit is called a voltage-controlled oscillator (VCO) and one common method to implement it is to control the frequency of the oscillation by the reverse voltage across a specialized diode call "varicap". On the other hand an arbitrary frequency and waveform can easily be synthesized by digital means using a processor with a phase accumulator, wave table and digital-analog-converter for synthesis. This is called a numeric-controlled oscillator (NCO). The stability of the NCO only depends on the performance of the processor and the stability of its clock, which is fixed.

This example is indicative of a novel design approach, which only recently became practical and since then gained momentum. It is known under the term "Software Defined Radio" (SDR). The first ideas of soft radios were developed in the late 1970s as classified projects for the defense sector and the first semi-public project was the "Software radio

proof-of-concept" laboratory of "E-Systems" (today Raytheon) in 1984. The concrete term "software defined radio" was coined by Joseph Mitola in his 1993 publication ([MI93, MIMJ99]). Thus, while the concept of SDR has already been known for almost 20 years only recent advances in processor technology and high speed sampling and synthesis made the concept feasible.

Very simplified, the approach states that the computer should be moved as close to the antenna as possible. Everything that can be done in software, should be done in software, preferably on the computer. This effectively reduces the special radio hardware to sampling (A/D- or analog-digital-conversion) and synthesis (D/A- or digital-analog-conversion) of the already modulated radio signals. The processing, using digital-signal-processing techniques, can then happen in a host or embedded computer. The following section will give a short overview of the components usually encountered in an SDR device.

6.3.2 SDR Architecture

The figure 6.7 below shows the basic architecture blocks that are usually contained in an SDR device. Each node represents some kind of transformation of the signal and will be shortly discussed below.

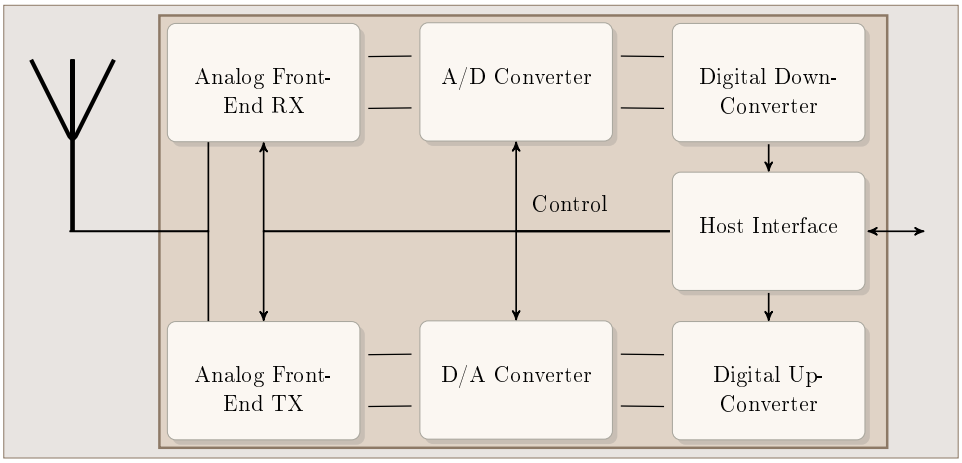


Figure 6.7: Software defined radio architecture

The paths for transmission and reception can be considered independently from each other, thus the description will focus first on the reception path and then on the transmission path.

Antenna The antenna is not directly part of the SDR device but is also necessary for correct operation. As mentioned above, the antenna converts between electrical and radiated power and vice-versa.

Analog Front-End RX To process the signals received by the antenna, it is first necessary to interface the usually weak signals by an analog front-end. Note that while this analog front-end needs to fulfill different tasks, many of the parameters of these tasks usually depend on the particular operating frequency range. Thus these analog front-ends are specific to their operating frequency range. Nevertheless newer front-ends may cover very large frequency ranges. Wide-band front-ends that handle the whole range from 50MHz to 5GHz are not uncommon.

As the first task in order to transfer a maximum amount of the very weak received signal to further processing stages, it is necessary to match the impedance of the receiving circuit to the antenna. This is even antenna specific, since different types of antennas have different feed impedances, though usually the antenna is not directly connected to the receiver. In that case, the matching rather matches against the impedance of the transmission line.

To further increase the power of the radio signal received by the antenna, additional amplification may be necessary before the signal can be further processed. For this amplification a low-noise amplifier (LNA), which exhibits linear amplification over a wide spectrum without shifting frequencies can be used. This is also called an antenna- or pre-amplifier.

Since the frequency of radio signals generally is still very high compared to the sampling rate, the frequency range of the signal has to be shifted down into a region which can be handled by the following digital circuit. The process to achieve such a shift is called tuning and down-conversion. Note that in an SDR device this analog tuning generally only needs to be coarse, since a larger portion of the spectrum is shifted downwards into the range which can be processed by the digital portion of the SDR. The signal at RF frequencies is usually called the passband signal whereas the down-converted signal is called the baseband signal.

A low-pass anti-alias filter further prevents mirrored or warped images of other signals interfering at the sampling process.

Analog-Digital-Conversion The main component of the reception path of a software defined radio is the analog-to-digital-converter (ADC). Generally speaking, this component translates an observed analog voltage into a digital code representing this voltage. SDR devices usually use a pair of high speed ADCs per receiving channel. Two of the most important characteristics of AD converters are the sample rate and the output resolution. Software-defined radio devices typically use very high speed converters with sample rates in the general range of multiple million samples per second. Typical output resolutions are 16, 14, 12 or 8 bits per sample.

Digital-Down-Conversion Since the sampling bandwidth of the A/D converter is usually larger than the bandwidth of the modulated signal, it may be desirable to further

down-convert the received signal and to reduce the data rate. This reduces the requirements on the host interface needed to communicate the sampled data. Due to the usually rough tuning resolution of the front-end, the center frequency of the desired signal is usually not yet at the center of the sampled spectrum. In other words, the digital down conversion can refine the coarse tuning of the front-end. Furthermore, reduction in sampling rate or sample resolution may be necessary in order to keep within the real-time communication and computation speed of the host computer. Thus it is possible to find a compromise between necessary host-communication speed, sample resolution and sampled bandwidth.

Host Interface After down-conversion and digitalization, the radio signal picked up by the antenna is available as a series of digital samples at baseband. These samples need to be transferred to the host computer for further computations and digital signal processing. A suitable interface for transmitting these data at real-time is necessary. Some choices are for example High-Speed USB, Gigabit-Ethernet or IDE. The result of the reception process, as well as the input to the transmission process, is a discretely sampled and quantized modulated signal at baseband.

For the transmission path, the function of the antenna and the host interface are unchanged and thus will not appear in the description below.

Digital-Up-Conversion Similar to the Digital-Down-Conversion in the receiver side, the communication between host computer and software-defined radio in general happens at a reduced rate as compared to the synthesis rate of the Digital-Analog-Converters. Thus it is necessary to digitally convert the sample rate arriving at the SDR to the rate as expected by the DACs, for example using interpolation. The fine tuning of the transmission frequency from the baseband samples, as produced by the host computer, to the passband frequency is done by the digital-up-converter, whereas the supplementary shift into the right RF band is performed by the analog front-end.

Digital-Analog-Conversion After adapting the rate of the samples to the synthesis rate, the analog signal is synthesized from the digital information. Similar to ADCs in the reception path, this conversion is done by a pair of Digital-to-Analog-Converters (DACs) per channel, which convert each sample into the corresponding analog voltage. The sample rates required and the sample resolution are also similar to the performance figures of the ADCs in the reception path.

Analog-Frontend TX After synthesis of the final radio signal which is to be transmitted, the signal is finally processed by an analog circuit. The analog front-end of the transmission path is also dependant on the concrete frequency range, similar to its reception path counterpart. Often the analog front-ends of both transmission and reception are combined into a single circuit, called a transceiver.

To reduce artifacts introduced by the digital synthesis process, a low-pass reconstruction-filter is often used to remove spurious high-frequent components. The signal at that point is at or near baseband and still needs to be up-converted to passband. This can be done by a tunable local oscillator and analog up-converter. Again, the tuning resolution only needs to be coarse, since the fine tuning is already taken care of by the digital up-converter.

The final components of the transmission path of an SDR are power-amplification which amplifies the signal to the desired power levels to supply the antenna as well as impedance matching to the antenna or transmission line.

Complex sampling The reason why for each transmission or reception channel of the SDR a pair of DACs respectively ADCs are required, is again because the software-defined radio in general operates at baseband. The discussion below will focus on sampling, but the same principles also apply to synthesis.

In general, the radio signals received are modulated signals with a carrier or center frequency ranging up to 5GHz, or even beyond. While it is unlikely that an analog-digital converter could operate at such high sample rates required to capture these oscillations directly, it would also be a waste of sampled bandwidth and consequently of processing power. Rather the sampling rate of the AD conversion has to conform to the bandwidth occupied by the modulated signal; that is the span from the minimal frequency to the maximal frequency associated with the signal or the range the signal allocates in the spectrum.

According to the sampling theorem, the maximal resolvable frequency f of a sampler with sample rate R is at most half the sample rate $f \leq R/2$. To shift the radio signal from its high frequency into the range of the AD-converter, the signal, as described for the analog front-end, is converted down to baseband, i.e. to a center frequency of 0Hz. Since the modulated signal occupies ranges of frequencies both above and below the center frequency, the bandwidth below the center is thus shifted into the negative range.

To represent these negative frequencies, it is necessary to move from real samples to complex samples containing both real and imaginary components. The real component of a complex sample is called the in-phase (or just I) part and the imaginary component the quadrature (or Q) part of the sample. Correspondingly the acquisition of complex samples requires two independent AD-converter channels, one for I samples and one for Q samples. Similarly the analog down-conversion has to produce both I- and Q-components of the down-converted signal. This can either be achieved with two independent down-converters, where the shift frequencies differ in phase by 90 degrees (e.g. sine and cosine of the same frequency) or by integrated complex down-converters.

This does not violate the sampling theorem since the maximum frequency, as an either positive or negative frequency, is still at most half of the sampling rate. Rather it could be said that the bandwidth of a complex sampler at sample rate R is also R, whereas

the maximum resolvable frequency f is $|f| \leq R/2$.

The shift frequency by which the radio signal is shifted down, usually does not coincide with the center frequency of the modulated signal. It would also be possible to shift the center-frequency to an intermediate frequency within, such that the complete signal bandwidth is inside the sampling range of the ADC. This would have the drawback that this frequency shift would be dependent on the bandwidth of the modulated signal, which would lead to a more complicated design, especially when considering the two-step tuning process as outlined above. Furthermore the total available bandwidth of the sampling process would be reduced.

Digital Signal Processing After reception of the radio signal by the software-defined radio, the signal may be transformed by any kind of digital signal processing method. The following summarizes some common techniques associated with digital radio signals. More in depth discussion of such methods can for example be found in [Fre02] or [Smi03].

Digital Filters One of the main applications of signal processing is digital filtering. Filters (or any linear digital signal processing system) can be described by their impulse response and are generally categorized as either filters with finite impulse response (FIR) or infinite impulse response (IIR). IIR systems may also include more complex systems such as oscillators.

There are many applications for filters including for example frequency shift-keying demodulation by matched filtering or pulse shaping to minimize inter-symbol interference.

Carrier Tracking Due to instabilities in both the transmitter and the receiver clocks the center frequency of the radio signal is usually not constant, but may be subject to wandering. To compensate for such wandering signals, the current carrier frequency can continuously be estimated and the tuning adjusted accordingly. For example a Costas Loop can be used for this application.

Modulation/Demodulation More generally, there are many different methods to modulate and subsequently demodulate an information stream onto a radio signal, that is to modify the radio signal to subsequently transfer the information stream. The smallest transmitted entity does not necessarily have to be a single bit. More complex modulation methods encode more information in a single step and thereby raise the data rate without increasing the number of modulation steps per second. The general term is "symbol" and the number of transmitted symbols per second is called the symbol-rate.

The most important parameters which can be used to encode symbols are the amplitude, the relative frequency and the phase of the carrier wave. This leads to fundamental modulations such as on-off-keying (OOK), amplitude modulation

or amplitude shift-keying (AM, ASK), frequency modulation or frequency shift-keying (FM, FSK) and phase modulation or phase shift-keying (PM, PSK). The term shift-keying implies the transmission of digital information, whereas direct modulation is rather used for analog information.

More advanced modulation methods combine multiple fundamental methods to achieve larger symbol spaces. An example for such a method is quadrature amplitude modulation (QAM), which can be interpreted as a combination of amplitude and phase modulation.

Synchronization In any asynchronously transmitted encoding scheme, the timing of the individual bits or symbols has to be recovered from the signal itself. Similar to the carrier frequency, the symbol duration may also fluctuate due to clock instabilities at either side. Thus algorithms need to be employed to recover and track the symbol duration in the received signal.

6.4 Applications

Since its introduction, many different applications for software-defined radio have emerged. The two most promising emerging paradigms are cognitive radio and intelligent antenna systems.

Cognitive radio aims to deal with the increasingly important problem of spectrum allocation. While the amount of concurrently operating wireless devices is rapidly increasing, the amount of usable electromagnetic spectrum is limited. Thus the range of available frequencies and bandwidth is becoming more scarce and also more expensive, especially in commercial applications. More bandwidth efficient modulation methods help to reduce the impact of many transmissions and help to reclaim certain frequency ranges. One example is the increasing digitalization of broadcast radio and television signals. Also more advanced spread-spectrum methods such as code-division-multiple-access (CDMA) allow the collision-free and concurrent use of the same radio band by multiple parties.

Cognitive radio presents a different approach to this problem. Since not all frequencies are in use at every place and at all times, two devices that wish to transfer data, can in advance negotiate a suitable and interference-free radio channel. Thus the participants in a radio communication adapt their behavior such that interference with other entities or environmental sources is avoided on a local level. This requires both the wide-band scanning of the spectrum to identify free segments and the ability to alter the radio devices parameters such that the free segment is used.

Intelligent antenna or smart antenna explores real-time signal processing in antenna arrays. The goal is to operate an array of antennas as a single antenna with dynamically

adaptable performance. Applications include estimating the direction of the communication partner and subsequently concentrating the antenna radiation pattern in that direction. This is also called beamforming. The effect is that especially spatial interference can be mitigated and also that the transmission itself causes less interference in other directions.

Apart from these prominent cases, there are still many different fields of application for SDR. Especially multi-purpose RF hardware greatly benefits from the inherent flexibility and reconfigurability. The SpeakEasy network (see [LU95]) was a first approach to unify many different wireless telephony protocols into a coherent system.

Mobile chipsets for embedded or mobile computers may profit from SDR approaches by including different LAN or PAN technologies into a single integrated module.

There are also multi-protocol receivers for GNSS handsets that offer seamless integration of GPS, Galileo and GLONASS systems. Finally any application where radio parameters, such as transmission power or frequency, are not known in advance can benefit from software-defined radio approaches. Thus it is possible to produce generic hardware and offer configuration flexibility to the customer.

Benefits for parasitic tracking Software defined radio also offers distinct advantages in the context of parasitic tracking scenarios.

While there are a number of dedicated wireless localization methods available, parasitic tracking focuses on the re-use of existing infrastructure with established protocols. Compared to traditional radio equipment, SDR receivers for such infrastructure make much of the signal processing internal to host computer. Thus the radio signal, as well as many intermediate steps during demodulation, are potentially available via software interfaces to the parasitic tracking framework. Non-digital components, such as those contained in the analog front-ends, are also more accessible from the software side. As the complete reception and transmission process is controlled by the host computer, any configurable analog components must expose their parameters via a suitable interface. This includes for example the tuning process or the amplifiers on the front-ends.

In summary, software-defined radio is beneficial for use in parasitic tracking scenarios, since it offers better access to relevant data and better control of the reception or transmission process.

7 IEEE 802.11 based Localization

This chapter describes the investigation of IEEE 802.11 Wi-Fi technology in relation to parasitic tracking and software defined radio.

7.1 Basic scenario

There are already technologies available which could potentially be of great interest for use in parasitic tracking contexts and ubiquitous augmented reality scenarios. One of these prime candidates for parasitic tracking is Wi-Fi wireless networking technology. Besides being interesting in its own right as a parasitic tracking method, it also serves as an example for the discussion on software defined radio and EM propagation presented in the previous chapter.

There are two main reasons for Wi-Fi to be interesting in parasitic tracking contexts. First Wi-Fi currently represents one of the technologies which already approach true ubiquity, because the increase of networked personal and mobile computing devices also required a similar increase of suitable data communication facilities. In consequence Wi-Fi is amongst the primary methods for network and internet access. This is especially true in home or local environments due to the usually reduced costs associated with Wi-Fi traffic, especially if compared to data traffic over cellular networks. There are many different local providers for Wi-Fi access, ranging from hotels offering free wireless internet access to their customers and cellular network operators offering Wi-Fi hotspots to amateur wireless metropolitan area network initiatives.Thus the density of Wi-Fi infrastructure approaches the definition of being ubiquitously available. While the development of higher network density in metropolitan areas naturally happens more quickly, the same progress can be observed in rural areas. This development can furthermore be documented by the various "War-Driving" efforts conducted since the beginning of Wi-Fi installations and culminating in the currently ongoing systematic mapping of the complete environment by operators such as Google Inc.

Note though that this concerns the primary existence of observable wireless networks while no assumptions on the ability of any user to use any network is made. It may very well be that a user can see many different wireless networks, but cannot access data through any of them. Thus while Wi-Fi networks may be ubiquitous, data access via Wi-Fi may not.

The second reason for the interest in Wi-Fi technology in a parasitic tracking context which is also partly responsible for this rise in ubiquity, is the ease of operation of Wi-Fi

equipment. It is important to realize that this ease of operation applies to both the user equipment and the infrastructure equipment.

This is a significant difference to other kinds of data communication infrastructure such as cellular networks. While there operation of such user equipment also is easily possible, infrastructure equipment can only be operated by the network provider. Wi-Fi installations, on the other hand, can be operated by any private or commercial entity. Thus Wi-Fi is an example of an uncontrolled and potentially very volatile infrastructure which is maintained by many third parties.

Importantly, Wi-Fi is based on radio frequency emissions and thus the software defined radio technologies previously discussed can be applied here.

Scenario With this background, the basic scenario of parasitic Wi-Fi tracking can be stated.

The fundamental assumption is that Wi-Fi access points are fixed arbitrarily in the environment and their existence is visible to the user. The function of these access points is that of beacons periodically signalizing their existence and the properties of the wireless network. It is possible that this happens entirely without any actual data communication except for Wi-Fi protocol maintenance messages and thus without an actual access to the wireless network. The Wi-Fi beacons can be seen as anchors in the environment and the tracking task is to determine the user's location based on these observations.

7.2 Related Work

The area of Wi-Fi-based localization has, as previously mentioned, already seen considerable interest, in part due to the ubiquitous availability of this kind of infrastructure. As a consequence, a number of different localization applications and frameworks already exist.

One of the earliest approaches is the RADAR system([BP00]). It mainly uses position estimation on the basis of reference maps. These maps for signal strength and signal-to-noise ratio for each Wi-Fi beacon were recorded as empirical measurements over a large area. This system also evaluated the impact of different user orientations and it compared the fingerprint based localization to a range based localization on the basis of a radio propagation model.

Another Wi-Fi localization framework is presented in [Zho06]. The WITS system also uses reference signal strength maps either with nearest neighbor selection or Bayesian probabilistic localization. The position estimate is further smoothed using history.

Similarly [EM06] used empirical reference maps with Bayesian probabilistic localization in a $40\,\mathrm{m} \times 40\,\mathrm{m}$ office building. The measurements are filtered using a particle filter,

which also combines the Wi-Fi localization data with an inertial orientation sensor in a "mutually correcting architecture".

A system for tracking of pedestrians using Wi-Fi is presented in [WLS+07]. The location of the pedestrian is estimated using reference signal strength maps with k-nearest neighbor localization. The data is further combined with accelerometer data using a particle filter.

In [SHR+08] the position of the user is determined using a modified weighted centroid localization scheme. The weight of each visible Wi-Fi beacon is determined by its signal strength as observed by the user, where stronger beacons are given more weight. The scheme furthermore uses dynamic weighting by adjusting the weighting function for each beacon individually.

The localization method described in [KPV10] uses a non-parametric information filter to simultaneously predict the motion model of the user as well as incorporate RSSI measurements, based on a reference signal strength map.

To reduce the number of required anchors with known locations, [LHSC05] performs self-mapping of a Wi-Fi environment starting from a set of "seeds" with known locations. The procedure determines ranges to the various beacons in the environment from the signal strength and constructs an edge-weighted graph with weights set to the estimated distances. The locations of the nodes are determined by constructing a graph embedding on the plane which minimizes contradictions.

A common characteristic of all systems so far is that in each case the actual Wi-Fi reception process is treated as a black box. Only signal strength or signal to noise ratio measurements are received from the Wi-Fi driver, whereas further reception parameters are not available. Yet more precise Wi-Fi signal measurements are possible. For example [FMZ+10] describes methods using a software defined radio to measure the channel-impulse-response of the propagation channel of any off-the-shelf Wi-Fi transmitter. While no range information or localization was computed in this instance, such information is potentially also of great use for improved propagation models and ranging methods.

The possibilities of attacks on such Wi-Fi based localization schemes are illustrated by [TRPČ09], which stresses the need for diversifying the localization methods.

7.3 IEEE 802.11 basics

The currently predominating standard for wireless network communications is the suite of IEEE 802.11 [IEE07] standards. These actually define a range of different wireless networking technologies using various frequency ranges and communication speeds. In the following the discussion is mostly concerned with the basic IEEE 802.11 standard without extensions.

Radio transmissions for the Wi-Fi network are modulated as binary phase shift keying

(BPSK) signals. This means that a binary symbol is encoded in the phase of the carrier wave. To account for phase ambiguities in the transmission process (rotation of the phase constellation by at least 90 degrees) the modulation furthermore uses differentially encoded BPSK (DBPSK) where a binary digit is encoded by the existence or non-existence of a phase shift.

For signal shaping and to increase the differentiability of the individual transmitted symbols the IEEE 802.11 standard also includes a root-raised-cosine filter at both the transmitter and the receiver side. This effectively distributes a raised-cosine filter across the two sides which has the property to eliminate or at least reduce (if the transmission channel introduced non-linear distortion of the signal) inter-symbol interference (ISI) of the transmitted signal.

To improve the robustness of the modulated signal, a technique known as *direct sequencing spread spectrum* (DSSS) is used. By multiplying the modulated carrier wave by a very fast pseudo-random sequence (chirps) with chirp-rate much higher than the frequency of the signal, the engery of the signal is distributed across a larger bandwidth. At the receiver the spectrum-distributed signal is multiplied by an inverse sequence resulting in the original signal. Since any narrow-band interference in the spread signal is also multiplied by the inverted chirp sequence at the receiver, its impact on the recovered signal is greatly diminished. These procedures are called *spreading* respectively *despreading* of the signal and IEEE 802.11 employes a sequence of the well-known class of "Barker codes" as chirp sequence, which feature minimal autocorrelation.

On the media access control layer (MAC-layer) the IEEE 802.11 network implements a carrier sense multiple access network (CSMA), where each transmitting entity is required to sense the medium prior to starting its own transmission. While this is similar to Ethernet for example, a wireless node cannot both transmit and listen simultaneously. Thus collision detection as in the Ethernet example, cannot be implemented and Wi-Fi rather employs optional collision avoidance (CA) by using a request-to-send (RTS) / clear-to-send (CTS) handshake between sender and receiver. These handshake packets are transmitted as special short bursts to reduce the probability of themselves causing a collision. In case of an occupied medium or a failed CA handshake a randomized truncated binary exponential backoff algorithm is used. On this layer also individual Wi-Fi devices can be distinguished by their MAC-address.

To quantify the quality of a wireless network link the received signal strength indicator (RSSI) is used. While this name suggests to be indicative of the energy of the received signal, no such assumption can be made in the case of IEEE 802.11. The standard only specifies that the RSSI should be a "relative value" within the context of a vendor specific implementation. There are no requirements for this value to be interpretable in any physical sense. Note that this situation is different for example for the WiMAX wireless networking standard IEEE 802.16, where RSSI is defined quite rigorously and has definite ties to physical quantities.

All these techniques and methods need to be implemented in software for a SDR based

Wi-Fi receiver. The IEEE 802.11 standard operates in the 2.4GHz range of the radio spectrum, which is reserved for industrial, medical and scientific use (ISM-band) and thus requires no licence to operate transmitting devices. This is one of the reasons for the easy availability of Wi-Fi infrastructure equipment, as mentioned above.

7.4 Improving model based ranging

The basic premise of the following inquiry is that by replacing commercially provided, closed Wi-Fi user equipment by SDR based receivers, the physical significance of the received signal strength indicator can be improved. This, in turn, can be used to improve localization performance.

As mentioned previously, one of the two usual strategies to exploit the reception performance of Wi-Fi signals for localization is fingerprinting, i.e. to record maps of the environment and later approximate the user's position by matching the current sensing situation to the pre-recorded values.

As already discussed, there are disadvantages associated with this approach. First, the preparatory step of recording the environment is extremely time consuming and usually has to be redone even when only a single Wi-Fi beacon is modified since the reception of the individual beacons may be correlated.

A different approach circumventing these disadvantages is to estimate the ranges between the user and the beacons based on the signal information. While in this case the setup time is very short and each beacon can be handled completely independently, the crucial requirement of this method is the ability to interpret signal reception strength values in a physically relevant way. This is usually not possible with commonly available hardware, as described above.

Black box receiver The main problem preventing physical interpretation is that usually Wi-Fi reception hardware is viewed as a black-box. No additional data or parameters on the current reception process are available nor is the reception process controllable beyond basic activation or deactivation. Nevertheless these internal parameters can severely influence the end product of the transmission chain and thus influence the estimated RSSI value.

The most important parameter influencing the reception of a Wi-Fi signal is the receiver gain. As explained in section 6.2.2, the energy of a wireless signal is influenced by many factors during the complete transmission chain between the transmitting and the receiving equipment. This includes final amplification of the radio signal that is picked up by the receiving antenna prior to demodulation and extraction of the payload information.

For example, an additional amplification in the receiver by 3dB (doubling the energy) results in a signal where the voltage across the fixed impedance is $\sqrt{2}$ larger. Thus

the amplitude of the signal, as seen for the RSSI estimation, will similarly be $\sqrt{2}$ times larger. Consequently the local receiver gain can obviously dominate or at least drown any distance dependant channel attenuation of a wireless signal.

At the same time the main goal of consumer Wi-Fi access equipment is to provide best possible communication quality to any visible Wi-Fi access point. This means that usually the receiver gain of a Wi-Fi access equipment will not be kept constant but is rapidly adjusted individually for each Wi-Fi beacon. This leads to the fact that RSSI values still indicate the quality of the communications link between Wi-Fi beacon and Wi-Fi receiver, but can no longer be interpreted in any consistent physical model.

Software Defined Radio The proposed way to overcome this limitation is to replace the usual consumer grade Wi-Fi receiver with a software defined radio solution. This reduces the number of unaccounted influences during the reception process and makes all internal processes and parameters visible to the controlling software. Thus, it provides an interface for direct access and direct control of both reception parameters as well as the received signal in various states.

In such a system there are still unknown or at least imprecisely known influences (for example thermal effects at the receiver, antenna impedance matching or antenna impedance variations over the frequency range) which make an exact calculation of the received signal energy difficult. Nevertheless a controlled SDR based Wi-Fi receiver can be used to at least establish a physically based interpretation of the received signal by incorporating the known radio reception parameters.

Note that such a system to measure absolute signal energy would require an even more tightly controlled setup. These are usually called test receivers and are used for example for quality assurance, EMC testing or even for radio astronomy.

As SDR based radio hardware in consumer equipment is becoming more and more common, this step towards SDR based receivers can be seen as the early adoption of a future trend. The assumption is thus that by moving from specialized radio hardware to more generic SDR controlled receivers the reception process on the software side will become more transparent and controllable and thus more exploitable. On the other hand, this approach would also be possible with common non-SDR radio hardware, if the relevant interfaces were available. But since in the standard use case these interfaces do not contribute functionality but rather only add costs to design and implementation, such accessibility can only rarely be found.

7.5 Experimental Setup

To study the impact of using SDR based receiver technology, different experiments were conducted. This section describes the common components of the setup of these experiments.

SDR Wi-Fi receiver As SDR hardware platform the "Universal Software Radio Peripheral (USRP)" version 1 hardware from "Ettus Research" was used. The USRP hardware is an open design and features all necessary components required for a wide range of SDR applications. For a more detailed description of this device and its applications see [Ham08] or [Val08].

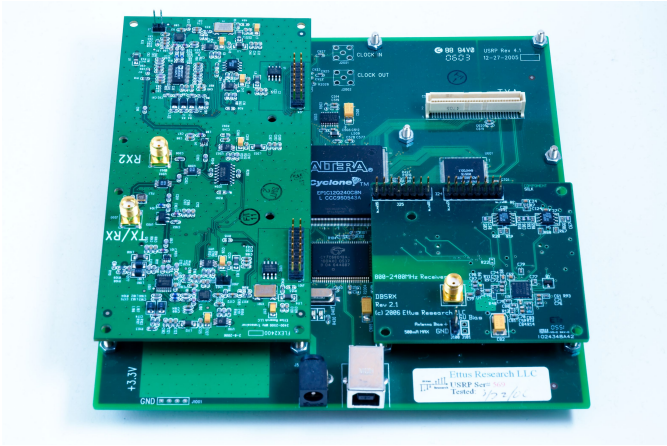


Figure 7.1: USRP mainbaord

The main conversion between analog RF signals and digital information is performed by 4 of high-speed A/D (64 MS/s) and 4 high-speed D/A (128 MS/s) converters. Note that for each reception or transmission channel two independent A/D respectively D/A converters are needed as both in-phase and quadrature (I/Q) signals are required to represent any signal, resulting in two parallel reception and two parallel transmission channels. A further crucial component is the high-performance FPGA which can be reconfigured at power-up and which performs various time-critical tasks such as digital up- or down-conversion, time-stamping, data decimation or other digital filtering. The communication between USRP and host computer is performed by a High-Speed USB 2.0 controller which achieves data rates up to 8 MS/s using complex samples of 16 bit length.

To use the USRP at a certain radio-frequency range, a suitable RF frontend is required which handles tasks such as impedance matching, amplification, tuning and down-conversion for reception. These frontends are attached to the base USRP board as daughterboards. Similar to the number of available A/D and D/A channels, there are two receiver and two transmitter slots available. For these experiments the "DBSRX" receiver daughterboard was used, which operates in the frequency range from 880MHz to 2500MHz, therefore sufficiently covering the ISM band used for Wi-Fi ranging from 2.4GHz to 2.5GHz.

On the software side the associated RF signal processing framework "gnuradio" was

used. Gnuradio features many ready digital communications algorithms and procedures as building blocks. Among the available blocks are digital filters (FIR or IIR), various modulation and demodulation methods (e.g. MSK, QAM) as well as more involved algorithms such as bit-time-recovery or carrier-tracking.

These blocks can be connected in a directed data flow graph, similar to the previously described Ubitrack dataflow framework. Time critical data processing blocks are written in C++, while the connection and reconfiguration of the flow graph is done in Python. The framework can furthermore be easily extended by new algorithms.

For the experiments the software package from SPAN Lab at Utah University was used which implements a complete IEEE 802.11 physical layer using the gnuradio framework. An additional feature of this implementation is that the Barker code despreading of the spread spectrum signal is already performed in the FPGA on the USRP board, which greatly reduces the required communications bandwidth between USRP hardware and host computer. This package was further modified to include more convenient RSSI logging.

The SPAN Lab framework computes the RSSI value by computing the mean of the squared magnitudes of the complex samples representing the first 16 bits of the IEEE 802.11 packet header. This average magnitude is represented in a logarithmic scale as referenced to the maximum possible amplitude as the RSSI of the packet. Thus the RSSI value in this implementation is directly related to the energy of the wireless signal as seen by the A/D converter. Note that despreading the signal again concentrates the signal energy in more narrow bandwidth, but keeps the overall signal engery constant.

The final component of the Wi-Fi SDR reception setup was a omnidirectional Wi-Fi antenna which was mounted to the antenna connector of the DBSRX daughterboard. The directional gain of the antenna is about 8dB as compared to an isotropic antenna, mainly by compressing the antenna field vertically while still maintaining a radial reception pattern in the horizontal plane.

Further components To compare the effectiveness of the SDR approach two different consumer Wi-Fi access devices were evaluated. The first device was a PCMCIA "Belkin F5D7010" network card, whereas the second card was an "Atheros AR5212 802.11abg" internal network card.

These were used as standard Wi-Fi interfaces in a computer system. For each individual device the list of currently visible Wi-Fi beacons could be requested including annotations about the associated RSSI for each access point.

To provide and simulate the necessary Wi-Fi infrastructure in the following experiments, six identical Wi-Fi routers "USRobotics MAXg 5461a" were used as Wi-Fi beacons at fixed positions.

The configuration of the devices was exactly identical with the obvious exception of the MAC address in order for them to be individually distinguishable.

Figure 7.2: Wi-Fi Beacons

7.6 Range estimation

In the first experiment the impact of using SDR based reception hardware was evaluated as compared to standard consumer Wi-Fi reception hardware. The focus of this experiment was a straightforward range estimation task. The performance in this range estimation forms the basis for further range based localization methods using fixed anchors.

More precisely the subject of this experiment is to evaluate whether consistent relationships between a combination of RSSI values and receiver gain and physical distance between sender and receiver can be established. The test environment for this experiment uses short distances (in the context of Wi-Fi localization) between anchors and furthermore uses the same ESSID on all anchors. Thus semantic localization (based solely on the visibility of a combination of networks) is not possible in this scenario.

Setup For this evaluation a physical test environment was created. First a room of suitable size was emptied in order to minimize interference from objects in the vicinity of the radio equipment. Along the longer axis in the middle of the room the individual Wi-Fi beacons were placed as anchors at regular intervals of 1m. The individual positions were determined by measuring tape which was fixed to the floor. Note that the positioning error due to the measuring tape is expected to be very small compared to the overall resolution of the localization methods and can therefore be ignored.

In a similar fashion the above described collection of Wi-Fi receiver systems was placed at a similar interval of 1m from the first anchor. As described, this collection consists most importantly of the USRP SDR receiver hardware and of the two consumer Wi-Fi access devices.

Execution With this setup, the experiment was conducted in the following manner: After powering up the Wi-Fi routers they periodically sent out beacon frames. These beacon frames were received by the different Wi-Fi access devices, which calculate the link quality for each access point. During the run of each repetition the measurements from

Figure 7.3: Hardware setup

all devices were logged periodically. As described above, the SDR receiver calculates a new RSSI value for each received packet independently and thus for each received packet a measurement value was logged. The other Wi-Fi receivers did not actively send out the RSSI values but had to be polled periodically. For these devices the polls were performed and logged as fast as allowed by the device and the OS driver.

The receiver gain of the SDR receiver was held constant during each test run and was systematically altered between different repetitions. That way both the influence of the receiver gain on the measured values could be explored and suitable ranges for further localization methods could be identified. The consumer Wi-Fi access devices didn't have tunable parameters and so no changes were made between different repetitions.

Each repetition was run for about 180 seconds; during this time the complete setup was kept static and no persons were present in the room.

Finally between different repetitions subsets of Wi-Fi routers were selectively turned off or on thus different scenarios could be evaluated and the independence of individual measurements could be verified. The different activation patterns can be seen in figure 7.4.

Analysis and Results For analysis and interpretation of the measured values, a normalized histogram plotting the RSSI vs the number of packets was generated for each of the test runs and each of the input devices. From these histograms a number of properties are visible.

First the actual positions of the different Wi-Fi anchors on the RSSI scale can be located. Comparing the positions of different Wi-Fi anchors both the spread of the measured values as well as the relative positions of the anchors amongst each other were determined. A greater spread denotes a more expressive situation in which different ranges produce greater variations in the measured values. A consistent ordering between the stations is indicative that distance and received signal strength are indeed correlated.

Moreover, comparing the positions of the anchors across different repetitions shows

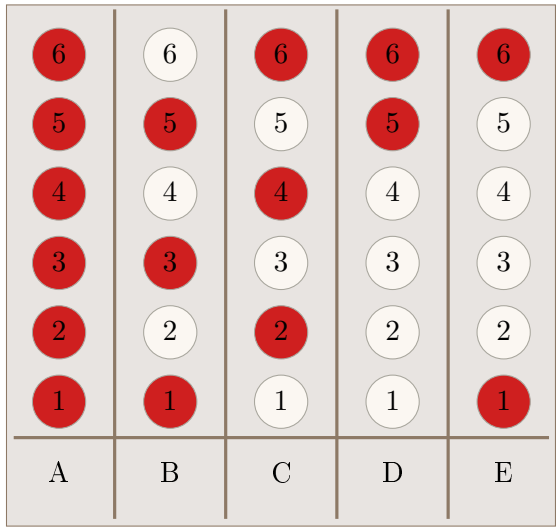


Figure 7.4: Beacon configurations

the stability of the setup and also indicates the dependability of the association between measurement and estimated range. Finally by comparing the abscissae of Wi-Fi anchors between SDR and reference systems the difference in performance can be deduced.

Furthermore for each repetition the mean and standard deviation of the reception parameter of each anchor was calculated.

The first result of this experiment is that the SDR based receiver indeed shows a significant correlation between distance and RSSI. This is also true for any gain setting. As expected the mean values of the Wi-Fi beacons cannot be directly compared between different gain settings, since using a different gain setting each beacon can be shifted across the entire RSSI range. Thus while a correlation between RSSI and distance is possible, the receiver gain value needs to be constant or at least known. The two parts of figure 7.5 show both the linear ordering of the Wi-Fi beacons in accordance to their respective distance and the possible shift using different gain values.

At extreme low or high gain setting not all beacons were received either due to being too weak for correct reception or due to drowning the receiver with too high signal levels. Also note that at these extreme gain levels while the overall ordering is still visible, the spread between receivable beacons is reduced. The beacons are compressed against the limits of the RSSI range. This is visible in figure 7.6.

On the other hand, the data obtained from the two reference consumer devices exhibited very different behavior, as can be seen in figure 7.7. The histograms for these devices show that the mean abscissae of the various beacons received show no direct relationship to the individual distances of the beacons. Furthermore the values of all the Wi-Fi beacons were clustered in the medium range of the scale. This was expected

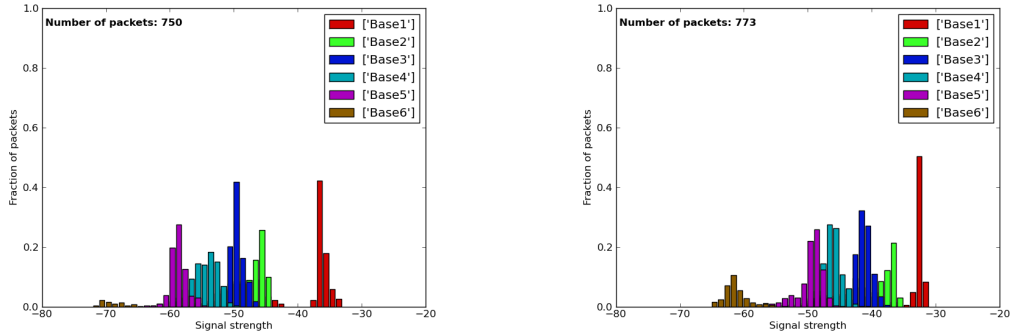


Figure 7.5: SDR based RSSI measurements

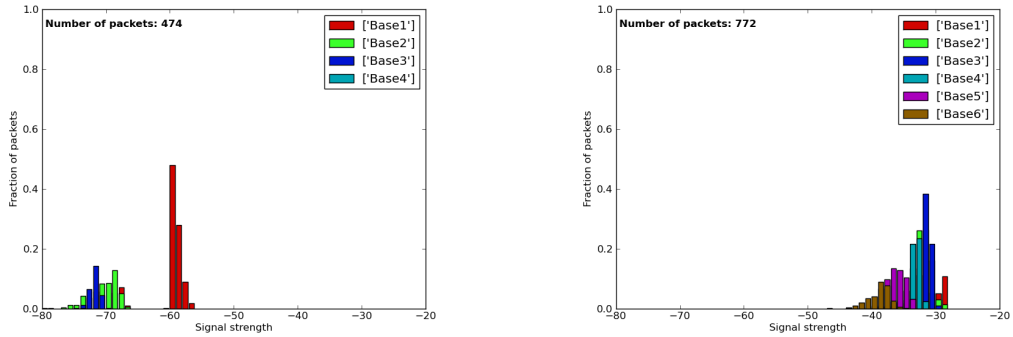


Figure 7.6: RSSI at extreme low or high receiver gain

and indicates the action of internal control mechanisms which try to keep all received packets within "good" link quality. Thus the spread was much lower and no distance related ordering of the RSSI values was discernible.

Furthermore comparing results from different repetitions taken at different times it can be seen that for the SDR based receiver not only the relative ordering of the visible anchors is stable, but also the range of the absolute values is similar. This is visualized in figure 7.8.

A final observation involves the RSSI update rates of the different devices. As already mentioned, the SDR based device produces a separate RSSI estimation for each individual packet received. This resolution could even be increased by providing multiple estimates from different portions of a Wi-Fi packet. This could possibly be used to monitor channel fluctuations and to devise adaption strategies.

The "F5D7010" receiver on the other hand produced estimates at a much slower rate.

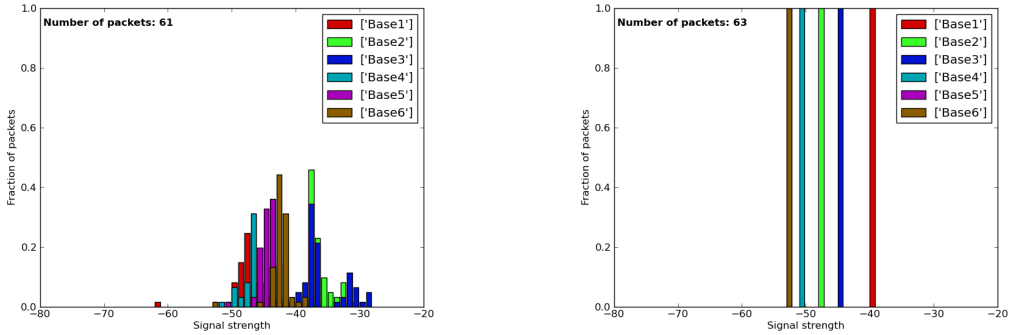
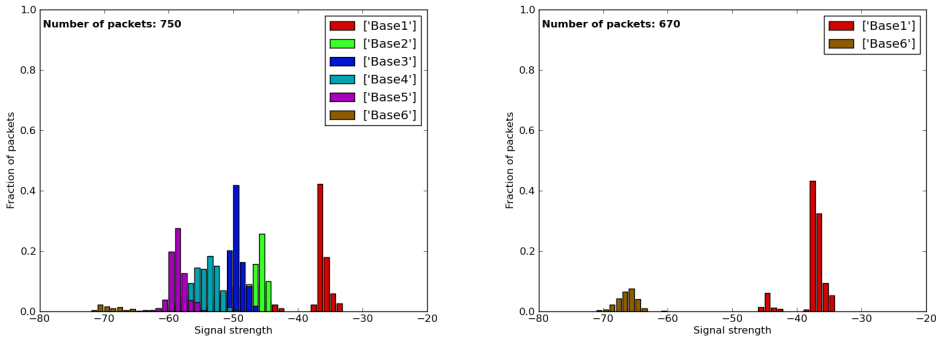


Figure 7.7: RSSI as received by reference devices



(a) Configuration A

(b) Configuration E

Figure 7.8: Different beacon configurations (SDR receiver)

During a fixed time interval, this receiver produced about 60 measurements per base station, whereas the SDR receiver produced about 750. This is mostly due to the delay of the involved polling operation, which took several seconds. As can be seen in the corresponding histograms the measurements nevertheless show a certain distribution around its mean. This indicates that the measurements are still produced in real time and are subject to fluctuations and noise.

The "Atheros" receiver, however reported the exact same RSSI value for any query during a repetition. Even after switching off several Wi-Fi beacons they were still reported for several minutes. Figure 7.9 shows the RSSI values of this receiver after beacons 1,3 and 5 had been turned off for at least eight minutes. As can be seen, the cache is updated during the test run and thus the peak of these stations is smaller. From this it can be concluded that the RSSI value of each beacon is only very seldomly updated and cached

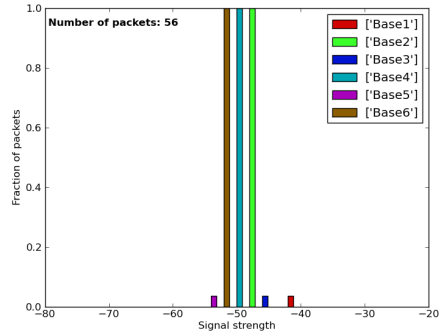


Figure 7.9: RSSI caching artefact as received by reference device

for long durations.

This makes the last receiver especially unsuitable for even moderately moving users, while the SDR receiver potentially could also be used for dynamic tracking.

Conclusion In conclusion of the ranging experiment it can be summarized that SDR receivers greatly improve the ability to link RSSI values to actual physical properties such as distance, while this is much more difficult for standard consumer grade Wi-Fi equipment. Both the expressiveness and the robustness of the SDR approach can be seen as superior compared to other devices. Furthermore the update rate of the SDR receiver is much higher.

A range of receiver gain values for the USRP hardware was identified that should be suitable for range estimations of the order occurring in the experiment.

7.7 Localization based on range model

The next experiment tried to apply this SDR based ranging method to an actual localization scenario. By placing the Wi-Fi beacons at known positions and measuring the RSSI with known receiver gain a range estimate should be possible. By using a sufficient number of ranges to different anchors, the position of the user should be determinable.

Setup The setup of the experiment consisted of the above described Wi-Fi beacons arranged in a 3x2 grid in an outdoor area. The spacing of the grid was equally 5m in each direction, resulting in total area of 15m x 10m. In order to guarantee a sufficiently accurate positioning of the beacons in the grid, the positions were verified with a total station.

The SDR hardware setup as described above was again used as the Wi-Fi receiving equipment and was placed at various unknown locations which had to be inferred from

Figure 7.10: Outdoor setup

the Wi-Fi measurements. The comparison against the two consumer Wi-Fi receivers was not performed due to their very poor performance during the previous experiment. For reference, these locations were also recorded using the total station. To enable untethered operation, both the beacons and the SDR receiver were modified to run on battery power.

Execution After powering up the Wi-Fi beacons the SDR receiver was placed at an arbitrary point inside or at one of the edges of the grid. The RSSI measurements of the different beacons were recorded together with the current selected receiver gain. During each test run, the gain of the receiver was periodically changed every 5 seconds. The range of the gain parameter was determined from the results of the previous experiment. Each test run took 300 seconds, after which the SDR receiver was placed at a new location. Overall 5 different locations were evaluated.

Analysis and Results To analyze the obtained measurements and perform the actual localization a range between the receiver and each of the Wi-Fi beacons was estimated based on the recorded RSSI and the associated gain. In case a beacon was not received at all, it can still be considered negative evidence of the proximity to this beacon. The minimal receiver gain needed for a beacon to be visible can be used to further constrain the estimated range.

These ranges can further be used to estimate the location of the receiver. Areas of probable location were constructed by intersecting positive and negative circles around the anchor points. The radius of the circles correspond to the estimated range between receiver and the individual anchor.

For evidence of presence of the receiver at a certain distance to a anchor point, a positive circle is applied. This indicates that the location of the receiver needs to be

Figure 7.11: SDR receiver

within the estimated range. For evidence that the presence of the receiver cannot be within a certain range of an anchor point, a negative circle is applied. For example this is the case if a beacon cannot be received with a certain receiver gain. The collection of positive and negative circles are used to estimate the position of the receiver.

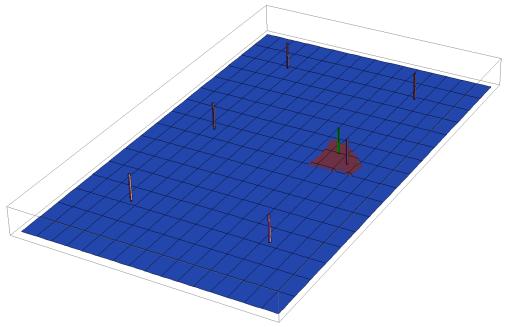


Figure 7.12: Localization analysis

As a result of this experiment, although a localization was obtained in each case, not every result was correct. While some test runs produced encouraging results, similar bad cases were also encountered. The average RMS error between all five test runs was about 1.2m.

A more advanced method to combine presence evidence from different anchors is needed and especially the handling of contradicting evidence needs to be addressed in the future.

8 Long-range RFID based localization

The work described in the first part of this chapter was partially funded by the Trackframe project and was conducted in cooperation with Benjamin Becker and additional support from Marcus Fey. Furthermore parts of the description of the first part of this chapter are based on the papers [BHK08] and [HBK10].

8.1 Basic scenarios

Continuing in the spirit of the previous chapters, the usefulness of other kinds of radio-based infrastructure can be taken into consideration. An interesting example of the possible use as a parasitic tracking infrastructure is radio-frequency identification technology (RFID) and RFID tags.

In general, RFID infrastructure consists of a large number of low-cost RFID tags, which are applied to various goods or items. In this sense RFID tags operate similarly to barcodes, as they allow the easy and computer-aided identification of these items.

The advantage of RFID tags over classical identification features is that these tags can be scanned contact-free and most types of tags support to be read at a distance. Thus precise placement of the item in relation to the reader is no longer necessary, and, depending on the technology, even multiple items can be detected simultaneously. These kinds of tags seem most suitable to the parasitic tracking approach.

The assumption that RFID tags form a kind of exploitable infrastructure is derived from the increasing density of tags installed for various applications. RFID tags see increasing attention in areas such as logistics, supply chain management or quality assurance. Thus the collective cloud of installed tags can be regarded as a set of fixed anchors for localization.

RFID technology has already seen some attention in ubiquitous computing, although mostly short-range technology is used. In this chapter the applicability of parasitic tracking to long-range passive RFID infrastructure is investigated. The setup and evaluation are discussed on the basis of two different scenarios.

Airplane localization scenario The first RFID tracking scenario is derived from an application which was investigated by EADS Innovation-Works in the course of the Trackframe project. It is directly oriented towards the needs of an industrial maintenance worker.

Figure 8.1: Computer Assisted Maintenance Worker

Industrial scenarios are especially interesting as they offer the possibility to introduce new technology in a controlled manner and therefore more diverse technologies are likely to be present. At the moment, RFID technology is predominately used by supply-chain and life-cycle management applications.

The subject, by which this scenario is motivated, is the setup of aircraft cabin maintenance (Figure 8.1). The maintenance worker is instructed to inspect and eventually service various parts of a completed aircraft cabin. This process is supported by giving the worker navigation guidance to the affected sections and by providing context related information, such as assembly or disassembly instructions for the device at hand.

It can be assumed that RFID tags are already distributed in the cabin and the aircraft fuselage for the purpose of quality assurance. It is possible to use these tags for inside-out tracking and pose estimation at a reasonable quality with very little setup time. To improve the usability of the application, the localization was supported by an inertial measurement unit mounted to the interaction device.

This is demonstrated on a prototype of an automated maintenance support system. Location awareness herein reduces the need for direct user input since the system is able to detect the current state of the worker.

Showroom scenario Opposed to the airplane cabin scenario, which was directly motivated by an application, this scenario intends to examine the possibility to locate the user solely based on RFID measurements.

In this scenario a user is trying to localize himself in the presence of RFID tags. As in the previous setup, the tags are considered to be the infrastructure distributed

throughout the environment and their locations are assumed to be known. Thus the tags can be seen as anchors in the environment.

Similar to the previous scenario, the tags are passive, long-range RFID tags without any special functionality besides being uniquely distinguishable. The tags themselves do not possess any capability of active radio communication.

To detect and communicate with these tags, the user is equipped with a suitable RFID reader-device. The localization is then performed by exploiting changes in the signal power required to communicate with the RFID tags, which is dependant on the distance between transmitter and receiver.

Note that passive long-range RFID technology is based on modulated backscatter, the energy received by the RFID reader is part of the energy initially transmitted by the same reader. The electromagnetic wave travels the distance between reader and tag, is reflected and travels the same distance back. Thus the power received by the RFID reader depends even more strongly on the distance than in a scenario with only a transmitter-receiver path.

The analysis and the localization itself are similarly constructed as the Wi-Fi localization experiments, described in the previous chapter. Furthermore, opposed to the airplane localization scenario, no extra sensors such as a inertial sensors are used. While this in itself is neither more nor less in agreement with the parasitic tracking approach, it aims to explore the performance of pure RFID-localization.

Furthermore this approach again shows the advantages of software defined radio for parasitic tracking approaches. Similar to the better controlled reception process as described for Wi-Fi localization, the RFID based localization benefits from more control over the transmission process. Contrary to a normal RFID reader, as was used in the first RFID scenario, a software defined radio RFID reader allows quick reconfiguration of the communication parameters. This includes the transmission power of the initial electromagnetic wave sent by the RFID reader.

8.2 Related Work

Computer aided maintenance and automated worker support for maintenance tasks in general have already been studied in the past (see for example [ON04] or [SL06]) and their benefit has been established.

Tracking methodologies are generally classified into outside-in and inside-out tracking. For outside-in tracking a global set of observers determines the positions of all participating entities, whereas in inside-out tracking each participant determines his pose independently using landmarks in the environment.

Similarly existing approaches for RFID based tracking can be classified into these categories. The inside-out RFID tracking uses distributed tags and a single mobile reader for the position estimation of the reader whereas outside-in uses a certain number of fixed

readers to track a set of mobile transponders. Outside-in based systems generally suffer from the need to distribute numerous readers in the tracking area, impeding the possible scalability to large areas, due to complex communication needs and high investment costs.

Although outside-in tracking does not match the general constraints of the aircraft use cases, it does use the same technological basis and therefore serves to benchmark the system presented here. Another system [NLLP04] demonstrates the possibility to enable tracking by distance grouping of transponders by varying the antenna's power, albeit using active RFID tags.

Inside-out based tracking was already used to update a robot's position estimation [HBF+04] by reading tags with known location, or more recently by [SVZ07] and [VSYZ08] who use reference based fingerprint localization. Another promising approach [BM04] is introducing the *Super-Distributed RFID Tag Infrastructure.* This scenario uses a small robot vehicle equipped with a very short-range RFID reader pointing at tags distributed in the floor. The tags are distributed on the floor using their ID to reference a specific position. In contrast to the approach presented here, their setup benefits from the short reading range increasing the quality of the position estimate but dramatically reducing the event of detecting a tag and therefore the estimation rate. When integrating and extrapolating via the inertial and movement sensors of a robot this is a feasible approach. This is also similar to the work presented in [KMA+08]. A different approach for robot-navigation is described in [LSP+10], which also focuses on the use of long-range RFID tags.

As discussed earlier, a fingerprinting approach is unfavorable for the aircraft use-case, due to its required long per-case calibration phase. The emphasis is on being able to navigate in an unknown environment using only information describing the general structure of the scene, such as CAD data or similar.

There are also further approaches to accurately model the readability parameters of long-range RFID tags, for example [HTMF07]. Another interesting approach to determine the distance between reader and tag is presented by [HWGL+10], who measures phase differences using a software-defined radio and active tags.

8.3 Technical Background

As already stated, logistics, quality assurance and life-cycle management are major domains of RFID technology. To match these requirements, tags vary in levels of complexity from simple ID transmitters to smart cards. For a survey of different RFID technologies and applications, see for example [Fin02].

Passively powered tags use the energy transmitted via the RF field for the communication response whereas active tags depend on an additional power source. Currently there are two common physical communication layers [Fin02]: near field (LF at 134.2 kHz, HF

at 13.56 MHz and 27.125 MHz) and far field (UHF at 433 MHz or 868 MHz (EU) and 915 MHz (US), SHF at 2.54 GHz or 5.8 GHz). These two technologies especially differ in their respective operating distance, which ranges from less than few centimeters in the first case to typically up to 6 m in the second case. Active powered tags achieve even greater ranges and often implement additional features like complex encryption. On the other hand, active tags suffer from a limited lifespan and additional maintenance issues due to their battery life.

In summary, active tags allow more complex applications at a higher investment per tag. Usually they are used to identify only few but special objects for a short period of time, whereas passively powered tags are often used as a long-term identification of (infra-)structural parts.

Motivated by the requirements of the aircraft cabin application, both scenarios focus on passively powered, long-range RFID tags. The underlying assumption is that these tags are being used for long-term quality assurance of structural parts and cabin components, which leads to a massively-distributed RFID infrastructure within the application area. Recent use cases in quality assurance stress the need for long-range readability beyond 1 m, from which parasitic tracking methods profit much more than they would from short range RFIDs. In the latter case the operating volume as well as the distance and readability classification via the power modulation process would be severely limited.

8.4 Aircraft scenario

This section describes the aircraft scenario, the prototype's overall hardware setup as well as the position estimation algorithm used. This is followed by an analysis of the obtained evaluation results.

The prototype demonstrates the possibility to achieve 6DoF localization and tracking of a mobile device in an aircraft cabin using a long-range reader and a cloud of distributed passive tags, taking a gyroscope's current orientation estimate into account. The goal of this application is to allow automated maintenance support based on location awareness. Maintenance of aircrafts is time critical as every minute on ground means lost revenue to the airline.

Clearly, the more promising approach for this case is an inside-out based tracking system, having no or little impact on the environment by bundling the required hard- and software on a single mobile device. The system's design for a single reader and inexpensive passive RFID tags allows the coverage of a large area with little investment. Besides, often the required infrastructure is already present from other RFID applications as listed in the prior paragraph. Also a mobile computing device can be easily extended with additional sensors like a gyroscope or Wi-Fi tracking, increasing range, robustness and precision of the localization algorithm.

8.4.1 Hardware Setup

Since RFIDs are only just being introduced in the aircraft production and maintenance process the tags in the prototype's cabin mock-up were added supplementarily. This section describes how the tracking system was embedded into the aircraft cabin and gives additional detail about the hardware used.

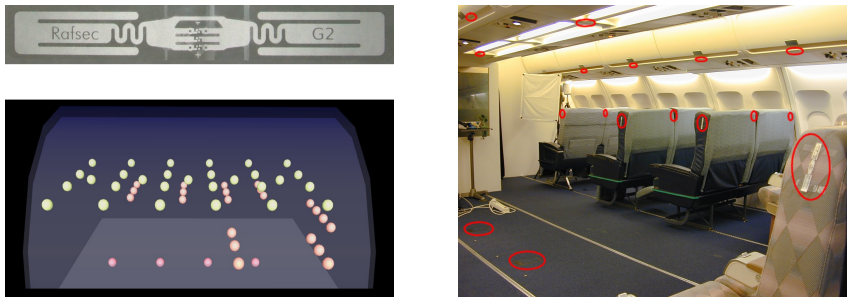


Figure 8.2: RFID Tags Distributed in Aircraft Cabin

To increase robustness as well as the position estimation quality, a large and reliable reading range is beneficial. The RFID tags which had already been deployed in the aircraft cabin testbed are passive ultra-high-frequency transponders, which match these requirements (Figure 8.2, *Rafsec G2 ShortDipole* at 868 MHz). The reader used in this setup was a *Feig LRU2000 UHF Long Range Reader*, equipped with a directional antenna. This device (and every other investigated reader) does not provide any tag quality metrics (such as received signal strength indicators), and thus each scan only resulted in binary "detected" or "not-detected" information per tag.

The antenna's gain increases the reading range along a single axis, allowing this system to detect tags up to a distance of 6 m. As illustrated in Figure 8.2 (right and lower left), the transponders in this scenario are distributed within the aircraft's cabin, especially at the hatrack, the seating and the floor panels. In the final deployment in a real aircraft cabin usually every major structural part has its own identification tag, resulting in a far denser distribution. To measure the current orientation, a gyroscope is directly connected to the antenna (Figure 8.4) with a precalibrated offset transformation.

8.4.2 Basic Localization Concept

The localization procedure is based on two sensors: the RFID reader and the gyroscope. The reader's output is a list of tags read by the antenna within a certain timeframe. Note that these measurements only constitute the existence of the relevant tags in the reading range of the antenna and do not contain further distance information. Facing this handicap a more sophisticated search routine was implemented, which takes the antenna's readability pattern into account.

Figure 8.3: RFID Reader

Figure 8.4: Xsens Gyroscope

The developed RFID tracking system has three basic characteristics:

- A large number of RFID tags are attached to structural parts of the cabin. For each of them the exact location and ID are known and available to the system as part of the Digital MockUp (DMU) data.
- A gyroscope measures the current orientation. This is used internally to increase the accuracy of the position estimate and returned as part of the pose estimation.
- The reading range of the RFID antenna is taken into consideration. It has to be determined beforehand.

The localization is based on the readability as well as the coordinates of the currently detected transponders, which can be derived from the database. The antenna is estimated to be at a position from where it is able to read all currently detected tags at their current readability level. This is accomplished through an iterative process which imposes requirements on how the detection range is modeled, as described in the following section.

For this scenario the position of each RFID tag and the corresponding ID was recorded in a preprocessing step. It is assumed that this will become unnecessary in a future scenario by having predefined IDs and positions of tags on certain components or parts. This assumption seems realistic, since RFID technology is in the process of being integrated into the aircraft manufacturing process.

The following algorithm is structured in a way to reduce the computational cost of the pose estimation. This is required since mobile devices, which are the target platform of the maintenance support application, often lack computational power in favor for mobility and uptime.

8.4.3 Antenna Readability Model

Before actually using the system, the detection range of the RFID reader's antenna had to be determined. Due to missing details concerning the radiation pattern on the antenna

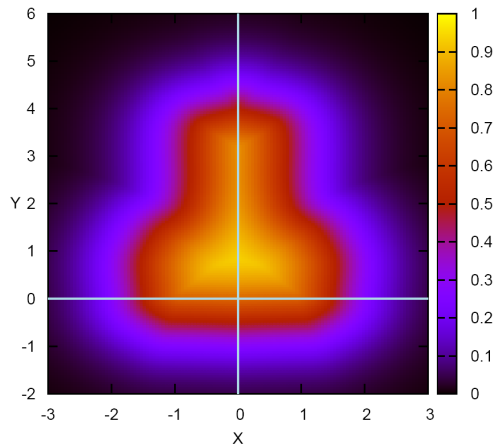


Figure 8.5: Antenna Readability

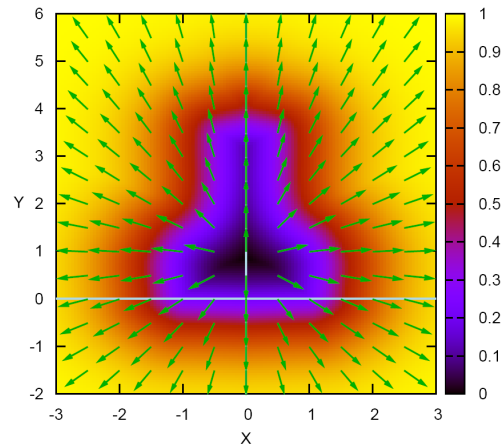


Figure 8.6: Displacement

assembly, an empirical method for experimentally estimating this range was developed.

Since the system uses a directional antenna with a rotationally symmetric RF field, the dimension of the testbed was reduced to a 2D plane through the central axis of the antenna. A number of tags were attached to polystyrene cubes and placed on a grid to evaluate the reading range and probability, resulting in a discrete readability field.

For each point $\vec{p}$ relative to the antenna, the position estimation algorithm requires a scaled displacement vector to be computed from the antenna model. This vector $v(\vec{p})$ indicates in which direction the antenna needs to be moved to fit the probability of detecting a transponder at location $\vec{p}$ to the perceived reading probability. A scalar value $d(\vec{p})$ is derived, defining the intensity of the displacement vector to fit each tag accordingly.

Both will be derived from the reading field of the antenna, containing the probability of a tag being read at a certain position. Since real-time computation of the readability for all tags is computationally expensive, a gradient field was derived by interpolation (Figure 8.5) which serves as a lookup table. The resulting gradient field is used to orientate the displacement vector, which is shown in Figure 8.6 also visualizing a possible scale of the displacement that a positive read at this location would induce. Both fields use coordinates in the antenna's reference frame. The antenna is placed at $\vec{p} = (0, 0)$ and the point with the maximum reading possibility (100 %) is located at $\vec{p}_{max} = (0, 0.85\,\mathrm{m})$.

Optimization of the Position Estimate

The reader scans its environment for visible tags within the antenna's range in slices of $100ms$.

Initially the position estimation assumes that the antenna is in the centroid of the visible tag positions. This is iteratively refined by a simple optimization procedure that

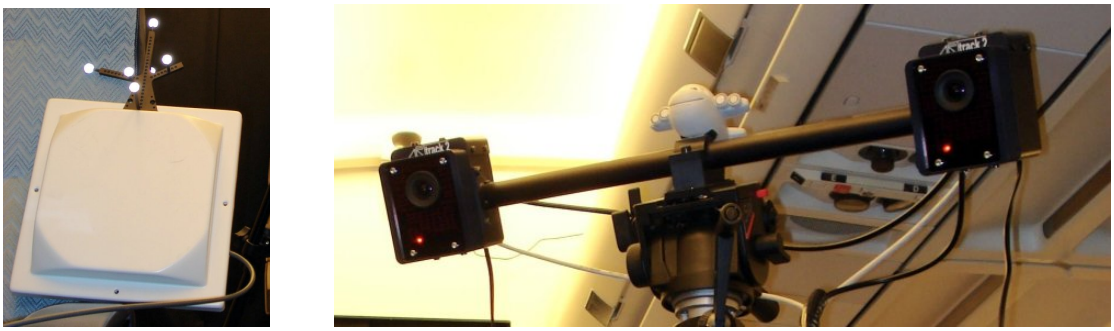


Figure 8.7: Antenna with IR-Marker and *A.R.T.* Tracking System

is performed at each step.

First, the visible tags are rotated into the reference frame of the antenna by applying the inverse of the orientation measured by the gyroscope. This relative position is projected onto the 2D plane and used to access the readability as well as the uniform displacement vector.

In the next step, the median of all displacement vectors is computed and added to the current position resulting in a new position estimate. The estimated quality of this position is calculated as the sum over the readability of all detected tags, taken from the readability field. Iteratively this procedure optimizes the position estimate until a certain number of improvements have been performed or the magnitude of the improvements is below a certain threshold. When finished the position as well as the likeliness with the highest quality rating is returned.

In case no tags have been detected, the last position is retained.

8.4.4 Error Analysis

The setup for the test is shown in Figure 8.7. For the estimation of the ground truth position an optical infrared tracker is used, allowing the reflective marker on the antenna to be localized within millimeter precision. The offset between the infrared reflection marker and antenna was calibrated beforehand. This assures that the entire tracking system used for validation and testing introduces a negligible error compared to the expected precision of the RFID tracking.

The typical performance of the system is discussed on the basis of a one minute excerpt from a test run. The mobile device was moved in the cabin with varying orientations. The currently visible tags and the resulting position estimation of the RFID tracker as well as the ground truth position were timestamped and logged. In a post-processing step the recorded data was examined to determine the precision and to analyze the current setup.

To better distinguish the error induced by the distribution of the tags and the antenna

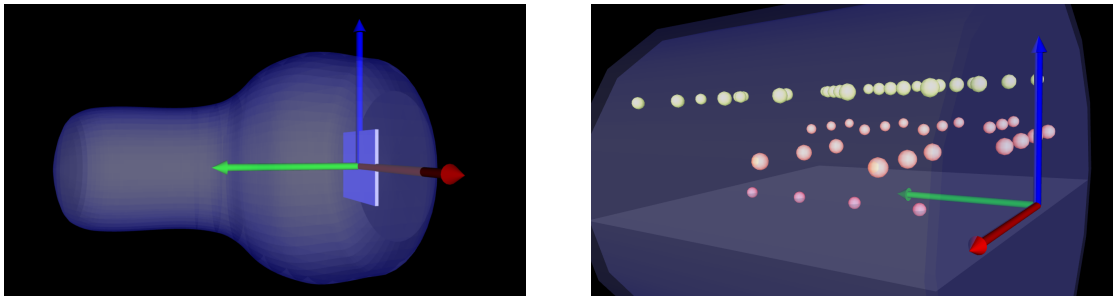


Figure 8.8: Antenna Field (Median Readability) and Cabin Coordinates

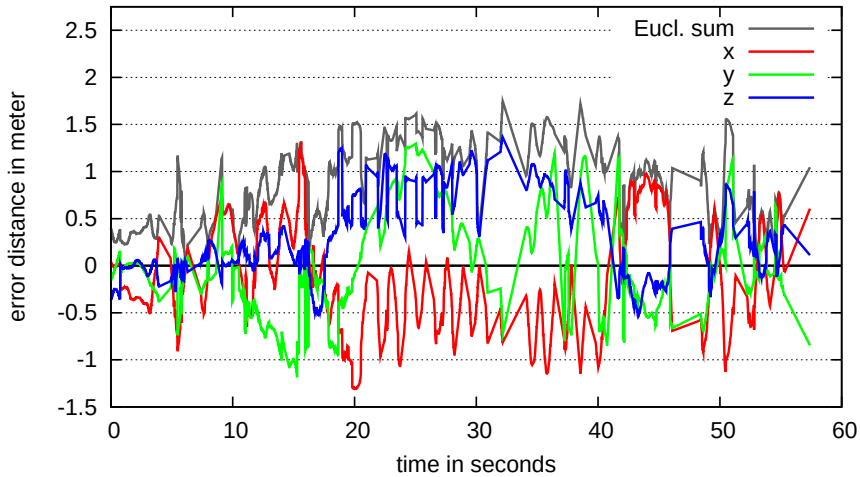


Figure 8.9: Error in the Test Run in the Cabin's Coordinate System

model, the coordinate system's axes are indexed either by *'a'* for antenna or *'c'* for cabin. In Figure 8.8, the cabin has the x_c axis (red) pointing to the side, the y_c axis (green) to the front and the z_c axis (blue) to the ceiling. The antenna's coordinate system has the x_a axis pointing to the side, the y_a along the antenna's normal and the upward axis z_a.

The overall performance of this approach can be seen in figure 8.9. It shows the error distributed along all three axes of the cabin as well as the average error across the entire testing time. The output is the raw data taken from the position estimator without any filtering or continuous position estimation for jitter reduction. The Euclidean distance between the RFID tracker's estimate and the ground truth ranges up to 1.5 m with an average of 0.48 m.

In the beginning of the test run, the antenna was moved while pointing towards the front of the cabin. This results in a small y_c axis error since a more homogeneous distribution of the RFID tags (Figure 8.8, right) allows the antenna model to fit perfectly

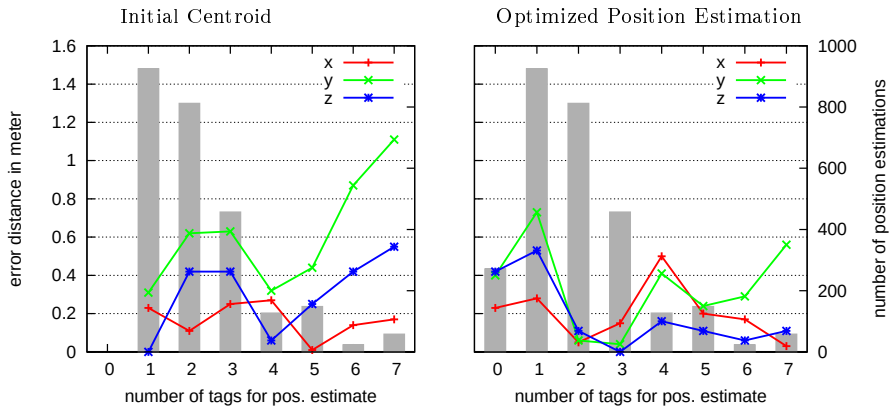


Figure 8.10: Error in the Antenna's Reference Frame (x, y, z) and Histogram (gray bars) with Respect to the Number of Detected Tags

with its long reading range along its y_a axis. Therefore the effect is homogeneous in both reference systems.

In the time range between 10 s and 40 s the antenna was additionally pointed upwards and downwards while still moving through the cabin, increasing the error along the $\vec{z}_c$ axis. The uncertainty along the other two axes as well as the overall error are also increased, since the antenna's reading field is mainly outside of the cabin's volume where no tags were placed.

Within the final time interval beginning at 40 s, the antenna's position was kept still but was rotated mainly around its z_a axis and therefore often pointing along the side of the cabin (x_c axis). This shifts the error between the y_c and the x_c axis. The distribution density of the tags towards the side of the cabin is also insufficient for robust estimation.

Not only are the distribution and the homogeneity of the tags important for a proper function of the algorithm, but the number of tags used for the position estimation is also of interest. Looking at Figure 8.10, it is evident that the simple approach to use the centroid of the tag cloud is inferior to the optimization based on the antenna model. The overall median error is reduced from 0.71 m to 0.48 m. Furthermore the possible uncertainty introduced by the antenna's long reading range in direction of the y_a axis could be compensated with the orientation based method. The errors on the x_a and z_a axis could be improved only slightly.

For the case of zero detected tags the error is high because without measurement information supplied no further estimation can be performed. In the case of a single tag the algorithm's estimate induces an even larger error. This indicates that a tracking using a motion model, based on preceding estimates, could achieve an increased performance since a good initial estimate seems superior to the estimation based on a single tag. Single tag discoveries often occur when the reader is pointing directly into an obstacle,

reducing the range of the RF field.

Although the error should be indirectly proportional to the number of tags this can not be derived from the current test results. As the histogram in Figure 8.10 illustrates, in only about 3 % of the frames more than 4 tags were detected. Nevertheless it seems obvious that the error increases in a high detection case. A possible reason for this can be traced back to the reader buffering recently read tags when a quick rotation is performed, resulting in more detected tags but partially from an outdated orientation.

Although neither the algorithm nor the implementation can be expected to be optimal, the results clarify four main error sources in the system setup:

- inhomogeneous distribution of tags cause the position algorithm to fail
- low density of RFID tags yield an inferior pose estimation
- obstacles changing the antenna's reading probability field increases the error
- accumulation of visible tags over a certain period of time introduces an error when a quick rotation or fast movement is performed

These characteristics need to be considered in future optimizations of the system.

8.4.5 Application and Further Use Cases

For the maintenance application the tracking is precise enough to enable location aware assistance software. It is possible to guide the worker to the correct frame of the aircraft and to prioritize the tasks according to his current position, so that maintenance tasks in the current working area are favored which minimizes time spent between tasks. It is also possible to reduce user input which would occupy the worker.

The use of a tracking system allows to extend the existing workflow of a mechanic. Nowadays it is common to have a digital task list on a mobile device to verify that all important maintenance tasks have been performed correctly. With the additional location sensitivity it is possible to guide the worker towards the next task and automatically give additional information regarding the current step.

As stated before RFID technology is expected to become more common and enter areas of our daily life. As a consequence this also can be seen as a low cost tracking infrastructure becoming ubiquitously available which can lead to numerous applications.

One such envisioned possibility is assisted wayfinding and navigation in supermarkets [VSYZ08] or warehouses. This scenario is further supported by the fact that today many consumer products already start to be tagged by RFIDs, as some preliminary field studies show. Another similar scenario could be the installation of tags in underground parking lots to facilitate customer routing and orientation between parking decks and cars.

8.5 Showroom scenario

The focus of the previous scenario was to derive location information for an industrial maintenance worker inside an aircraft cabin. In that setup the localization is restricted to process only symbolic information about the properties of the infrastructure, and cannot rely on parameters with direct physical correspondence. The RFID reader can only provide binary information about the tags currently present inside the reading volume of the antenna. Instead, the localization implicitly uses the inherent spatial relationships between the anchors provided by the environment. While the dependence on such explicit information on the geometric configuration of the infrastructure is valid in this industrial scenario, it may be disadvantageous in a more volatile environment.

The results gained from this evaluation seem to be encouraging and stress the potential of inter-landmark relationships as additional localization constraints. Nevertheless it is still interesting to consider RFID based location estimation from distance estimates obtained by observing physical communication parameters. This chapter will derive a method to obtain such physical parameters and evaluate a localization method based on these measurements.

8.5.1 Power cycling and analysis

As described earlier, the received electromagnetic power at a receiver is dependant on the distance between transmitter and receiver. Using this model it is possible to estimate this distance by estimating the path-loss between sender and receiver.

To account for the doubled path in case of RFID backscatter communication, it is necessary to modify this path-loss model, as the reflecting RFID tag can be seen as an intermediate receiver.

Starting with the path-loss model as described in section 6.2.2, the signal power at the RFID tag is given by $P_{tag} = c^{(0)} \cdot P_{TX} G_{Ant} G_{tag} \left(\lambda/4\pi d\right)^2$, for some constant $c^{(0)}$. Here P_{TX} is the power transmitted by the RFID-reader, G_{Ant} the antenna gain of the reader in the direction of the tag, G_{Tag} the antenna gain of the tag, λ the wavelength of the radio wave and d the distance between tag and reader.

As mentioned the RFID tag answers a request by modulating the matching of its antenna and thus changing the amplitude and phase of the reflected electromagnetic wave. This is called backscatter modulation. Assuming the tag could use all of the incoming power to answer, the power arriving at the RFID reader can be written as

$$\begin{aligned} P_{RX} &= c^{(1)} \cdot P_{tag} G_{tag} G_{Ant} \left(\frac{\lambda}{4\pi d}\right)^2 \\ &= c^{(1)} c^{(0)} \cdot P_{TX} G_{Ant}^2 G_{tag}^2 \left(\frac{\lambda}{4\pi d}\right)^4 . \end{aligned}$$

Note that this is analog to the equation derived in [Fin02].

If this equation is solved for the distance, it can be written as

$$d = \frac{\lambda}{4\pi} \cdot \sqrt[4]{\frac{c^{(1)}c^{(0)} \cdot P_{TX} G_{Ant}^2 G_{tag}^2}{P_{RX}}}.$$

By assuming that the frequency and the antenna gain at the tag are constant, this can be simplified to

$$\begin{aligned} d &= c' \cdot \sqrt[4]{c^{(1)}c^{(0)} \cdot G_{tag}^2} \sqrt[4]{G_{Ant}^2} \sqrt[4]{\frac{P_{TX}}{P_{RX}}} \\ &= C \cdot \sqrt{G_{Ant}} \sqrt[4]{\frac{P_{TX}}{P_{RX}}}, \end{aligned}$$

for some constant $C = (\lambda/4\pi) \sqrt[4]{c^{(1)}c^{(0)}G_{tag}^2}$.

Furthermore the actual received signal power is not assumed to be known in this scenario. Nevertheless in case of a successful communication between RFID reader and tag, the reflected signal as received by the reader, must be above some minimal signal power $P_{RX,min}$, associated with the sensitivity of the reader. If the communication was successful, it must hold that $P_{RX} \geq P_{RX,min}$ and thus $P_{TX}/P_{RX} \leq P_{TX}/P_{RX,min}$. This leads to

$$\begin{aligned} d &\leq C \cdot \sqrt{G_{Ant}} \sqrt[4]{\frac{P_{TX}}{P_{RX,min}}} \\ &= C' \cdot \sqrt{G_{Ant}} \sqrt[4]{P_{TX}}, \end{aligned}$$

with $C' = C \cdot \sqrt[4]{1/P_{RX,min}}$, since $P_{RX,min}$ is also constant.

Thus, in case that the actual received power is not known, an upper bound for the distance between RFID-reader and tag can be derived. The intuitive reason for this upper bound is that in a sense if a communication between the reader and a tag is possible, the tag "cannot be too far away".

Similarly, in the case of a negative communication outcome for a certain transmission power, a lower bound for the distance between reader and tag can be derived using the same equations. For successful localization, these bounds should be applied conservatively in the sense that the constant C' in both cases should be chosen in such a way that the lower bound does not overestimate and the upper bound does not underestimate the actual distance. The range of this constant can be determined experimentally.

Power cycling An arbitrary transmission power in general gives neither a sharp lower bound, in case of no communication, nor a sharp upper bound, should the tag be detectable. This is because in general an arbitrary power setting is neither the minimum power required for communication, nor the maximum power at which no communication

is possible. In order to improve the expressiveness of a single tag, the transmission power of the RFID reader is periodically changed in a cyclical pattern. For each power setting, a complete tag detection in the environment is performed. This is possible for a reader employing SDR technology.

The response behavior of a tag depending on the transmission power can be classified as a sharp threshold. Idealized, for any power setting below this threshold, no communication is possible, whereas any setting above the threshold results in certain communication. This threshold transmission power will be called the critical power value and is the value producing both the best upper and the best lower bound for the distance between reader and tag.

In reality the behavior is not that ideal and there usually is a certain range of transition from zero reception probability to one. To determine the critical power value for a tag, a histogram of multiple reception trials at different power settings can be used. The critical value is estimated as the power setting where the number of successful detections of the tag drops below 70% percent of the total number of possible detections at a single power setting. This assumes that only complete power cycles are performed, and thus the number of trials at each setting is equal. Furthermore hysteresis could be used to employ different power values for the lower and the upper bound.

Further analysis and localization By determining the critical value for each tag which was successfully detected, separate lower and upper bounds for its distance can be obtained. Since the location of the tags is also assumed to be known, the user's location can be estimated by the intersection of the regions indicated by the individual estimates. Each anchor together with lower and upper bounds for the distance to the user, defines the shell of a sphere with thickness equal to the disagreement between upper and lower bound, and which is centered around the location of the tag.

To determine the area in space which best corresponds to the location of the user, a weighting according to the number of agreeing evidence is applied. Each detected tag generates a positive sphere around the tag's location with the upper bound for the tag distance as radius. This indicates that the user has to be inside this radius according to the evidence produced by this tag. Similarly each detected tag generates a negative sphere, that is a sphere contributing negative evidence, around the tag's position, but with the lower bound as radius. This indicates that the user has to have a certain minimum distance to the tag, due to the necessary critical power value. Note that these two radii still differ, due to the range of the constant C' above. This analysis is similar to the estimate employing positive and negative circles used for Wi-Fi localization.

Finally tags which were not detected at all during a number of cycles could be treated as evidence that the user has to be outside the maximum reading range from the tag. This is equivalent to a negative sphere of radius corresponding to the lower bound at maximum power. In practice, this assumption is too strong, as there could be a number

of different effects preventing communication with a tag, for example shadowing. Thus not detected tags generate an exclusion zone, in the form of a negative sphere, albeit of rather small radius. The idea behind this approach is that in immediate vicinity of the tag, communication is almost guaranteed, especially at higher power settings. Obstacles are unlikely to co-inhabit this range with the user and thus shadowing is unlikely. Nevertheless the radius of this area is chosen to be rather close to the tag.

To finally determine a localization from this weighting, a threshold is applied, either to the 3-space or to a 2D projection, depending whether 3D or 2D localization is desired. This threshold is applied to remove areas of low probability or for which only a small amount of evidence exists. The location of the user can then be calculated as the center of mass of this weighted region.

To further quantify the uncertainty of this location estimate, a principal component analysis (see for example [Fod02]) of the thresholded space can be performed. This estimates the variance of the remaining area in multiple dimensions.

Estimate orientation As described above, the estimate of the distance between sender and tag is dependent on the antenna gain in the direction of the tag, as seen from the antenna's orientation. In the ensuing discussion, this effect was implicitly assumed to be constant and was not further discussed. This is accurate for omni-directional antennas, or antennas with weak directionality.

For antennas with pronounced directionality, on the other hand, this effect could be used to both increase the confidence of the location estimate and to estimate the orientation of the antenna. The gain characteristics of antennas are usually specified in a spherical coordinate system. Thus the distance estimate becomes

$$\begin{aligned} d &\leq C' \cdot \sqrt{G_{Ant}} \sqrt[4]{P_{TX}} \\ &= C' \cdot \sqrt{G_{\Theta,\phi}} \sqrt[4]{P_{TX}}, \end{aligned}$$

where Θ, ϕ are the inclination respectively the azimuth of the tag in the antenna coordinate system. With known orientation, this can be used as an additional distinguishing factor to improve the distance estimates.

To estimate the antenna orientation itself, the uncertainty of the location estimate can be minimized over the parameter space of all possible orientations. The rationale behind this is the assumption that an incorrect orientation will produce distinguishably more contradictions than the correct one. This approach is more suitable for antennas with pronounced directionality, for example yagi or log-periodic antennas. Finally, since most antenna radiation patterns are symmetric concerning at least one axis, such an estimate must be ambiguous in these symmetries.

8.5.2 Setup

To explore the feasibility of the approach as described above, a set of different experiments were conducted. In the following the common experimental setup is described.

Location and infrastructure The experiments were conducted in a computer science showroom containing standard hardware encountered in a desktop environment. While most of the mobile equipment was removed from the area during the experiments, furniture and stationary equipment, such as computers or displays, were kept in place. The goal was to not simulate a perfect undisturbed setup, but to evaluate in a real-world scenario including obstacles and distortion. Overall the cross section of the available area was $3\,\mathrm{m} \times 7\,\mathrm{m}$ and the height of the room was also $3\,\mathrm{m}$.

To simulate the RFID infrastructure, a number of RFID tags were distributed throughout this environment. The tags were attached to two perpendicular walls, as well as to furniture. Altogether 75 tags were distributed.

Figure 8.11: Experimenation area with RFID tags

The RFID tags used were "GEN II 1X3.5" passive UHF tags from Texas Instruments. Early tests showed that tags directly attached to metal surfaces greatly suffered in readability, up to the point that they could only be detected by placing the antenna directly overhead. To improve this, the tags were attached to small wooden supports instead, which in turn were fixed to the intended location.

To obtain reference data, each tag, as well as the planes of the walls, were measured using the Total-Station, as was already mentioned previously. This produced a model of the tags in the room with precision of about 1mm (see figure 8.13).

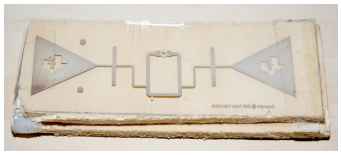


Figure 8.12: RFID Tag on wooden support

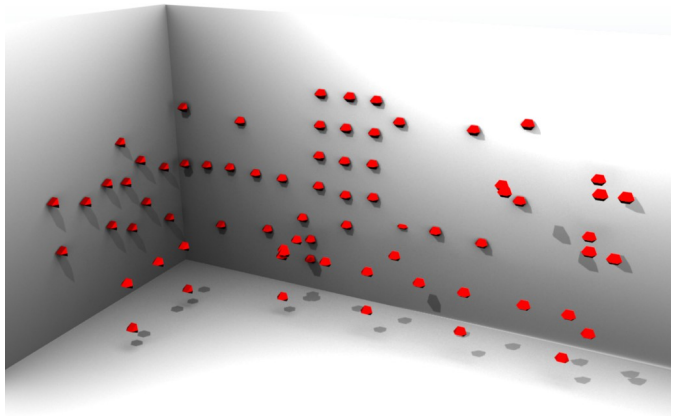


Figure 8.13: Recorded tag locations and wall planes

Reader Equipment The RFID reader used in the experiment was a "ThingMagik Mercury M5e" embedded RFID reader (see figure 8.14). One of the main features of this reader is that it employs software defined radio internally. While direct access to the reception parameters in not possible in this case, the reader offers good control over the transmission process. Especially the transmission power is adjustable in a wide range (from 25mW to 1W) and doing so is sufficiently fast. This is a distinct advantage of this kind of RFID reader and enables the power cycling strategy described above.

The associated antenna used was a Centurion MicroSphere, omnidirectional antenna. This antenna offers no pronounced directionality and thus the orientation estimation was not evaluated. Instead, the following experiments focused on localization.

The reader was controlled by the Ubitrack framework, which also captured the data produced during the tests.

8.5.3 Long-term experiment

The goal of the first experiment was to determine the characteristics of the tag detection in a typical setup. In this context this includes the estimation of suitable ranges for the

Figure 8.14: Mercury M5e RFID reader and MicroSphere antenna

constant C' defined above, for both the lower and the upper bound. Furthermore the Mercury M5e RFID reader offers up to 2000 different power settings and each complete scan of the environment at each setting takes about 2.5 seconds. Thus the number of different settings used in a localization experiment has to be reduced and for this a number of relevant and distinct settings has to be identified. Finally the stability of the reception of the individual tags can be investigated.

Setup The setup of this experiment additionally placed the RFID reader and its antenna inside the experimentation area, at an arbitrary position that is typical for further experiments and not pathological. This position was also recorded using the total station.

The transmitted power was cycled over the entire possible range in the smallest increments possible. Each increment increased the transmitted power by 1/100 dBm, ranging from 1mW (10 dBm) to 1W (30 dBm). Thus each cycle required about 5000 seconds or 1.4 hours.

The complete runtime of the experiment was 18 hours, thus at least 12 complete cycles were performed. During the runtime of the experiment the setup was left completely undisturbed.

8.5.3.1 Results

The first result of this experiment identifies distinct transmission power values to be used in further localization experiments. A histogram showing the number of detected tags versus the current power setting is shown in figure 8.15.

This figure shows that certain ranges of power setting form plateaus of no significant change in the number of detected tags. Note that for the presentation in figure 8.15 the scale has been adapted to watts instead of dBm. In the parameter space as accepted by the RFID reader, the plateaus at low transmission power are accordingly larger due to the logarithmic nature of the decibel scale.

From this histogram a set of 14 distinct power values corresponding to the different plateaus were identified. These values are used as the power cycle in further experiments,

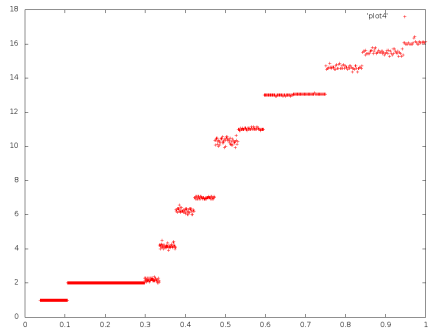


Figure 8.15: Number of detected tags vs. transmission power

thus leading to cycle times of about 35 seconds.

For each tag the critical power value was computed as well as the true distance between tag and antenna. By comparing these two values using the distance estimation equation it was possible to empirically estimate the range of the constant C'. Using the lower end of this range for estimation of the upper bound on the distance of the tag and similarly the upper end of the range for the lower bound, a pair of conservative estimates were obtained.

The range of C' was determined to be concentrated at 1.5 to 3, while a noticeable amount of outliers ranged from 0.8 up to 8. For the further localization the range 1.5 to 3 was finally accepted to be used.

8.5.4 Localization Experiment

The next experiment evaluates the feasibility of locating the user from RFID tag measurements using the previously described model and the experimentally derived constants.

Setup The setup for this experiment was similar to the long-term experiment described above. The RFID reader and its antenna were placed at 5 different locations inside the experimentation area and the positions as well as the orientation of the antenna were recorded, again using the total station.

At each location ten complete power cycles of the previously identified 14 values were performed during a runtime of 350 seconds. The IDs of all tags detected at each trial were recorded together with the currently selected power value.

After collecting all experimental data, the localization process as described above was performed for each data set. The data gathered at the individual locations was processed to estimate the 2D location of the antenna in the room.

For each data set, the critical value of each visible tag was determined. In order to increase the resolution of the critical value, the histogram of the number of successful detections versus power value was furthermore interpolated. Using the critical value the respective lower and upper bounds were determined and negative respectively positive spheres were generated. For tags which were not detected at all during a certain test run, a negative sphere of radius 0.7 m was additionally generated.

Since the experiment focuses on 2D-localization, the space was finally projected onto the X-Y-plane and the areas for which only a small amount of evidence exists were eliminated. The location of the user was determined as the barycenter of the remaining mass. In order to classify the uncertainty of the localization, the spread of the remaining area is characterized by the deviation along the basis axes as determined by principal component analysis. The convention used here is that the axis with the larger deviation is designated as the first axis.

8.5.4.1 Results

The overall result of the experiment was satisfactory, albeit the localization was not perfect in each case. A summary of the results is given in Table 8.1. The table shows the absolute difference between the estimated position and the reference data as well as the deviation along the first and second axis as determined by the PCA.

Experiment	Localization error	1. axis deviation	2. axis deviation
Experiment 1	0.56 m	0.59 m	0.17 m
Experiment 2	0.78 m	0.34 m	0.15 m
Experiment 3	1.18 m	0.31 m	0.11 m
Experiment 4	0.45 m	1.26 m	0.29 m
Experiment 5	0.41 m	0.26 m	0.15 m

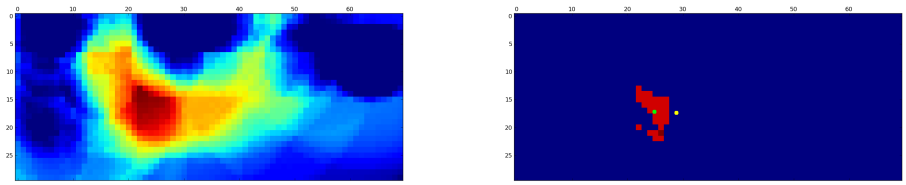
Table 8.1: RFID localization results

In some of the well-conditioned cases the localization worked adequately, resulting in errors less than 0.5 m. Figure 8.16 shows the projected space as well as the thresholded mass for such a case. In the thresholded images below, the true position of the antenna is marked in yellow and the final position estimate in green.

Other cases resulted in acceptable localizations but with large and deformed uncertainties. In that case the intersection of two large spheres results in a stretched area. Figure 8.17 below shows such a case.

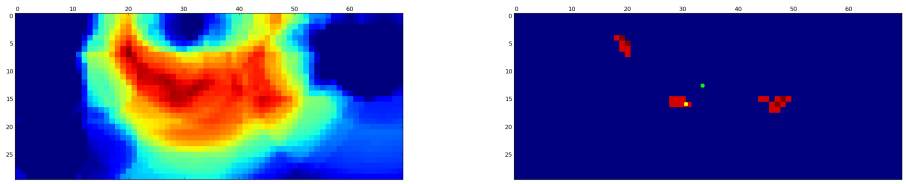
Finally some cases also resulted in unsatisfactory localizations, with error above 1 m. For these cases the assumed ranges of the constants or the model itself fail, and these cases should be further investigated. Figure 8.18 shows such an example.

The overall mean error was 0.67 m and the mean uncertainty along the first axis was 0.55 m.



(a) (b)

Figure 8.16: RFID localizaiton example – 0.4 m error

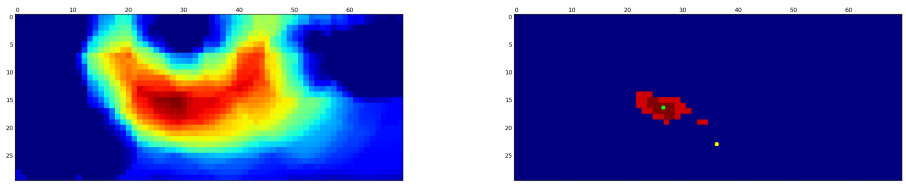


(a) (b)

Figure 8.17: RFID localizaiton example – large uncertainty

Conclusion Thus in conclusion it was shown that pure, range-based localization using detection of RFID-tags at different transmission power settings is possible. This method produced comparable results as the previous scenario, but without the additional use of a gyroscope.

Further use of directional antennas or the more precise characterization of the involved constants could lead to improved localization confidence and accuracy.



(a) (b)

Figure 8.18: RFID Localizaiton example – 1.26 m error

9 Optical odometric tracking

The work described in this chapter was partially funded by the Trackframe project and partially by the AVILUS project and was conducted in cooperation with Michael Schlegel. Furthermore parts of the description below are based on the paper [WHK+10] presented at ISMAR 2010 and the corresponding demo presentation at ISMAR 2008.

Further research on the "computer vision optical odometric tracking setup" topic was supported by hardware donations from A.R.T. GmbH.

9.1 Odometric tracking scenario

There is a certain number of commonly recurring types of situations encountered in different UAR scenarios. One such situation is that the scenario may be divided into different areas which exhibit different requirements regarding tracking accuracy and availability. Usually these scenarios feature areas where high tracking accuracy is necessary in order to provide appropriate support to the user for the relevant task. For this purpose the installation of dedicated tracking infrastructure, albeit with reduced working area, is sensible.

Between these high-precision areas usually only tracking with reduced accuracy is required in order to provide navigation information or to continuously monitor the flow of goods. The space between the high-precision areas is usually much larger than the high-precision areas and in general no dedicated tracking infrastructure can be assumed.

These areas can be seen as "islands" of high-precision tracking surrounded by large spaces with no tracking information. It is thus desirable to use parasitic tracking methods to bridge these gaps and enable non-interrupted tracking even between these islands.

In the following, two concrete scenarios are summarized briefly.

Forklift navigation The first scenario is derived from one of the scenarios explored by BMW in the Trackframe project.

In this scenario the primary task of the user consists of transporting automobile parts or production goods inside a large production facility or workshop. The production goods in question are assumed to be large and have to be transported by a forklift. The user is thus connected to a mobile platform which can serve as the basis for both localization and augmentation.

The setup of the workshop is outlined in figure 9.1 and consists of four basic types of areas encountered.

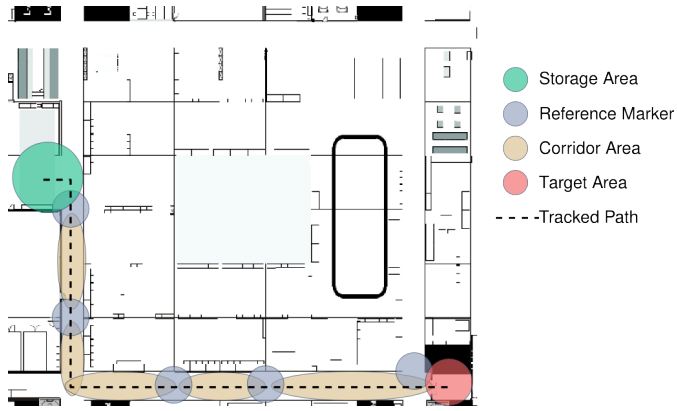


Figure 9.1: Exemplary layout of workshop area

Storage area At the storage area the user is guided as to which specific item should be picked up and delivered to another part of the workshop. Since the correct selection of the specified item is crucial and since this area is of limited space, a dedicated tracking infrastructure can be installed.

Target area The destination of the good to be delivered is specified by the target area. There the user has to deposit the item which was previously picked-up. Depending on the individual target areas, the precise location where the good is deposited can either be crucial or not. Thus dedicated tracking can either be present or absent.

Corridor area All the remaining space, which is neither storage nor target area can be considered to be corridor area. This space has to be traversed in order to travel from the storage to the target area. While it is assumed that no tracking infrastructure is available in this area, the user should nevertheless receive navigation information in order to reach the target area. Furthermore, uninterrupted monitoring of the production goods is desirable for logistics purposes. Parasitic tracking has to be used in this area.

Reference anchors To supplement the parasitic tracking in the corridors, sparse reference anchors can be fixed to known positions in parts of the corridor area. By sensing these reference positions the accumulated error of the tracking platform can be reduced at certain points. These references for example can be realized as optical markers.

The scenario thus consists of two islands of high-precision tracking (the storage and the target areas), where the user can be guided by full augmented reality displays. Between these two areas there is a large gap of uninstrumentalized space where reduced AR interaction should be maintained using parasitic tracking methods. To reduce tracking

error and to provide reinitialization opportunities, additional reference anchors are fixed at known positions. Using dynamic sensor fusion and tracking reconfiguration the change between the different areas should be seamless and uninterrupted operation should be maintained.

Tracking competition The second scenario is derived from the challenges presented at the ISMAR Tracking Contests 2008 and 2009. At these contests a reference coordinate system of the contest room was given to the participants in the form of a set of visibly marked points in the room together with the corresponding 3D coordinates. The contest room furthermore exhibits different challenge booths, which usually contain a set of similar items at different positions. The contestants are given half a day for setup and calibration of their tracking systems.

After setup the actual challenge is presented as a list of unknown 3D points in the room, which specify points at each of the booths, usually referencing single objects. The unknown points are different for each contestant, who has to correctly identify the individual items using only augmented reality overlays. Each correctly identified item earns a point for the contestant. In the case of a tie (equal number of correct items) the speed of the respective contestants decides the winner. Thus, it is also beneficial to be as fast as possible.

Whereas the first contest 2008 focused more on academic examples for difficult tracking situations for feature-based tracking, the 2009 contest included booths inspired by industrial scenarios.

On the one hand, the scenario requires very precise tracking at the individual booths, whereas on the other hand quick navigation between the booths is also desirable. This leads to a possible combination of a feature tracker with smaller local maps and a global tracking system with reduced accuracy requirements. This global tracking system can be used for navigation between the feature maps and to select the correct map, thus improving the performance of the feature tracker. Figure 9.2 shows the resulting multi-sensor tracking platform, which is also described in [WHK+10].

Optical odometric tracking The parasitic tracking system that was used in these two scenarios and that is to be described in the remainder of this chapter is called an *optical odometric tracking system*. It exhibits moderate tracking accuracy at very high robustness, thus being a good candidate for the situations described above.

The system consists of a number of integrated optical flow sensors which are fixed around the edge of the sensor platform. The sensors themselves are facing down and register the movements of the platform relative to the floor, thus describing the pose of the platform in three degrees of freedom: These are two directions of translation and the orientation of the platform. By using multiple but rigidly connected sensors, the task of determining the platform's movement is simplified since here it suffices to determine only

Figure 9.2: The carriage mounted with tripod, sensors and computing unit

two translations at each sensor. Apart from that, using more than two sensors increases the robustness by providing redundant sensor data.

The measurements of the distributed sensor modules are collected by a central control unit and forwarded to the host computer, where the tracking data is computed.

9.2 Related Work

Odometric tracking using mechanical sensors is a commonly used localization method, especially in the area of robotics. A concise description of this tracking modality and its associated error sources is given for example in [BF96b]. Similarly the combination of odometric tracking data with other spatial sensors has previously been explored for robots. For example, [BF96a] describes the combination of robot odometric tracking with gyroscope data.

A similar approach using optical, odometric sensors was used for the navigation of the "Planar" VR platform for industrial building acceptance [SS05]. This system used only two sensors and computed the platform movement using an explicit, geometric formulation of the problem. In a later prototype of the "Planar" system this tracking modality was apparently removed, in favor of outside-in infrared tracking (see [SS08]) or infrared, laser-based inside-out tracking (see [SM10]). Another approach to use an optical, odometric sensor for localization of a robot is presented in [LS04]. In this setup a single sensor is used and also the working distance of the sensor is not adjusted to the use-case, thus requiring very short distances between sensor and floor. Since only a

single sensor is used, the tracking computation requires the additional assumption that movement of the robot is restricted to the direction perpendicular to the axis of the robot's wheels and also requires monitoring of the robots differential drive to determine the orientation.

Other uses of the same optical flow sensing technology and the combination of multiple sensors includes the design of various user interfaces, as for examples in [TEHK09]. The same integrated optical flow sensors were used to construct two user input devices featuring a trackball or soap-metaphor respectively.

A further study of the operating parameters and performance of an optical flow sensor is given in [KKDK07]. The optical flow sensors are tested for measurement errors compared to an actuated system using a controlled, rotating disc with various surface textures.

9.3 Construction

The complete optical odometric tracking system consists of three optical sensor modules connected to a central control unit which communicates with the host computer. Each sensor module (Figure 9.3) in turn is composed of an integrated motion sensor, suitable optics and an illumination module. These parts are mounted in a durable metal encasing to ensure robust operation.

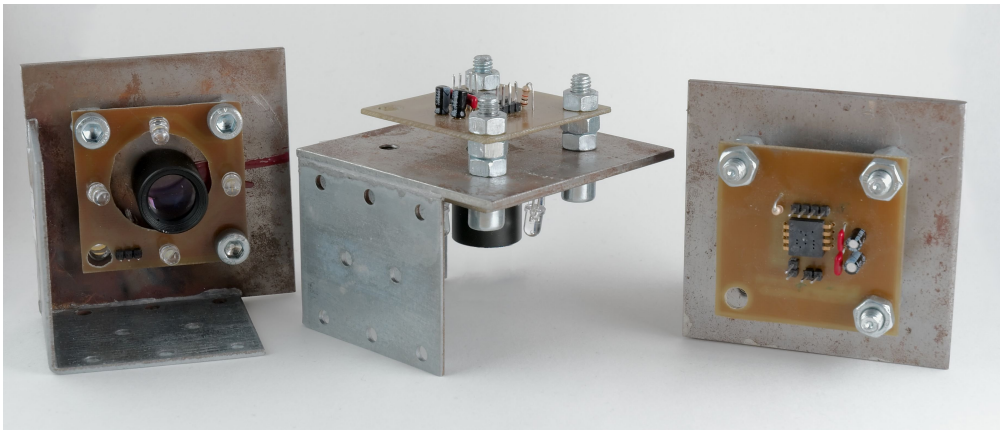


Figure 9.3: The assembled sensor modules

These individual parts will be shortly described below.

Optical-Flow Sensor The actual individual sensor is designed around the *Avago ADNS-5020* (see [Ava08]) integrated optical flow sensor. This sensor is the same kind of optical flow sensor usually found in optical computer mice. It features a high-speed 15×15

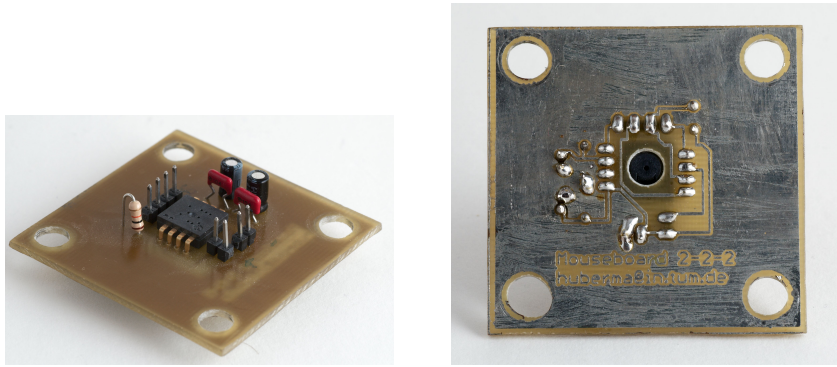


Figure 9.4: Sensor board

pixel sized imaging sensor with 3000 frames per second frame-rate and a computation engine which determines the motion of the sensor between two frames in integral units or "ticks". This computation is performed by a correlation of two consecutive frames and maximum extraction. The sensor is mounted on a circuit board with a minimum number of discrete external components as well as a connector attached to the three-wire synchronous SPI port of the sensor.

The translation of ticks into physical distances depends on the perceived scaling of the sensor and will be calibrated later.

Figure 9.5: Telephoto lens

Telephoto lens The height at which the sensor module is mounted on the platform is critical to the smooth operation of the system. If the motion sensor is mounted too close to the floor dirt tends to accumulate on the unit, reducing its effectiveness. Also pebbles,

floor bumps or tilting of the platform can damage the sensor unit. On the other hand, if the sensor unit is mounted too far away from the floor, it requires an unreasonable amount of space beneath the sensor to be free of obstructions. Otherwise the continuous visibility of the floor cannot be guaranteed. As a consequence, an acceptable distance of about 20 cm was identified. To make the operation of the optical flow sensor possible at this distance, a miniature telephoto lens with 25 mm focal length was mounted in front of the optical sensor, resulting in a visible area of about 2 mm × 2 mm on the floor.

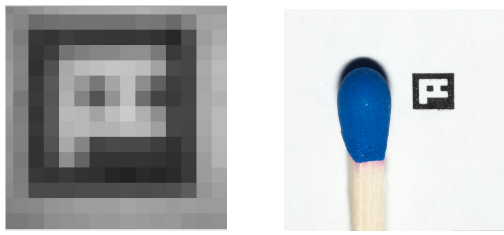


Figure 9.6: Sensor debug image

The correct positioning of the lens was adjusted and verified using debug images from the optical sensor. While the correct adjustments could only be determined with some uncertainty (see figure 9.18) due to the low sensor resolution, later evaluations showed that the overall system is robust against these uncertainties.

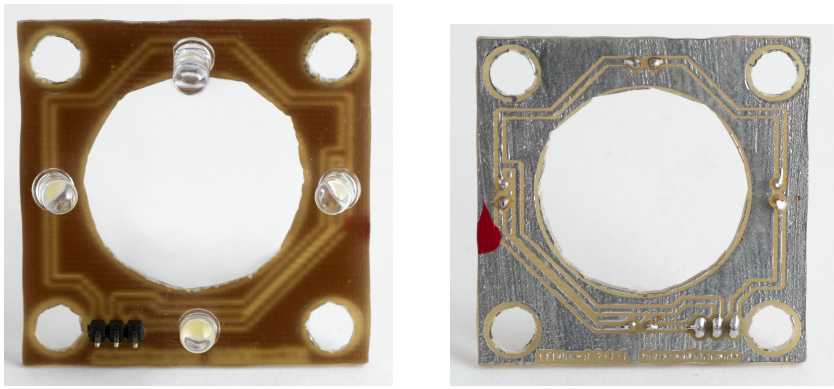


Figure 9.7: LED Board

Illumination To ensure sufficient and constant illumination of the floor area beneath the optical sensor, a ring of four high-power white LEDs was placed around the lens. The orientation of the LEDs was adjusted such that the brightest spots converge at the observed area of the sensor. For better stability and robustness of the illumination,

the LED configuration of each sensor module was driven separately by independent constant current sources. This reduces component dependent variations and aging while maintaining high light output.

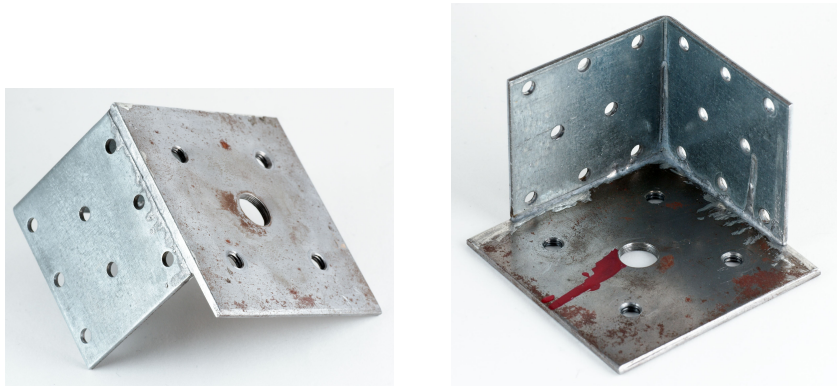


Figure 9.8: Mounting bracket

Mounting bracket The optical sensor together with the lens and the illumination module is mounted in a stacked architecture on a robust metal plate. This assembly is furthermore encased by metal brackets to ensure ruggedness.

The base consists of a 3 mm iron plate into which five threaded holes were drilled. The center hole serves as the opening for the optical path of the sensor so that the optical flow sensor can be mounted directly above it. Moreover, the telephoto lens is mounted directly into the center hole (S-mount, M12x0.5). By screwing the lens more or less deeply into the base, the focus of the sensor can be adjusted. The remaining four holes are used to stack the sensor PCB above and the LED PCB below the base with M6 screws.

In order to provide a mounting point to attach the sensor unit to the platform, an iron angle is welded to the base plate. This construction further serves to protect the lens and sensor assembly from external forces and makes the unit robust against bumping into the environment.

Processor board The individual sensor units are connected to a central controlling board, which consists of an *Atmel ATmega168* microcontroller and peripheral electronics. The firmware for this microcontroller was written in C++ and compiled with the AVR-GCC port.

The microcontroller is connected to each sensor unit by a separate synchronous three-wire serial interface, so that all sensors can be queried in parallel. The SPI interface consists of the signals SDIO (Serial Data Input/Output), SCLK (Serial Clock) and nCS

Figure 9.9: Controller board and enclosure

(Chip Select, active low). Since each sensor is connected by an extra interface, all sensors can be selected in parallel and thus all nCS-lines are tied low. To ensure perfect synchronicity when reading movement data from the sensors, the SCLK line is not actually duplicated, but rather all sensors are driven by a single clock. The operation of an SPI communication with a single sensor is depicted in figure 9.10, whereas the simultaneous multi-sensor communication can be seen in figure 9.11.

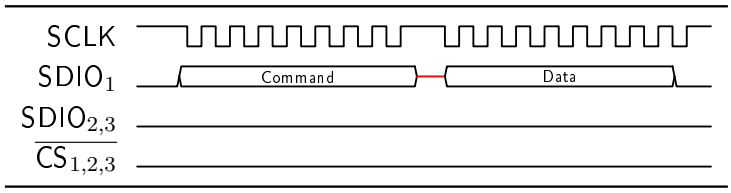


Figure 9.10: Single Sensor SPI operation

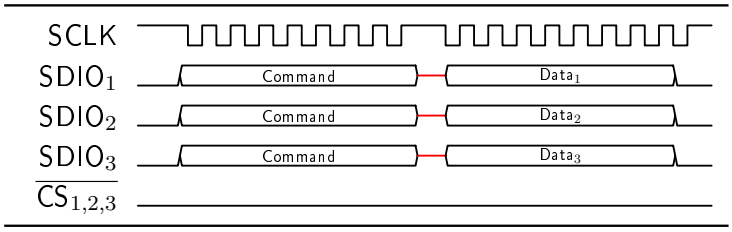


Figure 9.11: Multi-Sensor SPI operation

After receiving data from the sensor units, the microcontroller associates these data with a generated timestamp and sends the annotated raw data to the host computer. A *Lantronix XPort* ethernet adapter is used for communication between the microcontroller and the host computer. The XPort directly translates an asynchronous serial communication into a TCP stream, which can easily be captured by the host computer, for example using the Ubitrack framework.

The timestamp is generated by a constantly running hardware counter which is configured to count the number of milliseconds since the initialization of the microcontroller.

Apart from the primary task, the controlling unit manages further utility tasks such as initialization of the system or acquisition of debug information or debug images.

The controller board also includes a set of eight independent constant-current power sources to drive the illumination LEDs and a further switched-mode power converter which is used for regulating the power supply.

9.4 Calibration and Tracking Using Cooperative Sensor Fusion

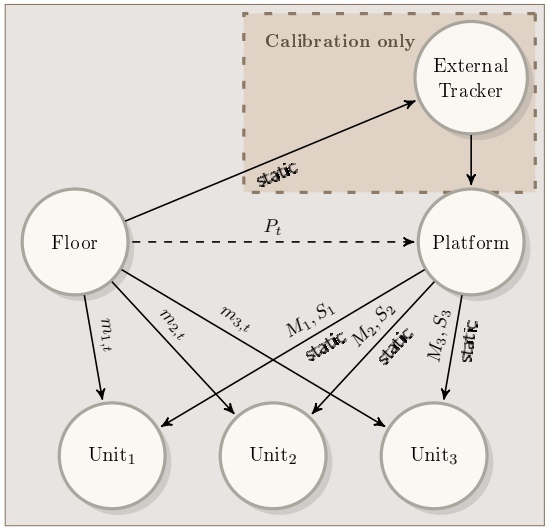


Figure 9.12: Odometer System Spatial Relationship Graph

To derive the tracking and calibration procedures, first the concrete relationships between the coordinate frames involved have to be considered. The situation is illustrated in figure 9.12.

The individual sensor units are fixed on the platform and are represented by the static relationships M_1, M_2 and M_3. The platform itself can move relative to the floor. Its 3DoF floor-pose at time t is denoted as P_t. The actual motion sensor data $m_{1,t}, m_{2,t}$ and $m_{3,t}$ represents the perceived motion at time t of the floor as seen by the individual sensor units. Furthermore the scaling factors of the individual sensor units are represented as S_1, S_2 and S_3.

Let each 3DoF floor-pose X be represented as a 3×3 homogeneous matrix

$$X = \begin{pmatrix} \cos\gamma & -\sin\gamma & x \\ \sin\gamma & \cos\gamma & y \\ 0 & 0 & 1 \end{pmatrix},$$

where x, y represent the two dimensional translation and γ the rotation angle around the axis perpendicular to the floor surface. For ease of notation the 2DoF measurements of sensor units are also represented in the same way although with γ set to zero and the scaling factors are represented as 3×3 diagonal matrices where the third diagonal element is 1.

This leads to the following equation, which relates the movement of the platform to the observed sensor data at time t. This equation is the motion equation of the odometric

sensor system. As mentioned before, note that the motion sensor data is inherently relative, that is, the sensor observes its relative motion from frame to frame.

$$\left[S_i^{-1} M_i^{-1} P_{t-1}^{-1} P_t M_i\right] (0,0,1)^T = m_{i,t},$$

at time t and for sensor $i = 1, \ldots, 3$. Note that the multiplication by the $(0,0,1)^T$ vector represents projection of the 3DoF floor-pose into the 2DoF sensor data, effectively discarding the rotational part. Furthermore this holds true regardless of the number of sensors involved. While the current setup consists of three units, additional units could be manufactured and integrated at any time.

This equation can be used for both the calibration of the system as well as the actual tracking during normal operation.

For calibration an additional optical outside-in tracking marker was fixed on the platform and registered to the coordinate frame of the odometric system.

By simultaneously capturing the sensor data and the external tracking data, a large number of motion equations can be collected. Solving this set of equations for $M_{1,\ldots,3}$ and $S_{1,\ldots,3}$ yields constellation of the sensor units and the associated scaling. The equations are solved such that the sum of squares of the residuals (difference between left and right-hand side) of the equations is minimized. For the calibration the total number of degrees of freedom to be determined is 15 (for each sensor unit a 3DoF pose on the platform and additionally two degrees of freedom for scaling).

This number can be reduced by assuming various degrees of scaling uniformity. In the most general case each sensor has two independent scaling factors and all sensors are considered independent from each other. By assuming that the scale perceived of an arbitrary sensor does not depend on the direction, this could be reduced to one scaling factor instead of two per sensor. These cases are called "isotropic" (one factor) or "non-isotropic" (two factors). A different assumption is that the scale of the different sensors is equal, thus reducing the multiplicity resulting from the number of sensors (three in the current setup) to one global scale. Similarly these cases are called "uniform" (one scale) or "non-uniform" (number of sensors scales). An evaluation described below suggests that "non-uniform isotropic scale" shows the best trade-off between simplicity and accuracy. Thus, the number of degrees of freedom for the scale is assumed to be three (number of sensors times one scalar), resulting in 12 degrees of freedom overall.

The motion equation can also be used to estimate the movement of the platform from the sensor data. In this case three equations per update are established and similarly solved in a least squares manner. The main difference to the calibration setup is that now the optimization parameter is the relative motion of the cart $P_{t-1}^{-1} P_t$, whereas M_i and $S_i; i = 1, \ldots, 3$ are fixed.

9.5 Operation

Constellation calibration The operational part of the calibration procedure is currently implemented as a package for a computational software suite and has to be done offline. The data of the sensors and the external outside-in tracking system are captured and processed in the suite. This results in the determination of the constellation of the odometric sensor units on the platform and the associated scales.

The initial values of the minimization process turned out to be critical for the successful computation of the calibration. Thus, rough estimates of the positions of the sensors as determined with the help of the external tracking system are used as starting values, which are further refined by the optimization process.

Temporal calibration In order for the calibration package to produce correct data, it is imperative that the relative lag between the odometric sensor data and the external tracking data is calibrated and corrected. This can be accomplished by using the method described in chapter 5. The temporal calibration revealed that the lag of the odometric sensor setup relative to the A.R.T. system is extraordinary large and measures around 400 ms.

Figure 9.13 shows the impact of this large relative lag. The first two diagrams show the misalignment between the motion as perceived by the external tracking system (black) and the odometric tracking (red). In the uncorrected case the external system perceives the start of the motion at sample 300 (6 s), whereas the odometric tracking detects a similar vector only at sample 322 (6.44 s). The second two diagrams show the same data after having undergone temporal calibration. Here the agreement between the two systems has greatly improved.

This large lag can most be attributed to the poor performance of the serial-to-ethernet conversion for realtime tracking applications.

Tracking The tracking operation of the system is fully integrated into the Ubitrack tracking framework. For this purpose the complete task was split up into three separate parts which are realized as individual components. First the communication with the hardware is implemented as a pure abstraction layer which directly inputs the observed sensor data into the dataflow. At the same time the hardware timestamps are converted into system timestamps of the framework.

Second a motion estimation component deduces the corresponding relative motion of the platform for every tuple of sensor data. The estimation is based on the motion equation as outlined above. This component also has access to the calibration data as determined in calibration step.

Third these relative 3DoF motions are integrated into a 3DoF pose by an accumulation component. Since the odometric tracking system provides solely relative data, the

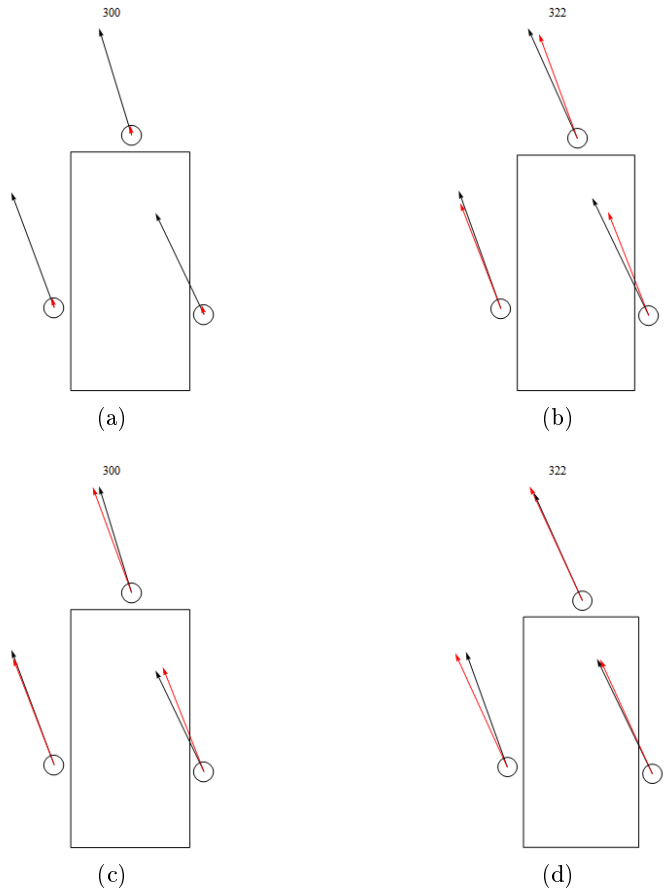


Figure 9.13: Impact of temporal calibration: The misalignment between motion as perceived by external (black) and odometric tracking (red).

accumulator can only operate after suitable initialization with an absolute pose. This can for example be determined by vision based methods (for example in the forklift scenario described above) or other external tracking systems. To reduce accumulation error, reinitialization with absolute poses can be used periodically, depending on their availability.

For the real-time tracking, the motion equations are solved using Levenberg-Marquardt optimization. Since the tracking is relative and the motion is perceived as increments, the zero 3DoF-pose is a reasonable choice as initial value for the optimizer.

9.6 Performance evaluation

In oder to assess the quality of tracking results obtained by the odometric tracking device, different evaluation scenarios were used.

9.6.1 Integral system performance

An initial performance evaluation of the complete system using different surfaces and platform motions showed a translation and rotation error of about 10 % of the way travelled respectively of the angle turned, as compared to an external ground-truth tracking system. The sensor platform was set up on a suitable surface and similar to the calibration phase, an optical outside-in marker was fixed to the platform. In this evaluation the A.R.T. system was again used as the reference system. The initial pose of the platform was obtained from the reference system and the error was computed as the difference between the two tracked poses.

The main contribution to this error is expected to be related to the scaling factors of the individual sensors, which is also dependent on the floor properties and texture.

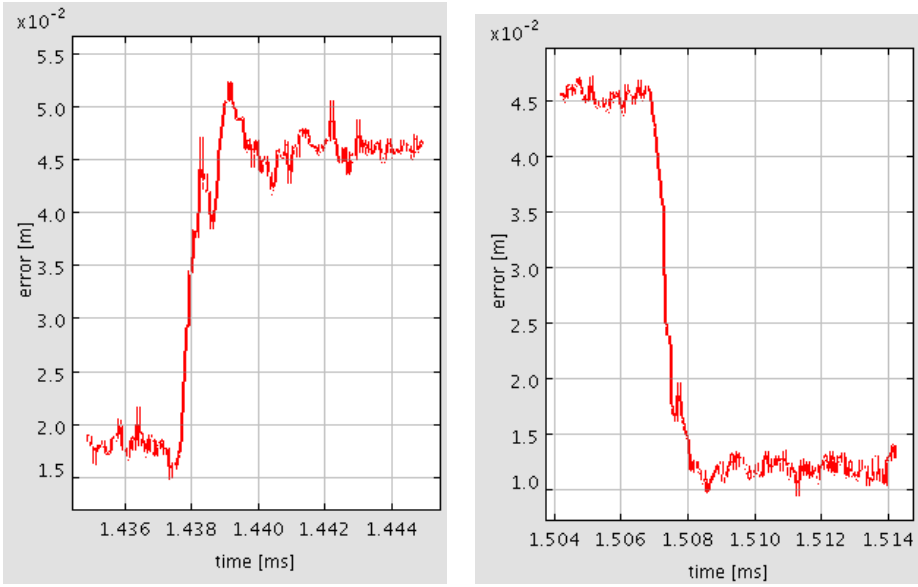
This is illustrated in the figure 9.14, which shows the error when moving 30 cm forwards and 30 cm back into the original position. As can be seen, in this case the error increases by about 10 % of the way traveled. Furthermore it can be seen that the error actually decreases when moving backwards into the starting position. This indicates a difference between the calibrated scale and the scale at that moment of the evaluation. On the other hand, this also indicates that the overall number of ticks for both the forward and backward motion is similar.

It should also be noted that for this evaluation, the calibration of the relative lag was not yet available. Considering the relative large temporal offset between the sensor pair used both for the calibration and the later evaluation, this has two direct consequences, both of which lead to unnecessarily bad apparent performance.

First, the calibration of the odometric constellation was performed using suboptimal sensor data correspondences, leading to inconsistent estimates. Second, also the computation of the tracking error used similarly shifted sensor correspondences, thus generally overestimating the tracking error. This can also be seen in figure 9.14. At the end of the forward motion, after the platform actually stopped, the error ceased increasing shortly afterwards and actually decreased before settling on a level. This overshoot is caused by the misalignment between the reference system and the platform, which becomes irrelevant when the platform rests.

A similar evaluation which, amongst other things, takes care of this aspect is planned for future work.

Further practical deployment at the ISMAR 2008 demonstration and during the ISMAR 2009 tracking competition suggests that the performance in real-world scenarios is better than in the evaluation and furthermore demonstrated the actual feasibility



(a) Forward (b) backward

Figure 9.14: Error vs. time for platform movement

of this approach. These deployments utilized the temporal lag compensation for both calibration and operation.

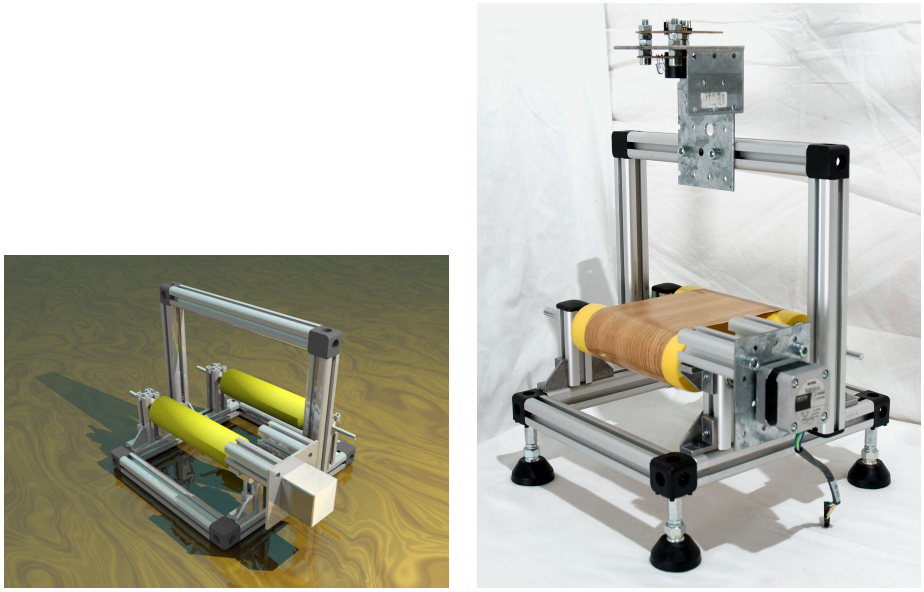
9.6.2 Single-Sensor Evaluation results using controlled actuators

To gain a better understanding of the limits of this approach and to explore further sources of tracking errors, additional systematic evaluations were performed. These evaluations focused on single sensor performance, especially texture dependence, consistency of measurements and scale parameters.

The evaluations were performed using two different actuator systems.

Conveyor Belt The first system is a conveyor belt-like construction. The overall setup can be seen in figure 9.15, both as designed and as finally realized including mounted device under test.

The evaluation setup consists of a rigid aluminum framework, designed to host two rollers. Each of the rollers is sustained in an appropriate height and mounted on low-friction ball bearings. As material for the rollers hard rubber foam was chosen, to make the rollers soft enough to make good contact to the belt, while still being able to transfer enough tension. The rollers are positioned facing each other at a distance of about 13 cm. The actual distance is adjustable for different sizes of belts.



(a) (b)

Figure 9.15: Treadmill construction

Between the rollers, different belt materials can be inserted, forming the conveyor belt. Finally the shaft of one of the rollers is fixed to a motor, which can be controlled by the computer. The motor is a stepper motor, which guarantees a known one-to-one correspondence between computer commands and roller rotations. Stepper motors, as opposed to DC motors for example have no minimum operating speed and can maintain full torque at very slow speeds.

The motor is controlled by an *Atmel ATmega 168* microcontroller and according power drivers (*L298/L297*) which are needed to drive the motor coils. The microcontroller communicates with the evaluation computer which also collects the sensor measurements. To ensure correct timing of the motor steps, only the desired speed is communicated to the microcontroller, which in turn uses hardware timers to schedule the steps at the desired frequency.

To host the sensor under test, an additional beam was attached to the frame. The height at which the beam is mounted is designed so that the usual working distance of the odometric sensors is achieved. This height is measured from the sensor mount to the belt between the rollers.

The idea of this setup is to measure longer runs of the sensor with uniform, yet known movements. By endlessly moving the simulated "floor" instead of the sensor, much less actual space is required. The focus of this evaluation is to determine the consistency of the

sensor measurements and to systematically explore the influence of different parameters on an otherwise repeatable setup.

The following parameter were investigated:

- Texture — Different materials with selected different textures were used as belt material. This gives insight into the range of the relationship between scale and surface texture. The different textures used are shown in figure 9.16. Care was taken to include a wide range of textures and to include both natural floor textures as well as more challenging textures.

- Speed — With the control of the stepper motor, different rotation speeds and thus different belt speeds can be produced. Ideally the number of ticks per second as measured by the sensor should be proportional to the motor step frequency. This would indicate that the scale of the sensor is consistent across different movement speeds. The speeds used were 50 Hz, 100 Hz, 250 Hz, 500 Hz and 1000 Hz, which subjectively represent belt speeds from relatively slow to very fast.

- Distance — Due to the nature of the projection of the telephoto lens, the scale also exhibits dependency on the distance between the sensor and the surface. By varying the height of the beam, different distances were simulated. Three different distances were used: 170 mm, 192 mm and 220 mm.

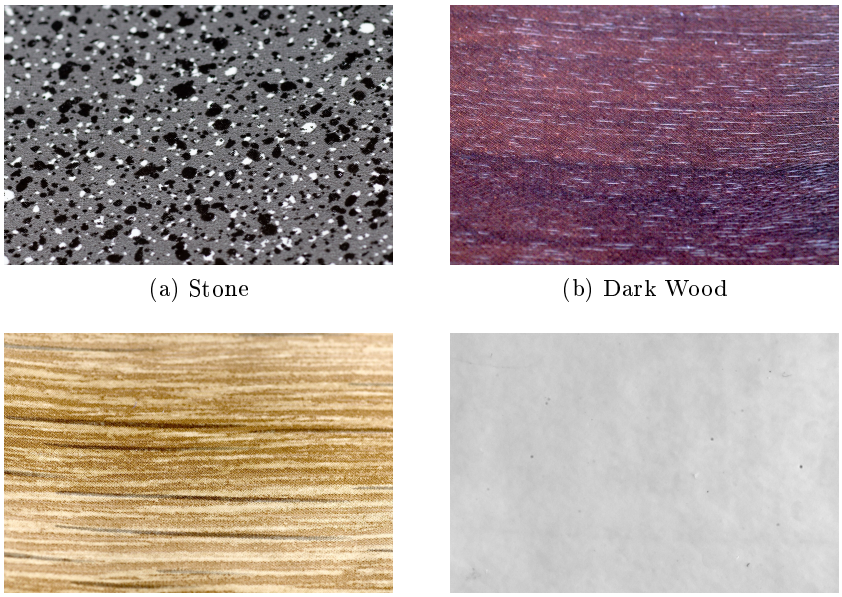
- Orientation — As was previously described, the scale of each sensor can either be assumed to be equal in both main axes of the sensor image plane (isotropic scale) or to be different for both axes (non-isotropic). Apart from surface features this could also be product or manufacturing specific. By mounting one sensor in two different orientations, the difference between these axes can be evaluated.

- Sensor — The scale can be assumed to be independent between different sensors (non-uniform) or can be equal for all sensors in the constellation (uniform). By keeping the setup undisturbed, while exchanging the sensor, the manufacturing specific differences between individual sensors can be examined.

Note that for this series of experiments, the absolute accuracy of the sensor data was not the focus.

Different combinations of these parameters were investigated in four different experiments. In each experiment, each combination was recorded for 10 seconds and each trial was repeated 10 times. In the following both the mean and the standard deviation will be presented for each combination. While the mean serves the purpose of direct comparison, the standard deviation shows the robustness or stability of the individual measurement.

It should also be noted that the evaluation frame still suffered from structural defects, that could be remedied in later versions. For example the rollers lacked lateral guidance



(a) Stone (b) Dark Wood

(c) Light Wood (d) Plain White

Figure 9.16: Different textures used for single sensor evaluation

of the belt, which thus suffered from sideways movements. The tension of the belt had to be chosen as a compromise between undesired fluttering of the belt (if the tension was too low) or missed steps due to slippage (if the tension was too high). A better roller material that is matched to the materials of the belt should help to remedy this problem. While some of the uncertainty in the sensor data may be due to these evaluation imperfections, interesting effects and problems of the sensor can nevertheless be observed.

The first combination explored the robustness of the speed estimation, as well as the relationship between measured speed and texture. Table 9.1 shows the data measured. As can be seen the absolute measured values are comparable for the different textures, with the exception of the plain white surface. Table 9.2 illustrates this point again by showing the overall spread of the measured values, after exclusion of the white surface. The spread again is shown as the standard deviation of the individual combinations, which is also given as a percentage of the mean. The relationship of the results for the white surface is shown as the percentage of the mean of this particular case as compared to the mean of the rest.

From these two tables various conclusions can be drawn. First it is evident that a dependency of the estimated speed on the texture used exists. Nevertheless, the impact of this dependency is less than the author would have anticipated. The measurements suggest that a worst case error of 3 % of the estimated speed should be feasible independent

Texture	50 Hz	100 Hz	250 Hz	500 Hz	1000 Hz
Lt. Wood	133.8 (5.9)	278.3 (13.0)	717.6 (17.0)	1337.0 (40.2)	2744.7 (17.5)
Dk. Wood	143.6 (2.4)	287.4 (9.3)	737.1 (25.4)	1379.2 (12.9)	2718.7 (104)
Stone	140.1 (3.8)	293.0 (4.5)	736.6 (8.7)	1388.5 (17.2)	2755.2 (10.1)
Plain White	74.0 (5.4)	220.9 (9.0)	583.9 (19.5)	1201.0 (23.0)	2472.8 (39.5)

Table 9.1: Texture / Speed combinations. All values are in ticks/sec. Values in parenthesis denote standard deviation.

Speed	Mean	Std. Deviation	Plain White
50 Hz	139.2	4.0 (2.9 %)	53 %
100 Hz	286.2	6.1 (2.1 %)	77 %
250 Hz	730.4	9.1 (1.2 %)	80 %
500 Hz	1368.2	22.2 (1.6 %)	88 %
1000 Hz	2739.5	15.3 (0.6 %)	90 %

Table 9.2: Combined Texture / Speed. Values in ticks/sec.

of textures encountered throughout an arbitrary scenario.

This observation is of course with exclusion of the plain white surface. This surface was chosen as a test which exhibits as few surface features as possible. As can be seen, the performance of white is much worse than that of the rest. Thus, while the dependency on the concrete form of the texture is less pronounced, the dependency on any observable texture is apparent.

Another very interesting aspect is that the robustness of the measurements, including the white surface, increases with increasing speed. The spread of the measurements is generally larger at slow speeds and is more concentrated at higher speeds. Even the white surface produced 90 % of the ticks compared to the other textures, considering the speed of 1000 Hz.

The relationship between the different speeds measured as compared to the real speed is explored in table 9.3. It shows the ratio between the different speed estimates compared to the ratio between the different clock rates of the motor. For the sensor data the speed measured at 50 Hz is taken as the reference set at 1.0.

The quotient estimated for the other motor speeds are represented as multiples of the reference speed. This can be compared to the ratios between the different clock rates and 50 Hz.

Again, white is excluded from the mean and shown in a separate column. This experiment suggests that indeed the measured speeds exhibit proportional increase with increasing speed.

Speed	Reference	Mean	White
50 Hz	1.0	1.0	1.0
100 Hz	2.0	2.1	3.0
250 Hz	5.0	5.2	7.9
500 Hz	10.0	9.8	16.3
1000 Hz	20.0	19.7	33.4

Table 9.3: Relationship between speed estimates

The next experiment investigated two different sensors while keeping the speed and the belt texture fixed. The result showed that there was no discernible difference of the data produced by the two different sensors. This implies that indeed uniform scale can be assumed for the sensors of a constellation, under the assumption (as will be shown later) that the distance of all sensors is equal.

Due to the exchange of the sensors, both sensors had to be re-focused prior to the experiments. An interesting observation in this context is that the re-focusing did not influence the performance of the sensor and that indeed the methods to adjust the focus of the sensor units are sufficient.

In a similar experiment, a single sensor unit is either mounted normally or with an additional 90° rotation, effectively exchanging the two image axes of the sensor. Only a single texture and a single speed were used in this experiment. The result was again that there is no discernible difference between the two image axes of a sensor. This suggests that isotropic scale can be assumed.

In the final experiment with the conveyor belt, different operating distances of the sensor were examined. The height of the beam was changed, such that the distance between the sensor and the belt surface was 170 mm, 192 mm or 220 mm respectively. Note that 192 mm represents the normal distance used for the other experiments. Again a single surface texture (light wood) and a single speed (100 Hz) were used. The sensor was refocused between the different runs to account for the changes in distance. Table 9.4 shows the results.

As can be seen the estimated speed as perceived by the sensor decreases with increasing distance. This is as expected, since the effective visible area increases with distance due to the projection of the telephoto lens. As single features need to travel a larger distance for a single tick in the image plane, the perceived speed decreases.

Consequently the calibration of a single sensor is not meaningful and the complete constellation after installation on the platform always needs to be calibrated. In contrast to the optimistic premise formulated earlier, it is advisable to assume non-uniform scale. As can be seen in the table a difference in distance of only 2 cm can already result in a 20 % mismatch of the estimated speed. Thus to ensure robust calibration results, the distances of the different sensors in a constellation should not be assumed to be equal,

Distance	Mean	Std. Deviation	Ratio
170 mm	355.6	7.6 (2.1 %)	122 %
192 mm	291.5	5.8 (2.0 %)	100 %
220 mm	242.0	7.8 (3.2 %)	83 %

Table 9.4: Influence of distance; Values are in ticks/sec

since the respective mounting heights may also vary. A different approach to solve this problem will be discussed later.

Linear actuator The second system used for evaluation of a single odometric sensor unit is a high precision linear actuator. An overview of the setup can be seen in figure 9.17.

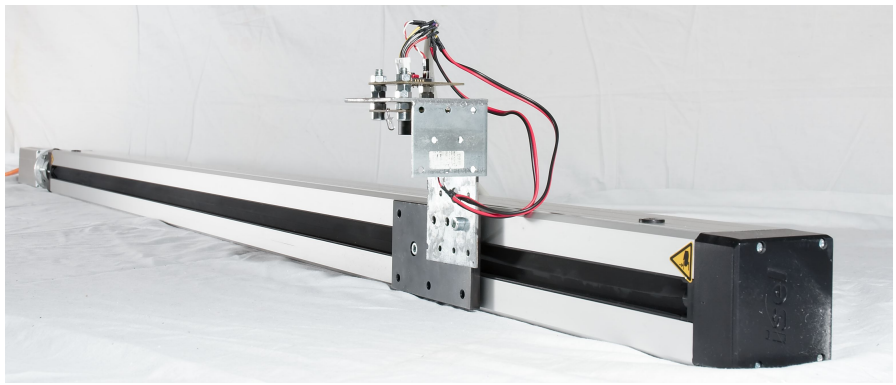


Figure 9.17: Linear actuator

The setup mainly consists of a heavy duty, industrial linear actuator with a ball screw drive and a mounted platform. The screw is connected to a stepper motor, which in turn can be controlled by the computer. One advantage of this setup is the high precision and repeat accuracy and the overall robustness of the actuator and the corresponding movement.

The screw has a pitch of 2.5 mm per rotation and the motor can be used in 1/4-step mode, which translates to 1600 steps per rotation. This results in a theoretical spatial resolution of 0.0016 mm per step. The motor can also be used in half- or full-step mode, which increases the speed while reducing the resolution.

The sensor unit under test is mounted on the platform of the linear actuator which is tilted 90°, such that the unit is held from the side and is looking downwards (as seen in figure 9.17). The actuator is then suspended in a suitable height, so that a suitable working distance for the sensor is achieved. To test different floor surfaces, various textured patches were fixed beneath the actuator.

In this evaluation only two parameters were investigated.

Texture	500 Hz	750 Hz	1000 Hz	1250 Hz
Table	401.6 (6.7)	393.2 (18.0)	406 (15.1)	425.4 (14.0)
Lt. Wood	406.4 (2.2)	402.0 (10.8)	418.8 (9.3)	432.6 (10.8)
Dk. Wood	415.8 (4.5)	429.2 (7.6)	437.0 (2.4)	451.0 (3.7)
Stone	416.2 (4.0)	420.6 (4.2)	436.2 (5.3)	449.4 (2.0)
Plain White	—	—	391.6 (9.3)	391.6 (12.7)
White Cloth	418.6 (3.9)	426.2 (7.1)	430.8 (6.4)	444.8 (4.9)

Table 9.5: Texture / Speed combinations. All values are in total ticks/100 mm. Values in parenthesis denote standard deviation.

- Speed — Different stepper frequencies and thus different spatial speeds were investigated. The speeds used were 500 Hz, 750 Hz, 1000 Hz and 1250 Hz at full-step mode of the stepper motor. Due to the known properties of the linear actuator, these can be translated to $3.13\,\mathrm{mm\,s^{-1}}$, $4.69\,\mathrm{mm\,s^{-1}}$, $6.25\,\mathrm{mm\,s^{-1}}$ and $7.81\,\mathrm{mm\,s^{-1}}$ respectively. Unfortunately, all these speeds subjectively still fall into the "slow" range of the first evaluation system.

  In principle, the linear actuator is able to achieve higher speeds, but the current implementation of the power electronics needed to drive the stepper motor imposes these limits.

- Texture — The same collection of textures as seen in figure 9.16 was used for this evaluation.

The main focus of this evaluation is to investigate if, depending on the texture, a meaningful correspondence between sensor ticks and physical movement can be established.

Each combination was recorded 10 times for a fixed distance of 100 mm. Thus each combination was evaluated over the distance of 1 m in total. In the following both the mean and the standard deviation will be presented for each combination. The values are given as total number of ticks per 100 mm. The results are shown in table 9.5.

The results for the plain white surface at speeds of 500 Hz and 750 Hz are missing, since an error in the recording was detected after the experiment. For these two runs an unfavorable lighting situation caused direct reflections of a lamp, which resulted in a complete loss of ticks. This also illustrates the importance of a kind of light guard around the sensor, which would help to eliminate interference from external light.

An interesting observation from the table is that the total number of ticks for the same distance increases with speed. As already mentioned, all the speeds used were unfortunately low, which leads to the assumption that possibly ticks are lost below a certain minimum speed.

In the next experiment this assumption was further investigated. The surface was

chosen to be light wood and the speed was set to 5 Hz at full-step mode or $0.03\,\mathrm{mm\,s^{-1}}$. A total number of 1600 steps were performed and thus the sensor was moved by exactly 10 mm in 5 minutes. Interestingly this produced no measured ticks at all, which suggests that the optical sensor itself tries to compensate very slow movements. This proves the existence of possible negative drift (the physical reality exhibits motion, whereas the sensor does not acknowledge this), a minimum measurable speed and further suggests that at low speeds indeed ticks may be lost.

The overall conclusion of this evaluation was that while the optical odometric sensors still have problems and exhibit undesired or unexplained behavior in certain circumstances, their performance can be better than determined previously. A positional error of at most 3 % of the traveled distance should be attainable. Even better results should be achievable if the tracking surface was sufficiently textured and homogeneous.

9.7 Computer Vision Based Optical Odometric Tracking

To explore the principal limits of the optical odometric tracking approach, a second prototype implementing a different approach was realized.

Similar to the original concept, the main idea is to achieve a best possible pose update using only images of relative floor motion. In contrast to the first prototype, actual practicability or cost efficiency of the system are not taken into consideration.

In the following some of the main components which are used in this design are highlighted.

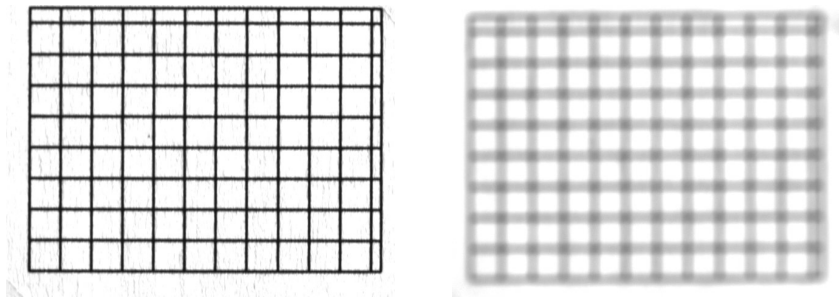
9.7.1 Design parameter

The integrated optical flow sensor is replaced by a high resolution, industrial computer vision camera. The field of view of this camera has been chosen to be comparatively small (i.e. no super-wide-angle or fisheye lens) and the operating distance to the floor of about 20 cm was maintained. One of the main features of the integrated optical flow sensor is its high frame rate (up to 3000 frames per second), which is necessary under these circumstances. Otherwise the time between frames will be too large, which would prevent correspondences from being established. Similarly the extraction of meaningful motion would become infeasible due to large motion blur.

Thus to accommodate at least moderate speeds for this prototype, a reasonably fast camera had to be employed. Currently the camera is a *PointGrey Dragonfly Express*, which achieves 200 frames per second at VGA (640×480) resolution.

Furthermore the original system exhibited the undesirable effect that the scale of the sensor was very sensitive to changes in the height of the sensor. To circumvent this effect, a special kind of lens is used in this setup. A *telecentric* lens projects parallel light rays and thus effectively has infinite focal length. The effect of this characteristic is that the size of an object in the image is independent of the distance to the object

and that instead of the scale only the sharpness of the object's projection varies with distance (figure 9.18 illustrates this property). This makes such lenses especially useful for photogrammetric purposes.



(a) (b)

Figure 9.18: Two images of a calibration grid taken with the telecentric lens; camera distance between left and right image is about 50 mm

Using such a special lens the scale of the image and thus of the measured ticks become well defined and largely independent of external factors. The lens used in this prototype is an *Edmund Optics silver-line telecentric lens* with 0.2x magnification. The magnification was chosen in the lower range available for telecentric lenses, in order to achieve a larger visible area. This is necessary to allow for reasonable speeds at the reduced frame-rate of the industrial camera.

Based on the pixel size of 7.4 µm of the Dragonfly Express sensor (*Kodak KAI-0340DM/C*), a pixel on the image corresponds to 37 µm on the surface, resulting in a total visible area of about 23.7 mm × 17.8 mm at VGA resolution. Note that pixels in this configuration still are smaller than those of the original prototype (about 2 mm × 2 mm at 15 × 15 pixels, resulting in a pixel size of 133 µm). The camera setup can be seen in figure 9.19.

Due to the high frame rate of the camera, this field of view suffices to achieve trackable speeds of at least $1\,\mathrm{m\,s^{-1}}$, which is sufficient for normal human movements. This assumes that in each frame at least 1/2 of the previous frame is visible to allow proper motion estimation. To further reduce the risk of motion blur, the shutter of the camera has to be set to an adequately short interval, and sufficient illumination has to be provided.

Finally the optical flow algorithm, which was previously integrated into the hardware of the sensor, has to be explicitly performed on the camera images. The standard method, as it is implemented by the integrated optical flow sensors, is to calculate the camera motion from one frame to the next by maximizing the cross-correlation between the two frames. This is similar to the time-delay estimation as described earlier, although both the input and the estimated parameters are two-dimensional. Note that in this case

Figure 9.19: Industrial camera with telecentric lens

the two dimensions can be treated independently and no reduction to one dimension is necessary.

The 2D cross-correlation $c(\Delta x, \Delta y)$ for a shift $\Delta x, \Delta y$ between the two images $f(x, y)$ and $t(x, y)$ can be defined as

$$c(\Delta x, \Delta y) \quad = \quad \sum_{x,y} f(x, y) t(x - \Delta x, y - \Delta y).$$

Maximizing this term over the possible shifts Δx and Δy yields the desired motion estimate.

An advantage of explicitly computing the motion estimation on the captured images is that this allows to experiment with different variations of correlation, optical flow or even feature tracking algorithms. In the first evaluation three different variations of the image correlation method were investigated.

Due to the high frame rate and since the different motion estimation algorithms can be computationally expensive if implemented in software, only raw images were captured online and the application of the tracking algorithms was only performed offline. This further simplifies the comparison of different estimation methods, since precisely the same data can be used with different methods.

Because of the high cost of this setup, only a single sensor was realized and evaluation was limited to single sensor characteristics, similar to evaluation in section 9.6.2.

9.7.2 Evaluation

This system has been evaluated using the precision linear actuator in a comparable setup to the single-sensor evaluation of the original design. The focus of this evaluation was to determine the limits of the attainable 2D motion accuracy for simple linear movements.

Motion estimation algorithms In the evaluation three different variations of the cross-correlation method were examined. The first variation directly copies the behavior of the correlation method as it is realized in the hardware optical flow sensor. This variation only computes the cross-correlation between two consecutive frames for integer shifts in both x and y direction. The motion estimate is then extracted as the integer shift that maximizes the cross-correlation. Note that this only can result in integer motion estimates.

The second variation tries to improve on this by thresholding the matrix of cross-correlations for all possible x, y shifts and estimating the motion as the barycenter of this thresholded matrix. In general, this produces non-integer results, especially if a certain area of shifts produce similar correlation measures.

As a third variation, the movement of the camera was artificially increased by deliberately dropping frames for computation of the cross-correlation. The estimates for a number (five in the concrete case) of different "temporal scales" are combined into an improved result.

More formally, let $m(t, t+1)$ denote the motion estimated between frame t and $t+1$, as computed by the previous correlation method. The motion $M(t)$ at frame t is thus computed as

$$M(t) = \frac{1}{5} \sum_{i=1,\ldots,5} \frac{m(t-i,t)}{i}.$$

Thus to compute the relative motion resulting in frame t, the history from the previous five frames is appropriately scaled and the mean over these measurements is calculated.

The idea is to reduce accumulated errors due to slow or not perceivable movements and is motivated by the observation that the original prototype performs better at higher speeds. Ideally the frame-drop rate should itself depend on the way traveled rather than use fixed decimation rates. Nevertheless the evaluation is valid to determine whether the motion estimation can be improved using this approach.

Evaluation setup The setup of the evaluation itself was very similar to the setup described in 9.6.2. The camera system was mounted on the cart of the precision linear actuator which was moved by different distances and speeds. Overall four different speed settings were used: 250, 500, 750 and 1000 steps per second at full-step mode. Furthermore three different surface textures were used: light wooden, dark wooden and

Test	Correlation method		
	Integer	Barycenter	"Multiscale"
Lt. wood; 250 Hz	13.2 %	0.67 %	0.45 %
Lt. wood; 500 Hz	6.4 %	2.8 %	0.18 %
Lt. wood; 750 Hz	0.9 %	0.35 %	0.31 %
Lt. wood; 1000 Hz	2.6 %	0.62 %	0.29 %
Dk. wood; 250 Hz	19.3 %	2.47 %	0.31 %
Dk. wood; 750 Hz	2.59 %	0.75 %	0.34 %
Plain white; 250 Hz	11.9 %	23.57 %	1.91 %
Plain white; 750 Hz	1.5 %	1.7 %	0.19 %
Lt. wood; 250 Hz, 1 m	25.6 %	1.84 %	0.32 %
Lt. wood; 750 Hz, 1 m	0.6 %	2.45 %	0.36 %

Table 9.6: Texture / Speed combinations. All values denote total distance estimation errors in percent.

plain white. In each test the cart was moved either 100 mm or 1000 mm and each test was repeated multiple times.

Results and interpretation The results of this evaluation are summarized in table 9.6. Interpretation of this data leads to several interesting observations.

The first remarkable aspect is that the first variation roughly reproduces the performance of the hardware sensor setup. The ranges of the overall error is 2 % to 3 %, which fits to the prediction mentioned above. Large deviations from this overall behavior are visible especially in cases with difficult textures or slow movement speeds. Even though both the visible area and the resolution are improved in the computer vision prototype, no real accuracy improvements are obtained in this manner. Furthermore the aforementioned dependency between the speed of the movement and the accuracy is still evident. As was the case with the hardware prototype, the measurement error decreases with increased speed.

A further effect, which cannot be seen from the data, is that the estimated direction of the movement "snaps" to discrete directions. In the experiment the camera setup was mounted on the actuator cart with a rotational misalignment of about 0.3°, whereas the direction of the movement was measured to be either 0° or 14°. Again, this "snapping" was also an observed effect with the hardware prototype.

In summary, this suggests that part of the inaccuracies experienced with the hardware sensor are indeed caused by the simple, internal processing of the hardware sensor. Increasing the resolution and view of the system only improves the accuracy of the system by a limited amount.

The second variation improves on these results and exhibits error rates almost univer-

sally below 1 %. This represents a substantial improvement over the first variation of the correlation algorithm and already achieves better results than were conjectured to be eventually attainable for the hardware sensor. The main reason for this improvement is that due to the fractional results of the frame-to-frame motion estimation the errors which accumulated in the first variation because of integer rounding are greatly reduced. While both the sensitivities to the movement speed and the surface texture are reduced, there are still problematic cases such as very slow movements or the plain white surface.

It should be mentioned that some trials were encountered which do not fit to this explanation so far and would require closer inspection on the reason of the concrete failure. An example of this phenomenon are some of the trials of "light wood at 500 Hz" which behave worse than the same experiment at 250 steps per second. This runs contrary to the general trend that slower motion produces worse results.

Another noteworthy aspect of this variation is that it exhibits almost the same computational cost as the first variation.

The third variation offers some final improvements over the previous results. The error is consistently below 0.5 % (which corresponds to a tracking error of $5\,\mathrm{mm\,m^{-1}}$). This is a moderate improvement over the well-behaved cases of the second variation, but a tremendous improvement compared to the pathological cases. The pronounced sensitivity on the movement speed is completely eliminated using this correlation variation, as was intended by increasing the history used for motion estimation. The only case where the estimation error is noticeably higher compared to the other cases, is the "plain white" texture at very slow speed. This combination of almost no texture and vanishingly small difference between consecutive frames is arguably the hardest scenario. While the performance is worse than in the other cases, it is still better than the attainable performance projected for the hardware prototype in the general case.

It should also be noted that errors below 0.3 % are below the designed accuracy of the experimental setup, and thus do not necessarily originate from the motion estimation. Other mechanical inaccuracies might have influenced the recorded data at these error levels. This includes vibration of the camera beam, vibrationally induced movement of the actuator relative to the texture or potentially missed steps of the stepper-motor. To rigorously measure errors below this level, a more precise setup would be needed which eliminates further error sources.

As a final note the computational cost of the third variation is obviously higher than the cost of the other two variations. For each frame not one but multiple (five in the concrete experiment) correlations to previous frames need to be computed. Thus the computing time is similarly multiplied by the same factor.

In summary, this experiment showed that errors of the same magnitude and characteristics as the hardware prototype can be reproduced using much greater hardware expenditure. At the same time rather simple modifications of the motion estimation algorithms drastically improve the accuracy results of the single sensor evaluation. Thus

apart from the sensing hardware also the sensing and interpretation method is equally important to produce a precise system for use in Augmented Reality applications.

9.8 Future Work

In order to address the various drawbacks of the current system and to realize the insights gained from the computer vision odometer an advanced prototype will be investigated in the future.

To improve operating performance on very smooth, glossy or virtually textureless surfaces, different lighting methods should be examined. A method derived from optical computer laser-mice is to illuminate the visible area of the optical sensor with unfocused, coherent light as opposed to normal white light. This produces the so called "Speckle" effect, which helps to amplify microscopic surface roughness of the floor, which in turn can be used as features for tracking. Two different lighting components have been designed that either use high-power LEDs or unfocused LASER modules for illumination.

The main component of the optical odometric sensor module is the optical flow sensor and the next prototype should investigate upgrading the sensor which is currently being used. For example, an upgrade of the *ADNS5020* to the more advanced *ADNS2620* promises improvements of operational parameters and robustness. The *ADNS2620* features a larger CCD array (19×19 pixels), a higher maximum frame rate and supports programming of fixed frame rates. Furthermore, the number of required external components for this circuit is further reduced.

The timestamp generation on the microcontroller also proved to be unstable, in the sense that the interval between two measurements was not constant but rather oscillated. This can be seen in figure 9.20. This is largely a consequence of the timer layout in the firmware, which could be restructured in order to improve this behavior. Another solution would be the inclusion of a dedicated external clock circuit.

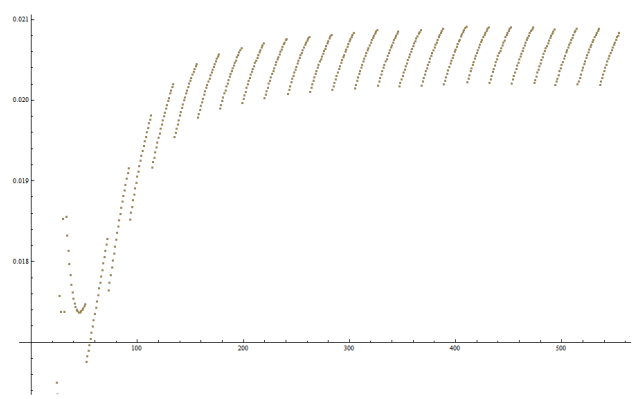


Figure 9.20: Time difference between consecutive odometer samples

Another component that had a negative influence on the performance of the timestamps is the *X-Port* communications module. While it provides very easy and comfortable access to ethernet communications for any microcontroller project, the timeliness of the packet transmission is compromised. Thus for the next prototype either UDP ethernet communications using for example an *ENC28J60* ethernet controller should be directly implemented or alternatively the less comfortable, but more robust method of USB low-speed communication or even plain RS-232 serial protocol could be used.

Further beneficial improvements to the mechanical construction of the sensor units have also been identified. These include using spacer threads between the sensor module layers for more easy assembly and disassembly of the modules and using standard PCB lens mounts as opposed to the currently used threaded hole in the iron base plate for mounting the telephoto lens.

Another promising idea that should be examined with the next prototype is the inclusion of inertial acceleration sensors. The interesting part is that the drift characteristics of these two types of sensors complement each other. On the one hand, the odometric sensor drifts in the sense that very slow movements are not detected and that the platform is estimated as more stationary than it really is. On the other hand, an accelerometer drifts in the sense that it reports small accelerations even when no movement occurs in reality. Thus the idea is to enhance each optical odometric sensor module by an additional 2-axis accelerometer. For a 3 sensor constellation a total number of 6 accelerometer channels would have to be captured. To reduce the signal processing load of the central microcontroller and to guarantee synchronicity, a dedicated analog-digital converter should be used for each accelerometer channel. A design consisting of a *LIS3L02AS4* 3-Axis accelerometer and a *AD7866* Dual 12bit-ADC together with sufficient signal conditioning per sensor unit has already been finished.

This should lead to interesting sensor fusion questions. For example a direct low-level fusion of the sensor data could either reduce accelerometer noise or increase the accuracy of the odometric tracking. More application specific fusion could help to adapt slow odometer movements to the number of expected skipped ticks or could help in automatic scale recalibration in case of a surface texture change.

To overcome the limits of the correlation-based motion estimation as it is implemented in hardware, different approaches are necessary. Since the optical flow sensors must be treated as black-box systems, no influence on the actual computation inside the integrated circuit is possible. However, one method to implement more advanced estimation methods is to increase the number of optical flow sensors per sensor unit. So far each sensor unit is designed around a single optical flow sensor. By placing a group of more than one sensor in close proximity of each other at each unit, different measurements can be taken simultaneously at each mounting point. For example by using a different lens for each sensor, measurements at different scales can be obtained simultaneously and a multi-scale correlation can be computed. The results should be similar to the

third variation of the correlation computation as evaluated above, albeit with integer results. Another possibility is to mount the sensors with slight rotational misalignments to each other. Doing so, movements with might produce rounding or truncation-errors in one sensor might produce discernible results in another sensor. Evaluation of single sensor units using such enhanced configurations will be performed using the evaluation framework presented above.

10 Conclusion

In this final chapter the results of the previous chapters will shortly be reviewed, the individual contributions of the author will be pointed out explicitly and future work will be discussed.

10.1 Summary

As stated earlier, the goal of this work was the advancement of the notion of Ubiquitous Augmented Reality and one of the most fundamental technologies required for its realization: ubiquitous tracking.

With Ubiquitous Augmented Reality, the AR metaphor has the potential to become a natural interface to ubiquitous information and computing capability that are constantly available in our environments. By extending both its reach and scope, augmented reality is transformed from a technology confined to laboratory or industrial setups to a common tool for interaction with diverse social realities.

To fully adapt to ubiquitous AR scenarios, the special characteristics of these environments need to be taken into consideration. The notion of Parasitic Tracking aims to cope with the fact that in unknown and unmanageable environments, the availability of dedicated tracking infrastructure of any kind cannot be assumed. By utilizing non-tracking related ubiquitous infrastructure in non-standard but unobtrusive ways, location or pose information can be obtained and Data from different sensing approaches can be further fused to provide more robust estimates.

As a prerequisite to implement Parasitic Tracking approaches, a suitable framework that is capable of dealing with ubiquitous tracking scenarios was required. To this end the "Ubitrack" library has been developed. It especially focuses on abstract formulation of both the tracking setup and the capabilities of tracking algorithms. The system furthermore supports the distributed computation of tracking tasks and the centralized coordination of these computations including dynamic reconfiguration to adapt to changing environments.

The dynamic fusion of sensor systems further requires the knowledge of the exact temporal relationship of the different sensors involved. To enable fusion of unknown or off-the-shelve sensors, a general method to determine the relative lag between arbitrary spatial sensors was developed and integrated into the Ubitrack framework.

Based on these concepts, three different sensing technologies, which fit into the scope of Parasitic Tracking were evaluated. Since radio localization is one of the most suitable

approaches, basic radio terminology and characteristics were reviewed, including the concept of software defined radio and its application to radio localization.

The approach examined was the localization based on IEEE802.11 Wi-Fi signals. By using a software defined radio receiver, signal strength indicators with increased physical relevance could be obtained which resulted in more stable range estimations to the various beacons. Assuming fixed anchor beacons at known locations, this was extended into a range-based localization approach.

Motivated by the increasing use of passive, long-range RFID tags in logistics and product life-cycle management, two localization approaches based on this kind of infrastructure were investigated. The first method estimated the user's position by fitting the reception pattern of a directional antenna into the currently observed tag cloud, whereas the second approach determined the range to each observed tag individually and derived the corresponding position estimate by determining the intersection. Scenarios and localization experiments using both methods were presented.

To finally bridge inevitable gaps in ubiquitous tracking environments, a local relative sensing solution was investigated. By combining multiple, low-cost optical flow sensors and observing the movement of the floor relative to the user, a robust position estimate on the floor could be derived. A prototype system and a high-end reference system were implemented and evaluated using various references.

10.2 Contribution

The following summary will shortly review the individual results and contributions described in the previous sections.

Ubiquitous Augmented Reality To define the context of this document a discussion on the various, common definitions of Ubiquitous Augmented Reality was presented. A new perspective which sees Ubiquitous Augmented Reality in the context of Social Constructionism was furthermore discussed.

This interpretation of UAR was developed by the author based on various experiences and discussions in the community.

Parasitic Tracking To overcome the additional constraints imposed by general ubiquitous environments, the notion of Parasitic Tracking was developed. It strives to provide location estimates in the absence of dedicated tracking infrastructure, by exploiting other kinds of infrastructure. In this chapter the general approach is outlined and its relation to other approaches is discussed.

The presentation in this chapter is based in parts on [Hub09], and the idea of "Parasitic Tracking" was developed and formulated by the author himself.

Ubiquitous Tracking The "Ubitrack" library with its abstraction and data flow layers was presented. For abstraction of tracking setups and tracking algorithms, the notions of the spatial relationship graph and the spatial relationship pattern were respectively introduced. The centrally coordinated peer-to-peer architecture featuring the ubitrack server was furthermore discussed and its operation described.

The presentation of the material in this chapter is based on the publications [PHBK06] and [HPK+07]. Development of the Ubitrack library is a larger, ongoing, community-based effort and thus includes the work of many participants. The collaborators include Daniel Pustka, Peter Keitler, Florian Echtler, Michael Schlegel, Benjamin Becker, Björn Schwerdtfeger and Christian Waechter. The tasks the author himself was involved in include the fundamental conception of the project, the formulation of the spatial relationship pattern notion, the design and realization of the ubitrack server and the pattern matching algorithm, as well as the implementation of various parts of the Ubitrack library.

Temporal Sensor Calibration A general method for calibration of the relative lag of two arbitrary sensors was presented. By maximizing the cross-correlation between two suitably projected signals over a range of time shifts, the temporal offset can be estimated. Various variations of the method as well as experimental results were presented.

The presentation in this chapter is partially based on the presentation [HSK09]. This research was conducted in close collaboration with Michael Schlegel. Further development of this topic beyond the material presented here was performed by Michael Schlegel and can be found in [Sch11].

Radio Localization This chapter gives an overview of basic radio and electromagnetic radiation terminology as well as the concept of software defined radio. The discussion is mostly focused on the exploitation of radio characteristics to location estimation.

The contribution of the author in this area is limited to the idea of applying software defined radio to common radio localization methods and to the identification of suitable radio parameters in a Parasitic Tracking setup.

IEEE 802.11 based Localization Due to the already near-ubiquitous permeation of Wi-Fi networks, the localization using IEEE802.11 signal is one of the most suitable approaches for Parasitic Tracking. It was shown that employing software defined radio technology can improve the stability of signal strength measurements and thus their usefulness as distance indicators. Experiments for range estimation and localization were presented.

The study presented in this chapter was performed by the author himself.

Long-range RFID based localization Passive, long-range RFID tags were identified as another suitable infrastructure for exploitation in a Parasitic Tracking manner. Two different scenarios and corresponding localization methods were discussed, focussing on different approaches to position estimation and sensing hardware. Experiments evaluating the two different scenarios were furthermore presented.

The aircraft scenario, the first localization method and its evaluation were conducted in collaboration with Benjamin Becker and Marcus Fey and were motivated by the scenario developed by EADS Innovation Works for the trackframe project. This part of the chapter is furthermore based on [BHK08] and [HBK10].

The second scenario and localization method were derived and implemented by the author without additional collaboration.

Optical odometric tracking As a final sensing method, optical odometric tracking was presented. In order to bridge gaps in ubiquitous tracking setups a hardware prototype using low-cost optical flow sensors was developed. Design and implementation considerations as well as various evaluations were described and the prototype was compared to a high-end optical odometric system using computer vision components.

The construction, implementation and initial evaluation of the hardware prototype were conducted in close collaboration with Michael Schelgel. Further evaluations using the mechanical actuators as well as the comparison to the computer vision reference system were performed by the author himself. The setup of the mobile tracking platform and the integration of the odometric tracking system for the ISMAR tracking competition (as described in [WHK+10]) was performed by Christian Waechter and Peter Keitler with input from the author.

10.3 Future Work

While the author hopes to have made at least a small impact on moving ubiquitous augmented reality forward, a number of new questions, problems and further directions of inquiry have emerged from this work. While some of the more immediate future work has been discussed in the individual chapters, the following list presents conceptual directions for future research.

Localization methods As was previously discussed, there is a number of different approaches to perform location estimation based on the sensor data obtained from observing different kinds of infrastructure. Each of these approaches has its individual advantages and disadvantages and they are based on diverse assumptions on the sensing process or the environment.

While the localization scheme that was used for the Wi-Fi and the RFID experiments was chosen based on the reasoning previously discussed (see section 3.3), the comparison to further localization methods would be very interesting. Various algorithms from different localization approaches should be evaluated based on the sensing technologies developed so far.

Of special interest in this regard are methods capable of directly integrating and modeling the inherent uncertainty of the sensor measurements gained from non-tracking related infrastructure. Since radio based sensing methods especially suffer from noise and uncertain or contradicting observations, the ability to deal with these defects promises to be advantageous for parasitic tracking.

The implementations and evaluation of various localization schemes will thus be an important future work in this context.

Mapping and distribution of anchor parameters Another important aspect of parasitic tracking approaches, is the assumption that the parameters, especially the location, of the infrastructural anchors are known in advance.

A possible solution to the problem of how to determine these properties is to employ a *SLAM*-like approach (see [DWB06] for an introduction). By bootstrapping the localization process using previously determined landmarks or even other localization schemes it might be possible to simultaneously map the environment and determine the user's position.

The determined anchor parameters can furthermore be distributed using a suitable communications channel, thus creating a distributed database of exploitable infrastructures and the associated anchor data. Note that this further underlines the need to cope with and detect contradicting information in the parasitic tracking process, as no participant can assume the correctness of any fact learned via this channel. In the case that the distributed information does not agree with the parameters as determined by a different user, it is furthermore expected that both parameter values will be distributed. Using a system based on social feedback on the correctness of the individually distributed parameters, it might be possible to gradually remove the incorrect or superseded facts by corrected information.

The further conception of such systems that map unknown environment and share this piece of information across social groups will be part of the future work on parasitic tracking.

Sensing technologies Apart from the evaluated sensing technologies, there is a number of further infrastructures, which would be suitable for the use in parasitic tracking setups. Further radio based technologies include for example GSM, DECT, Bluetooth infrastructure or broadcast radio stations. The search for additional ex-

ploitable infrastructures should nevertheless not be limited to radio based setups. For example the inclusion of environmental noise might produce interesting results.

It will thus be an ongoing future work to identify suitable technologies and to make them available in the Ubitrack framework.

Ubitrack server One of the main drawbacks of the ubitrack server is the still noticeable time required for automatic resolution of queries and for adaption to changes in the environment. This is mostly due to the fact that the deduction process is not directly driven by reasoning towards a certain goal, but rather by exploration of the possible graph derivations of a given spatial relationship graph. By implementation of additional graph heuristics or through a suitable deduction system, it is expected that this process can be sped up considerably. In this context, the further study of graph grammars or logic programming might also prove to be helpful.

Future work will thus focus on improving the performance of the query resolution and deduction process of the Ubitrack server.

Influence of human factors The experiments conducted for the temporal calibration have demonstrated that there is a distinct influence of human factors on the various parameters of sensor data and the quality of corresponding calibrations. More specifically, it was shown that it is impossible for a human to keep the sensor combination steady enough, so that the measured movement is too little to perform temporal calibration. While this result is positive, other human factors including repeat accuracy or characteristics of visual perception may negatively influence the accuracy of other, especially spatial calibrations.

A general study of the nature of these factors and their individual impact on various methods would be interesting.

Further discussion on the vision of Ubiquitous Augmented Reality Finally further discussion on the concrete vision of Ubiquitous Augmented Reality is required. The social aspect presented earlier is intended to be a starting point for further conversation in the community. Dedicated discussions on the foundations of Ubiquitous Augmented Reality, including elaborations on the concepts of Ubiquitous Displays and Ubiquitous Interaction seem desirable and necessary.

Bibliography

[AB94] Azuma, Ronald ; Bishop, Gary: Improving Static and Dynamic Registration in an Optical See-through HMD. In: *SIGGRAPH*, 1994, 197–204 see: 5.2

[AB95] Azuma, R. ; Bishop, G.: A Frequency-Domain Analysis of Head-Motion Prediction. In: *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* ACM, 1995, S. 401–408 see: 5.2

[ASB+04] Aron, M. ; Simon, G. ; Berger, M.O. ; Loria, I. ; Nancy, F.: Handling Uncertain Sensor Data in Vision-Based Camera Tracking. In: *Third IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2004, S. 58–67 see: 5.2, 5.4.2

[ASB07] Aron, M. ; Simon, G. ; Berger, M.O.: Use of Inertial Sensors to Support Video Tracking. In: *Computer Animation and Virtual Worlds* 18 (2007), Nr. 1, S. 57–68 see: 5.2

[ASSC02] Akyildiz, I.F. ; Su, W. ; Sankarasubramaniam, Y. ; Cayirci, E.: Wireless sensor networks: a survey. In: *Computer networks* 38 (2002), Nr. 4, S. 393–422. – ISSN 1389–1286 see: 2.2

[Ava08] Avago Technologies: *ADNS-5020-EN Optical Mouse Sensor*. Datasheet, 2008 see: 9.3

[Azu97] Azuma, Ronald T.: A survey of Augmented Reality. In: *Presence: Teleoperators and Virtual Environments* 6 (1997), Nr. 4, S. 355–385. – ISSN 1054–7460 see: 2.1

[BBK+01] Bauer, Martin ; Bruegge, Bernd ; Klinker, Gudrun ; MacWilliams, Asa ; Reicher, Thomas ; Riss, Stefan ; Sandor, Christian ; Wagner, Martin: Design of a Component-Based Augmented Reality Framework. In: *Proceedings of the International Symposium on Augmented Reality (ISAR)*, 2001 see: 4.2

[BF96a] Borenstein, J. ; Feng, L.: Gyrodometry: A new method for combining data from gyros and odometry in mobile robots. In: *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on* Bd. 1 IEEE, 1996. – ISBN 0780329880, S. 423–428 see: 9.2

[BF96b] Borenstein, J. ; Feng, L.: Measurement and correction of systematic odometry errors in mobile robots, IEEE, 1996. – ISSN 1042–296X, S. 869–880 see: 9.2

[BH81] Boucher, R. ; Hassab, J.: Analysis of Discrete Implementation of Generalized Cross Correlator. In: *Acoustics, Speech and Signal Processing, IEEE Transactions on* 29 (1981), jun., Nr. 3, S. 609 – 611 see: 5.3.7

[BHK08] Becker, B. ; Huber, M. ; Klinker, G.: Utilizing RFIDs for Location Aware Computing. In: *Proceedings of UIC 2008*, 2008 (LNCS 5061) see: 8, 10.2

[BL67] Berger, Peter L. ; Luckmann, Thomas: *The Social Construction of Reality.* Anchor Books, 1967 see: 2.3

[BM04] Bohn, Jürgen ; Mattern, Friedemann: Super-Distributed RFID Tag Infrastructures. In: Markopoulos, Panos (Hrsg.) ; Eggen, Berry (Hrsg.) ; Aarts, Emile (Hrsg.) ; Crowley, James (Hrsg.): *Proc. 2nd European Symposium on Ambient Intelligence (EUSAI 2004).* Eindhoven, The Netherlands : Springer-Verlag, November 2004 (LNCS 3295), 1–12 see: 8.2

[Boy03] Boyes, W.: *Instrumentation Reference Book.* Butterworth-Heinemann, 2003 (Chemical, Petrochemical & Process). – ISBN 9780750671231 see: 6.2.1

[BP00] Bahl, P. ; Padmanabhan, V.N.: RADAR: An in-building RF-based user location and tracking system. In: *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE* Bd. 2 IEEE, 2000. – ISBN 0780358805, S. 775–784 see: 3.3.1, 7.2

[BRT05] Blumenthal, J. ; Reichenbach, F. ; Timmermann, D.: Position estimation in ad hoc wireless sensor networks with low complexity. In: *Joint 2nd Workshop on Positioning, Navigation and Communication*, 2005, S. 41–49 see: 3.3.1

[BS08] Bleser, G. ; Stricker, D.: Advanced Tracking through Efficient Image Processing and Visual–Inertial Sensor Fusion. In: *Computers and Graphics* (2008) see: 5.2

[Bur03] Burr, Vivien: *Social Constructionism.* second. Routledge, 2003 see: 2.3

[Car81] Carter, GC: Time Delay Estimation for Passive Sonar Signal Processing. In: *IEEE Transactions on Acoustics, Speech and Signal Processing* 29 (1981), Nr. 3, S. 463–470 see: 5.3.1, 5.3.7

[CMJ04] Coelho, Enylton M. ; MacIntyre, Blair ; Julier, Simon: OSGAR: A Scenegraph with Uncertain Transformations. In: *Proc. IEEE International Symposium on Mixed and Augmented Reality (ISMAR'04)*. Washington, DC : IEEE, 2004, 6–15 see: 4.2

[Cob97] Cobb, H.S.: *GPS pseudolites: Theory, design, and applications*, Stanford University, Diss., 1997 see: 3.3

[Cou02] Coulouris, George: Review Report: The QoSDREAM Project / Laboratory for Communication Engineering, University of Cambridge. Version: 2002. `http://www-lce.eng.cam.ac.uk/qosdream/Publications/`. 2002. – Forschungsbericht see: 3.3

[CSC+06] Chen, M. ; Sohn, T. ; Chmelev, D. ; Haehnel, D. ; Hightower, J. ; Hughes, J. ; LaMarca, A. ; Potter, F. ; Smith, I. ; Varshavsky, A.: Practical metropolitan-scale positioning for gsm phones. In: *UbiComp 2006: Ubiquitous Computing* (2006), S. 225–242 see: 3.3.1

[Dan99] Daniilidis, K.: Hand-eye calibration using dual quaternions. In: *Journal of Robotics Research* 18 (1999), S. 286–298 see: 5.4.2

[Dav10] Davies, H.: Reality through the invisible interface. In: *Mixed and Augmented Reality-Arts, Media, and Humanities (ISMAR-AMH), 2010 IEEE International Symposium On* IEEE, 2010, S. 63–64 see: 2.3

[DSA01] Dey, Anind K. ; Salber, Daniel ; Abowd, Gregory D.: A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. In: *Human-Computer Interaction (HCI) Journal* 16 (2-4) (2001), 97–166. `http://www.cc.gatech.edu/fce/ctk/pubs/HCIJ16.pdf` see: 3.3

[DWB06] Durrant-Whyte, H. ; Bailey, T.: Simultaneous localization and mapping: part I. In: *Robotics & Automation Magazine, IEEE* 13 (2006), Nr. 2, S. 99–110 see: 10.3

[EBM05] Endres, Christoph ; Butz, Andreas ; MacWilliams, Asa: A survey of software infrastructures and frameworks for ubiquitous computing. In: *Mob. Inf. Syst.* 1 (2005), January, 41–80. `http://portal.acm.org/citation.cfm?id=1233803.1233806`. – ISSN 1574–017X see: 3.3

[EHP+08] Echtler, F. ; Huber, M. ; Pustka, D. ; Keitler, P. ; Klinker, G.: Splitting the Scene Graph - Using Spatial Relationship Graphs Instead of Scene Graphs in Augmented Reality. In: *Proceedings of the 3rd International Conference on Computer Graphics Theory and Applications (GRAPP)*, 2008 see: 4.2

[EM06] Evennou, F. ; Marx, F.: Advanced integration of WiFi and inertial navigation systems for indoor mobile positioning. In: *Eurasip journal on applied signal processing* 2006 (2006), S. 164–164. – ISSN 1110–8657 see: 3.3.1, 7.2

[ESK+03] *Kapitel* seventeen. In: Echtler, F. ; Sturm, F. ; Kindermann, K. ; Klinker, G. ; Stilla, J. ; Trilk, J. ; Najafi, H.: *The intelligent welding gun: Augmented reality for experimental vehicle construction.* Springer Verlgag, 2003, S. 333–360 see: 2.1

[Fin02] Finkenzeller, Klaus: *RFID-Handbuch.* third. Carl Hanser Verlag, 2002 see: 8.3, 8.5.1

[FLS63] Feynman, Richard P. ; Leighton, Robert B. ; Sands, Matthew: *The Feynman Lectures on Physics.* Pearson Addison Wesley, 1963 see: 6.1

[FMZ+10] Firooz, Mohammad H. ; Maas, Dustin ; Zhang, Junxing ; Patwari, Neal ; Kasera, Sneha K.: Channel Sounding for the Masses: Low Complexity GNU 802.11b Channel Impulse Response Estimation. In: *CoRR* abs/1007.3476 (2010) see: 7.2

[Fod02] Fodor, I.K.: A Survey of Dimension Reduction Techniques. In: *US DOE Office of Scientific and Technical Information* (2002) see: 5.3.3, 5.3.5, 8.5.1

[Fre02] Frerking, Marvin E.: *Digital Signal Processing in Communication Systems.* Kluwer Academic Publishers, 2002 see: 6.3.2

[GGV+10] Gruber, Lukas ; Gauglitz, Steffen ; Ventura, Jonathan ; Zollmann, Stefanie ; Huber, Manuel ; Schlegel, Michael ; Klinker, Gudrun ; Schmalstieg, Dieter ; Höllerer, Tobias: The City of Sights: Design, Construction and Measurement of an Augmented Reality Stage Set. In: *Proceedings of the 9th International Symposium on Mixed and Augmented Reality (ISMAR)*, 2010 see: 5

[GJ79] Garey, Michael R. ; Johnson, David S.: *Computers and Intractability; A Guide to the Theory of NP-Completeness.* W. H. Freeman & Co., 1979 see: 4.6

[Gol94] Goldsmith, Andrea: *Design and performance of high-speed communication systems over time-varying radio channels*, University of California at Berkeley, Diss., 1994 see: 6.2.2

[Gol05] Goldsmith, Andrea: *Wireless Communications.* Cambridge University Press, 2005 see: 6.2.2

[Ham08] Hamza, F.A.: The USRP under 1.5 X Magnifying Lens! In: *GNU radio project* (2008) see: 7.5

[HB01a] Hightower, J. ; Borriello, G.: Location systems for ubiquitous computing. In: *Computer* 34 (2001), Nr. 8, S. 57–66. – ISSN 0018–9162 see: 3.3

[HB01b] Hightower, Jeffrey ; Borriello, Gaetano: A Survey and Taxonomy of Location Systems for Ubiquitous Computing / University of Washington. 2001 (UW-CSE 01-08-03). – Forschungsbericht see: 3.3

[HBB02] Hightower, J. ; Brumitt, B. ; Borriello, G.: The location stack: A layered model for location in ubiquitous computing. In: *Mobile Computing Systems and Applications, 2002. Proceedings Fourth IEEE Workshop on* IEEE, 2002. – ISBN 0769516475, S. 22–28 see: 3.3

[HBF+04] Hahnel, D. ; Burgard, W. ; Fox, D. ; Fishkin, K. ; Philipose, M.: Mapping and localization with RFID technology. In: *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on* Bd. 1 IEEE, 2004. – ISBN 0780382323, S. 1015–1020 see: 3.3.1, 8.2

[HBK10] Huber, Manuel ; Becker, Benjamin ; Klinker, Gudrun: Location aware computing using RFID infrastructure. In: *Int. J. Autonomous and Adaptive Communications Systems* 3 (2010), Nr. 1, S. 23–38 see: 8, 10.2

[HHB+03] He, Tian ; Huang, Chengdu ; Blum, Brian M. ; Stankovic, John A. ; Abdelzaher, Tarek F.: Range-free localization schemes for large scale sensor networks. In: *MOBICOM*, 2003, S. 81–95 see: 3.3.1

[HKL+99] Hohl, F. ; Kubach, U. ; Leonhardi, A. ; Rothermel, K. ; Schwehm, M.: *Next Century Challenges: Nexus - An Open Global Infrastructure for Spatial-Aware Applications.* In: *Proceedings of the Fifth Annual International Conference on Mobile Computing and Networking (MobiCom '99)* Universität Stuttgart : Sonderforschungsbereich SFB 627 (Nexus: Umgebungsmodelle für mobile kontextbezogene Systeme), Seattle, WA, USA: not available, August 1999, 249-255 see: 3.3

[HLS06] Hightower, J. ; LaMarca, A. ; Smith, I.E.: Practical lessons from place lab. In: *Pervasive Computing, IEEE* 5 (2006), Nr. 3, S. 32–39. – ISSN 1536–1268 see: 3.3

[Hor87] Horn, B.K.P.: Closed Form Solutions of Absolute Orientation Using Unit Quaternions. In: *Journal of the Optical Society of America A* 4 (1987), April, Nr. 4, S. 629–642 see: 4.4.3, 5.4.2

[HPK+07] Huber, Manuel ; Pustka, Daniel ; Keitler, Peter ; Echtler, Florian ; Klinker, Gudrun: A System Architecture for Ubiquitous Tracking Environments. In: *Proceedings of the 6th International Symposium on Mixed and Augmented Reality (ISMAR)*, 2007 see: 4, 10.2

[HSK09] Huber, Manuel ; Schlegel, Michael ; Klinker, Gudrun: Temporal Calibration in Multisensor Tracking Setups. In: *8th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*. Orlando, USA, October 2009 see: 5, 10.2

[HTMF07] Hodges, Steve ; Thorne, Alan ; Mallinson, Hugo ; Floerkemeier, Christian: Assessing and Optimizing the Range of UHF RFID to Enable Real-World Pervasive Computing Applications. In: LaMarca, Anthony (Hrsg.) ; Langheinrich, Marc (Hrsg.) ; Truong, Khai N. (Hrsg.): *Proc. 5th International Conference, PERVASIVE 2007*. Eindhoven, The Netherlands : Springer-Verlag, 2007 (LNCS 4480), S. 280–297 see: 8.2

[Hub08] Huber, E.: *SmartGPS-Lokaltionsmodell für PerFlows*, Universität Stuttgart, Diplomarbeit, 2008 see: 3.3

[Hub09] Huber, Manuel: Parasitic Tracking: Enabling Ubiquitous Tracking through existing Infrastructure. In: *Proceedings of IEEE Pervasive Computing and Communications 2009, PhD Forum (PerCom'09)*, 2009 see: 3, 10.2

[HWB00] Hightower, J. ; Want, R. ; Borriello, G.: SpotON: An indoor 3D location sensing technology based on RF signal strength. In: *UW CSE 00-02-02, University of Washington, Department of Computer Science and Engineering, Seattle, WA* (2000) see: 3.3.1

[HWGL+10] Hekimian-Williams, C. ; Grant, B. ; Liu, X. ; Zhang, Z. ; Kumar, P.: Accurate localization of RFID tags using phase difference. In: *RFID, 2010 IEEE International Conference on* IEEE, 2010, S. 89–96 see: 8.2

[IEE07] 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. In: *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)* (2007), 12. http://dx.doi.org/10.1109/IEEESTD.2007.373646. – DOI 10.1109/IEEESTD.2007.373646 see: 7.3

[JLS97] Jacobs, M.C. ; Livingston, M.A. ; State, Andrei: Managing Latency in Complex Augmented Reality Systems. In: *Proceedings of the Symposium on Interactive 3D Graphics* ACM, 1997 see: 5.2

[JS93] Jacovitti, G. ; Scarano, G.: Discrete Time Techniques for Time Delay Estimation. In: *IEEE Transactions on Signal Processing* 41 (1993), Nr. 2, S. 525–533 see: 5.3.7

[JZU+10] Jadliwala, M. ; Zhong, S. ; Upadhyaya, S. ; Qiao, C. ; Hubaux, J.P.: Secure Distance-Based Localization in the Presence of Cheating Beacon Nodes. In: *IEEE Transactions on Mobile Computing* (2010), S. 810–823. – ISSN 1536–1233 see: 3.3.1

[KC76] Knapp, CH ; Carter, GC: The Generalized Correlation Method for Estimation of Time Delay. In: *IEEE Transactions on Acoustics, Speech and Signal Processing* 24 (1976), Nr. 4, S. 320–327 see: 5.3.7

[KCH03] Krumm, J. ; Cermak, G. ; Horvitz, E.: Rightspot: A novel sense of location for a smart personal object. In: *UbiComp 2003: Ubiquitous Computing* Springer, 2003, S. 36–43 see: 3.3

[KH06] Kolodziej, K.W. ; Hjelm, J.: *Local positioning systems: LBS applications and services.* CRC Press, 2006. – ISBN 0849333490 see: 3.3

[KHGE09] Kim, D.H. ; Hightower, J. ; Govindan, R. ; Estrin, D.: Discovering semantically meaningful places from pervasive RF-beacons. In: *Proceedings of the 11th international conference on Ubiquitous computing* ACM, 2009, S. 21–30 see: 3.3.1

[KKDK07] Kanburoğlu, Furkan A. ; Kilic, Ergin ; Dölen, Melik ; Koku, Buöra: A Test setup for Evaluating Long-Term Measurement Characteristics of Optical Mouse Sensors. In: *Journal of Automation, Mobile Robotics & Intelligent Systems* 1 (2007), Nr. 2 see: 9.2

[KMA+08] Kawashima, T. ; Ma, J. ; Apduhan, B.O. ; Huang, R. ; Jin, Q.: Robots in Smart Spaces-A Case Study of a u-Object Finder Prototype. In: *Ubiquitous intelligence and computing: 5th international conference, UIC 2008, Oslo, Norway, June 23-25, 2008; proceedings* Springer-Verlag New York Inc, 2008. – ISBN 3540692924, S. 61 see: 8.2

[KNU+07] Kobayashi, Kazuhiko ; Nishiwaki, Koichi ; Uchiyama, Shinji ; Yamamoto, Hiroyuki ; Kagami, Satoshi ; Kanade, Takeo: Overlay what Humanoid Robot Perceives and Thinks to the Real-world by Mixed Reality System. In: *Proceedings of the 2007 6th IEEE and ACM International*

Symposium on Mixed and Augmented Reality. Washington, DC, USA : IEEE Computer Society, 2007 (ISMAR '07). – ISBN 978–1–4244–1749–0, 1–2 see: 2.1

[KPH+10] *Kapitel* Management of Tracking for Mixed and Augmented Reality Systems. In: Keitler, P. ; Pustka, D. ; Huber, M. ; Echtler, F. ; Klinker, G.: *The Engineering of Mixed Reality Systems.* Springer Verlag, 2010 see: 4.2

[KPV10] Kushki, A. ; Plataniotis, K.N. ; Venetsanopoulos, A.N.: Intelligent Dynamic Radio Tracking in Indoor Wireless Local Area Networks. In: *IEEE transactions on mobile computing* 9 (2010), Nr. 3, S. 405–419. – ISSN 1536–1233 see: 3.3.1, 7.2

[Kus08] Kushki, A.: *A cognitive radio tracking system for indoor environments*, University of Toronto, Diss., 2008 see: 3.3.1

[LBMN09] Lieberknecht, S. ; Benhimane, S. ; Meier, P. ; Navab, N.: A Dataset and Evaluation Methodology for Template-Based Tracking Algorithms. In: *Proceedings of the 2009 8th IEEE International Symposium on Mixed and Augmented Reality* IEEE Computer Society, 2009, S. 145–151 see: 5.2, 5.4.3

[LCC+05] LaMarca, A. ; Chawathe, Y. ; Consolvo, S. ; Hightower, J. ; Smith, I. ; Scott, J. ; Sohn, T. ; Howard, J. ; Hughes, J. ; Potter, F. u. a.: Place lab: Device positioning using radio beacons in the wild. In: *Pervasive Computing* (2005), S. 116–133 see: 3.3.1

[LHSC05] LaMarca, A. ; Hightower, J. ; Smith, I. ; Consolvo, S.: Self-mapping in 802.11 location systems. In: *UbiComp 2005: Ubiquitous Computing* (2005), S. 87–104 see: 7.2

[Lip71] Lippold, Olof: Physiological Tremor. In: *Scientific American* 224 (1971), March, S. 65–73 see: 5.3.5

[LS04] Lee, S. ; Song, J.: Mobile robot localization using optical flow sensors. In: *International Journal of Control, Automation, and Systems* 2 (2004), Nr. 4, S. 485–493 see: 9.2

[LS10] Lock, Andy ; Strong, Tom: *Social Constructionism.* Cambridge University Press, 2010 see: 2.3

[LSP+10] Luimula, M. ; Sääskilahti, K. ; Partala, T. ; Pieskä, S. ; Alaspää, J.: Remote navigation of a mobile robot in an RFID-augmented environment. In: *Personal and Ubiquitous Computing* 14 (2010), Nr. 2, S. 125–136. – ISSN 1617–4909 see: 8.2

[LU95] Lackey, RI ; Upmal, D.W.: Speakeasy: The military software radio. In: *Communications Magazine, IEEE* 33 (1995), Nr. 5, S. 56–61. – ISSN 0163–6804 see: 6.4

[Mac04] MacWilliams, A.: Software development challenges for ubiquitous augmented reality. In: *GI Augmented Reality and Virtual Reality Workshop*, 2004 see: 2.3

[Mac05] MacWilliams, Asa: *A Decentralized Adaptive Architecture for Ubiquitous Augmented Reality Systems*, TU München, Diss., 2005 see: 2.3, 4.2

[Mac10] Machleidt, Stefan: *A Framework for Evaluation of Time Measurement based Tracking Approaches*, Technische Universität München, Diplomarbeit, 2010 see: 6.2.1

[MI93] Mitola III, J.: Software radios: Survey, critical evaluation and future directions. In: *Aerospace and Electronic Systems Magazine, IEEE* 8 (1993), Nr. 4, S. 25–36. – ISSN 0885–8985 see: 6.3.1

[MIMJ99] Mitola III, J. ; Maguire Jr, G.Q.: Cognitive radio: making software radios more personal. In: *Personal Communications, IEEE* 6 (1999), Nr. 4, S. 13–18. – ISSN 1070–9916 see: 6.3.1

[MK94] Milgram, P. ; Kishino, F.: A taxonomy of mixed reality visual displays. In: *IEICE Transactions on Information and Systems E series D* 77 (1994), S. 1321–1321. – ISSN 0916–8532 see: 2.1, 2.1

[MSW+03] MacWilliams, Asa ; Sandor, Christian ; Wagner, Martin ; Bauer, Martin ; Klinker, Gudrun ; Brügge, Bernd: Herding Sheep: Live System Development for Distributed Augmented Reality. In: *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, 2003 see: 2.3, 4.2

[MTUK94] Milgram, P. ; Takemura, H. ; Utsumi, A. ; Kishino, F.: Augmented reality: A class of displays on the reality-virtuality continuum. In: *Telemanipulator and Telepresence Technologies* Bd. 2351, SPIE, 1994 (Proceedings of SPIE), S. 282–292 see: 2.1

[NBP+07] Newman, J. ; Bornik, A. ; Pustka, D. ; Echtler, F. ; Huber, M. ; Schmalstieg, D. ; Klinker, G.: Tracking for distributed mixed reality environments. In: *Workshop on Trends and Issues in Tracking for Virtual Environments at the IEEE Virtual Reality Conference (VR07)*, 2007 see: 2.3, 2.2

[NIH01] Newman, J. ; Ingram, D. ; Hopper, A.: Augmented reality in a wide area sentient environment. In: *IEEE and ACM International Symposium on Augmented Reality (ISAR)* IEEE, 2001. – ISBN 0769513751, S. 77–86 see: 2.3

[NKS+08] Niwa, H. ; Kodaka, K. ; Sakamoto, Y. ; Otake, M. ; Kawaguchi, S. ; Fujii, K. ; Kanemori, Y. ; Sugano, S.: GPS-based indoor positioning system with multi-channel pseudolite. In: *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on* IEEE, 2008. – ISSN 1050–4729, S. 905–910 see: 3.3

[NLLP04] Ni, L.M. ; Liu, Y. ; Lau, Y.C. ; Patil, A.P.: LANDMARC: indoor location sensing using active RFID. In: *Wireless Networks* 10 (2004), Nr. 6, S. 701–710. – ISSN 1022–0038 see: 3.3.1, 8.2

[NWB04] Newman, Joe ; Wagner, Martin ; Bauer, Martin: Ubiquitous Tracking for Augmented Reality. In: *Proc. IEEE International Symposium on Mixed and Augmented Reality (ISMAR04)*. Arlington, VA, USA, Nov. 2004 see: 4.2, 4.7, 4.8

[NWP+03] Newman, J. ; Wagner, M. ; Pintaric, T. ; MacWilliams, A. ; Bauer, M. ; Klinker, G. ; Schmalstieg, D.: Fundamentals of ubiquitous tracking for augmented reality. In: *In Proc. of International Symposium on Mixed and Augmented Reality (ISMARâĂŹ04)*, 2003 see: 4.2, 4.7

[OBSR08] Oliveira, Filho J. ; Bunoza, A. ; Sommer, J. ; Rosenstiel, W.: Self-Localization in a Low Cost Bluetooth Environment. In: *Ubiquitous intelligence and computing: 5th international conference, UIC 2008, Oslo, Norway, June 23-25, 2008; proceedings* Springer-Verlag New York Inc, 2008. – ISBN 3540692924, S. 258 see: 3.3

[ON04] Ong, S. K. (Hrsg.) ; Nee, A. Y. C. (Hrsg.): *Virtual and Augmented Reality Applications in Manufacturing*. Springer-Verlag, 2004. – ISBN 1–85233–796–6 see: 8.2

[OVLDL05] Otsason, V. ; Varshavsky, A. ; LaMarca, A. ; De Lara, E.: Accurate GSM indoor localization. In: *UbiComp 2005: Ubiquitous Computing* (2005), S. 141–158 see: 3.3, 3.3.1

[Pat08] Patel, Shwetak N.: *Infrastructure Mediated Sensing*, Georgia Institute of Technology, Diss., 2008 see: 3.2.1

[PCB00] Priyantha, N.B. ; Chakraborty, A. ; Balakrishnan, H.: The cricket location-support system. In: *Proceedings of the 6th annual international*

conference on Mobile computing and networking ACM, 2000. – ISBN 1581131976, S. 32–43 see: 3.3.1

[PHBK06] PUSTKA, Daniel ; HUBER, Manuel ; BAUER, Martin ; KLINKER, Gudrun: Spatial Relationship Patterns: Elements of Reusable Tracking and Calibration Systems. In: *Proc. IEEE International Symposium on Mixed and Augmented Reality (ISMAR'06)*, 2006 see: 4, 10.2

[PHEK07] PUSTKA, Daniel ; HUBER, Manuel ; ECHTLER, Florian ; KEITLER, Peter: UTQL: The Ubiquitous Tracking Query Language v1.0 / Institut für Informatik, Technische Universität München. 2007 (TUM-I0718). – Forschungsbericht see: 4.8.2

[PHK08] PUSTKA, Daniel ; HUBER, Manuel ; KLINKER, Gudrun: Integrating Gyroscopes into Ubiquitous Tracking Environments. In: *Proceedings of IEEE Virtual Reality 2008*. Reno, Nevada, USA, März 2008 see: 4.2

[PK08] PUSTKA, Daniel ; KLINKER, Gudrun: Dynamic Gyroscope Fusion in Ubiquitous Tracking Environments. In: *Proceedings of the 7th International Symposium on Mixed and Augmented Reality (ISMAR)*, 2008 see: 4.2, 5.3.5

[Ple11] PLECHER, David: *Klassifizierung von Anzeigeprinzipien in AR-Anwendungen anhand von definierten Paradigmata*, Technische Universität München, Diplomarbeit, 2011 see: 2.4

[PRA08] PATEL, S. ; REYNOLDS, M. ; ABOWD, G.: Detecting human movement by differential air pressure sensing in HVAC system ductwork: An exploration in infrastructure mediated sensing. In: *Pervasive Computing* (2008), S. 1–18 see: 3.2.1, 3.2.1, 3.2.1

[PT01] PIEKARSKI, W. ; THOMAS, B.H.: Tinmith-metro: New outdoor techniques for creating city models with an augmented reality wearable computer. In: *Proceedings of ISWC* Published by the IEEE Computer Society, 2001. – ISSN 1530–0811, S. 31–38 see: 5.2

[PTA06] PATEL, S. ; TRUONG, K. ; ABOWD, G.: Powerline positioning: A practical sub-room-level indoor location system for domestic use. In: *UbiComp 2006: Ubiquitous Computing* (2006), S. 441–458 see: 3.2.1

[Pus06] PUSTKA, Daniel: Construction of Data Flow Networks for Tracking in Augmented Reality Applications. In: *Proc. Dritter Workshop Virtuelle und Erweiterte Realität der GI-Fachgruppe VR/AR*. Koblenz, Germany, September 2006 see: 4.7.3

[RD06] Reitmayr, G. ; Drummond, T.: Going out: robust model-based tracking for outdoor augmented reality. In: *Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality* IEEE Computer Society, 2006. – ISBN 1424406501, S. 109–118 see: 5.2

[RDB01] Rolland, J.P. ; Davis, L. ; Baillot, Y.: A survey of tracking technology for virtual environments. In: *Fundamentals of wearable computers and augmented reality* (2001), S. 67–112 see: 2.4

[RMT02a] Roos, T. ; Myllymäki, P. ; Tirri, H.: A statistical modeling approach to location estimation. In: *IEEE Transactions on Mobile Computing* (2002), S. 59–69 see: 3.3.1

[RMT+02b] Roos, T. ; Myllymäki, P. ; Tirri, H. ; Misikangas, P. ; Sievänen, J.: A probabilistic approach to WLAN user location estimation. In: *International Journal of Wireless Information Networks* 9 (2002), Nr. 3, S. 155–164 see: 3.3.1

[RS01] Reitmayr, Gerhard ; Schmalstieg, Dieter: OpenTracker: An Open Software Architecture for Reconfigurable Tracking based on XML. In: *Proceedings of IEEE Virtual Reality.* Yokohama, Japan, 2001, 285-286 see: 4.2

[SC92] Strauss, Paul S. ; Carey, Rikk: An Object-Oriented 3D Graphics Toolkit. In: Catmull, Edwin E. (Hrsg.): *Proceedings of SIGGRAPH 1992* Bd. 26, 1992, 341–349 see: 4.2

[Sch11] Schlegel, Michael: *Zeitkalibrierung in Augmented Reality Anwendungen*, TU München, Diss., 2011 see: 10.2

[SFN08] Sielhorst, T. ; Feuerstein, M. ; Navab, N.: Advanced medical displays: A literature review of augmented reality. In: *IEEE/OSA Journal of Display Technology* 4 (2008), Nr. 4, S. 451–467. – ISSN 1551–319X see: 2.1

[SGTH04] Schwaighofer, A. ; Grigoras, M. ; Tresp, V. ; Hoffmann, C.: GPPS: A Gaussian process positioning system for cellular networks. In: *Advances in Neural Information Processing Systems* 16 (2004) see: 3.3.1

[SHR+08] Schuhmann, S. ; Herrmann, K. ; Rothermel, K. ; Blumenthal, J. ; Timmermann, D.: Improved weighted centroid localization in smart ubiquitous environments. In: *Ubiquitous Intelligence and Computing* (2008), S. 20–34 see: 7.2

[SL06] Speckmann, H. ; Ley, M.: Multi-media NDT procedures and online-maintenance assistance in the frame of the European R&D project INDeT

(Integration of NDT). In: *ECNDT 2006, 9th European Conference on Non-Destructive Testing, Berlin.* Berlin, Germany : Deutsche Gesellschaft fÃijr Zerstörungsfreie Prüfung e.V., 2006 see: 8.2

[SM10] SCHEER, F. ; MÜLLER, S.: Large area indoor tracking for industrial augmented reality. In: *9th IEEE International Symposium on Mixed and Augmented Reality (ISMAR)* IEEE, 2010, S. 269–270 see: 9.2

[Smi03] SMITH, Steven W.: *Digital Signal Processing.* Newnes, 2003 see: 6.3.2

[SS04] SCHWALD, B. ; SEIBERT, H.: Registration Tasks for a Hybrid Tracking System for Medical Augmented Reality. In: *Journal of WSCG* 12 (2004), S. 411–418 see: 5.2, 5.4.3

[SS05] SCHÖNFELDER, R. ; SPENLING, F.: The Planar: an interdisciplinary approach to a VR enabled tool for generation and manipulation of 3D data in industrial environments. In: *Virtual Reality, 2005. Proceedings. VR 2005. IEEE* IEEE, 2005. – ISBN 0780389298, S. 261–264 see: 9.2

[SS08] SCHÖNFELDER, R. ; SCHMALSTIEG, D.: Augmented reality for industrial building acceptance. In: *Proceedings of Virtual Reality Conference, 2008. VR'08* IEEE, 2008, S. 83–90 see: 9.2

[SSFG97] SZALAVÁRI, Zsolt ; SCHMALSTIEG, Dieter ; FUHRMANN, Anton ; GERVAUTZ, Michael: Studierstube - An Environment for Collaboration in Augmented Reality. In: *Journal of the Virtual Reality Society* 3 (1997), S. 37–48 see: 4.2

[SSK+07] SIELHORST, T. ; SA, W. ; KHAMENE, A. ; SAUER, F. ; NAVAB, N.: Measurement of absolute latency for video see through augmented reality. In: *Measurement* 2007 (2007), S. 1–4 see: 5.2

[Sut65] SUTHERLAND, I.E.: The ultimate display. In: *Proceedings of the IFIP Congress* Bd. 2 Citeseer, 1965, S. 506–508 see: 2.1

[SVL+06] SOHN, T. ; VARSHAVSKY, A. ; LAMARCA, A. ; CHEN, M. ; CHOUDHURY, T. ; SMITH, I. ; CONSOLVO, S. ; HIGHTOWER, J. ; GRISWOLD, W. ; DE LARA, E.: Mobility detection using everyday GSM traces. In: *UbiComp 2006: Ubiquitous Computing* (2006), S. 212–224 see: 3.3.1

[SVZ07] SCHNEEGANS, S. ; VORST, P. ; ZELL, A.: Using RFID snapshots for mobile robot self-localization. In: *Proceedings of the 3rd European Conference on Mobile Robots (ECMR 2007)*, 2007, S. 241–246 see: 3.3.1, 8.2

[TCD+00] Thomas, Bruce ; Close, Ben ; Donoghue, John ; Squires, John ; Bondi, Phillip de ; Morris, Michael ; Piekarski, Wayne: ARQuake: An Outdoor/Indoor Augmented Reality First Person Application. In: *Proceedings of the 4th IEEE International Symposium on Wearable Computers*. Washington, DC, USA : IEEE Computer Society, 2000 (ISWC '00). – ISBN 0–7695–0795–6, 139– see: 2.1

[TEHK09] Tönnis, Marcus ; Echtler, Florian ; Huber, Manuel ; Klinker, Gudrun: Low Cost 3D Rotational Input Devices: the stationary Spinball and the mobile Soap3D. In: *Proceedings of the 11th Symposium on Virtual and Augmented Reality (SVR)*, 2009 see: 9.2

[THS+01] Taylor, II, Russell M. ; Hudson, Thomas C. ; Seeger, Adam ; Weber, Hans ; Juliano, Jeffrey ; Helser, Aron T.: VRPN: a device-independent, network-transparent VR peripheral system. In: *Proceedings of the ACM symposium on Virtual reality software and technology*, ACM Press, 2001. – ISBN 1–58113–427–4, S. 55–61 see: 4.2

[TLWK07] Tönnis, Marcus ; Lindl, Rudi ; Walchshäusl, Leonhard ; Klinker, Gudrun: Visualization of Spatial Sensor Data in the Context of Automotive Environment Perception Systems. In: *Proceedings of the 6th International Symposium on Mixed and Augmented Reality (ISMAR)*, 2007 see: 2.1

[Tön10] Tönnis, Marcus: *Augmented Reality*. Springer Verlag, 2010 see: 2.1

[TPK09] Tönnis, Marcus ; Plavšić, Marina ; Klinker, Gudrun: Survey and Classification of Head-Up Display Presentation Principles. In: *Proceedings of the International Ergonomics Association (IEA)*, 2009 see: 2.4

[TRPČ09] Tippenhauer, N.O. ; Rasmussen, K.B. ; Pöpper, C. ; Čapkun, S.: Attacks on public WLAN-based positioning systems. In: *Proceedings of the 7th international conference on Mobile systems, applications, and services* ACM, 2009, S. 29–40 see: 7.2

[Ull76] Ullmann, J. R.: An Algorithm for Subgraph Isomorphism. In: *Journal of the ACM* 23 (1976), Nr. 1, S. 31–42. `http://dx.doi.org/http://doi.acm.org/10.1145/321921.321925`. – DOI http://doi.acm.org/10.1145/321921.321925 see: 4.6

[Val08] Valerio, D.: Open Source Software-Defined Radio: A survey on GNU-radio and its applications / FTW Technical Report. 2008. – Forschungsbericht see: 7.5

[VCL+06] Varshavsky, Alex ; Chen, Mike ; Lara, Eyal de ; Froehlich, Jon E. ; Fox, Dieter ; Hightower, Jeffrey ; LaMarca, Anthony ; Potter, Fred ; Sohn, Tim ; Tang, Karen ; Smith, Ian: Are GSM phones THE solution for localization? In: *HotMobile* (2006). – ISSN 1550–6193 see: 3.3

[VLH+07] Varshavsky, A. ; Lara, E. de ; Hightower, J. ; LaMarca, A. ; Otsason, V.: GSM indoor localization. In: *Pervasive and Mobile Computing* 3 (2007), Nr. 6, S. 698–720. – ISSN 1574–1192 see: 3.3, 3.3.1

[VSYZ08] Vorst, P. ; Schneegans, S. ; Yang, B. ; Zell, A.: Self-localization with RFID snapshots in densely tagged environments. In: *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on* IEEE, 2008, S. 1353–1358 see: 8.2, 8.4.5

[Wag04] Wagner, Martin: Distributed Tracking with Multiple Sensors for Augmented Reality. In: *1. Workshop "Virtuelle und Erweiterte Realität*. Chemnitz, Germany, September 2004 see: 4.2

[Wag05] Wagner, Martin: *Tracking With Multiple Sensors*, TU München, Diss., 2005 see: 4.2

[WB97] Welch, Greg ; Bishop, Gary: SCAAT: Incremental Tracking with Incomplete Information. In: Whitted, Turner (Hrsg.) ; ACM SIGGRAPH (Veranst.): *SIGGRAPH 97 Conference Proceedings* ACM SIGGRAPH, Addison Wesley, August 1997 (Annual Conference Series). – ISBN 0–89791–896–7, 333–344 see: 5.2

[Wei91] Weiser, M.: The computer for the 21st century. In: *Scientific American* 265 (1991), Nr. 3, S. 94–104 see: 2.2

[Wei93] Weiser, M.: Ubiquitous computing. In: *Computer* 26 (1993), S. 71–72. – ISSN 0018–9162 see: 2.2

[Wei98] Weisstein, Eric W.: *CRC Concise Encyclopedia of Mathematics*. Chapman Hall CRC, 1998 see: 5.3.7

[Wel96] Welch, Gregory F.: *SCAAT: Incremental Tracking with Incomplete Information*, University of North Carolina at Chapel Hill, Diss., 1996 see: 5.2

[WF02] Welch, G. ; Foxlin, E.: Motion tracking: No silver bullet, but a respectable arsenal. In: *Computer Graphics and Applications, IEEE* 22 (2002), Nr. 6, S. 24–38. – ISSN 0272–1716 see: 2.4

[WHFG92] Want, R. ; Hopper, A. ; Falcao, V. ; Gibbons, J.: The active badge location system. In: *ACM Transactions on Information Systems (TOIS)* 10 (1992), Nr. 1, S. 91–102. – ISSN 1046–8188 see: 3.3.1

[WHK04] Wagner, Martin ; Hennauer, Sven ; Klinker, Gudrun: Easing the Transition Between Multiple Trackers. In: *Proc. of IEEE and ACM International Symposium on Mixed and Augmented Reality*. Arlington, VA, USA, November 2004 see: 4.2

[WHK+10] Waechter, Christian ; Huber, Manuel ; Keitler, Peter ; Schlegel, Michael ; Pustka, Daniel ; Klinker, Gudrun: A multisensor platform for wide-area tracking. In: *9th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2010)*. Seoul, Korea, October 2010 see: 9, 9.1, 10.2

[Wil07] Wilson, Mark J. (Hrsg.): *The ARRL Handbook For Radio Communications*. eightyfourth. The American Radio Relay League, 2007 see: 6.1

[WKC07] Whitehouse, K. ; Karlof, C. ; Culler, D.: A practical evaluation of radio signal strength for ranging-based localization. In: *ACM SIGMOBILE Mobile Computing and Communications Review* 11 (2007), Nr. 1, S. 41–52. – ISSN 1559–1662 see: 3.3.1

[WLS+07] Wang, H. ; Lenz, H. ; Szabo, A. ; Bamberger, J. ; Hanebeck, U.D.: WLAN-based pedestrian tracking using particle filters and low-cost MEMS sensors. In: *Positioning, Navigation and Communication, 2007. WPNC'07. 4th Workshop on* IEEE, 2007. – ISBN 1424408717, S. 1–7 see: 3.3.1, 7.2

[Zho06] Zhou, R.: Wireless indoor tracking system (WITS). In: *doIT Conference on Software Research*, 2006, S. 163–177 see: 3.3.1, 7.2