FAKULTÄT FÜR INFORMATIK

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

Determining Interaction Geometry with Ultrasound Sensors

Lukas M. Märdian

FAKULTÄT FÜR INFORMATIK

TECHNISCHE UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Informatics

Determining Interaction Geometry with Ultrasound Sensors

Bestimmung von Interaktionsgeometrie durch Ultraschallsensoren

Author:	Lukas M. Märdian
Supervisor:	PD Dr. Georg Groh
Advisor:	Dipl.-Inf. Alexander Lehmann
Advisor:	Dr. Michele Brocco
Date:	August 29th, 2013

I assure the single handed composition of this bachelor's thesis only supported by declared resources.

Munich, August 29th, 2013 Lukas M. Märdian

Acknowledgements

During the creation of my bachelor's thesis I hit some very difficult problems from time to time, where I did not know how to proceede any further. I would like to thank the people who helped me out in such situations and thus keept my research going.

First of all I would like to thank my supervisor PD Dr. Georg Groh and my advisors Dipl.-Inf. Alexander Lehmann and Dr. Michele Brocco for supporting and guiding me throughout the whole process of this scientific work.

Next, I would like to thank Dora Dzvonyar for her cooperation with me during our experiments and for all the organizational and marketing work she did, enabling us to have enough participants for the experiments.

I would also like to thank Dr. H. Nikolaus Schaller from Golden Delicious Computers for the lowlevel hardware support he provided me with, when I had big problems with the USB subsystem of my tracking devices.

And last but not least I would like to thank Michele Brocco, Rene Leitner, Christoph Mair and Orest Tarasiuk from the OpenPhoenux community for borrowing their Letux 2804 smartphones to me, which I used during my experiments.

Abstract

In this thesis I present an ultrasound tracking system, which works infrastructure less and is based on mobile phones. I present the software and hardware, which is needed for this system to work. Then I describe the experiment I conducted to collect data of the system's performance. Afterwards, this data gets evaluated in a way that it is compared to a high precision reference data set, which was already used to determine interaction geometry. Next, I discuss the system's and experiment's limitations and suggest some concrete improvements. Finally I conclude on the possibilities to determine interaction geometry, using the data my ultrasonic tracking system is able to collect and propose some future work in this area.

Zusammenfassung

In dieser Abschlussarbeit stelle ich ein von Infrastruktur unabhängiges, Ultraschall basiertes System zur Erhebung von Bewegungsdaten vor, welches auf Smartphones eingesetzt werden kann. Ich erkläre die Software und Hardware, welche vom System benötigt wird und erläutere das Experiment, welches ich durchgeführt habe um Daten zu den Leistungsmerkmalen des Systems zu erheben. Anschließend werden diese Daten ausgewertet, indem sie mit einem sehr genauen Referenz-Datensatz verglichen werden, mit Hilfe dessen bereits Interaktionsgeometrie bestimmt wurde. Danach werde ich einige Einschränkungen des Systems und des Experiments aufzeigen und konkrete Verbesserungen vorschlagen. Abschließend werde ich noch die Möglichkeiten aufzeigen, wie Interaktionsgeometrie mit Hilfe der Daten meines Systems bestimmt werden kann und weiter Arbeiten in diesem Themengebiet vorschlagen.

Keywords

social signal processing, SSP, social interaction geometry, interpersonal distance and orientation, social situation detection, ultrasound sensor, ultrasonic tracking

Contents

Outline of the Thesis

Part I: Introduction

CHAPTERS 1: INTRODUCTION

This chapter presents an overview of the thesis and its purpose. Furthermore, it presents the problem statement and motivation of the thesis.

Part II: Basics

CHAPTER 2.1: SOCIAL SIGNAL PROCESSING

In this chapter the social signal processing basics, used in this thesis, are introduced, by condensing and referring to related research in this area.

CHAPTER 2.2: DISTANCE MEASUREMENT WITH MOBILE DEVICES

This chapter presents other approaches on how to collect the data required to determine interaction geometry. It shows the possibilities of different sensor technologies and the different infrastructures involved.

CHAPTER 3.1: DECENTRALIZED TRACKING WITH ULTRASOUNDS

Chapter 3.1 describes the idea of a mobile, ultrasound based tracking system, usable to gather information for further application.

CHAPTER 3.2: DETERMINING INTERACTION GEOMETRY WITH ULTRASOUND SENSORS

This chapter summarizes the idea of an infrastructure less and decentralized tracking system, based on ultrasound sensors. Furthermore, it shows how this concept delimits itself from the aforementioned approaches and describes a plan how to evaluate the system.

Part III: Implementation

CHAPTER 4.1: HARDWARE

In this chapter the hardware built and used in the thesis are presented and its capabilities are described.

CHAPTER 4.2: IMPLEMENTATION PROBLEMS

Chapter 4.2 discusses the difficulties of a decentralized trigger and the problem of clock synchronization between tracking devices.

Part I.

Introduction

1. Introduction

1.1. Problem

These days smartphones are omnipresent in western cultures, most people carry one of those small computers with them all the time. They are mostly used to handle the interaction between the virtual world of social networks, emails and instant messages and the real world of the smartphone's user himself. So in fact they are tools to support the communication of people, but up to now mostly in the virtual world – even if two users would be nearby.

There is a discrepancy of how smartphones support our interaction in the virtual world and how they support our interaction in the real world: Up to now, they do very little to support the communication and interaction between people in the real world, whereas most communication in the virtual world is done using smartphones. The problem here is, that smartphones can easily receive and send signals and notifications from and to the virtual world (e.g. servers and other computers connected to a network), due to the fact that they are just little computers with networking capabilities. But they are not good in gathering information from the real world around them, e.g. the social situations the smartphone's user is in and thus they are not able to provide any support to their user in respect of his current real world situation. The user always needs to explicitly tell his phone what to do, e.g. call somebody. Smartphones should be able to autonomously gather information from the real world, without the user having to command, e.g. detecting the importance/relationship of the current conversation partner, in order to reject disturbing, less important incoming calls.

1.2. Motivation

Current research presents methods of how computers can be integrated with the real world the users live in. In the context of this thesis this means how computers are able to detect social situations and interactions of their users. It is already possible to detect social situations between people using computers in static environments [Groh et al., 2010] as well as on the go, using turn-taking patterns [Groh et al., 2011b] or *Received Signal Strength Indication* (RSSI) analysis [Matic et al., 2012] [Banerjee et al., 2010].

Detecting social situations in fixed environments reveals only a fraction of the potential of social interaction detection, as the people do not stay in the same places all the time. This is why this thesis' focus is on mobile, decentralized and infrastructure less detection of social interactions. Detecting social situation using mobile phones is already possible, by having them listen to their users conversation and analyzing it, in order to detect turn taking patterns. Having the smartphones record peoples conversations all the time has ethical problems, though. People do not like to have every word they say recorded. Furthermore, audio analysis is rather error prone in crowded places with a lot of background noise.

Analyzing the RSSI values of Bluetooth and/or WiFi signals is another possibility, usable on today's smartphones, which unfortunately does not yield a very good performance. This is why my thesis will focus on detecting social situations with ultrasound sensors, assuming those will be built into tomorrow's smartphones. This assumption is valid, because ultrasonic transceivers already are very cheap in price and are able to measure distances, whilst consuming very few energy. If the ultrasonic transceivers shrink in size a little bit more in the future, as most other electronic components do, it should be fairly easy to integrate them into a smartphone.

1.3. Overview

The thesis is structured in the following way: Part II presents information on related research topics, e.g. the idea of interaction geometry and other distance measurement methods. Furthermore, it delimits those topics to the topic of ultrasound sensors determining interaction geometry.

In part III the hardware, used in this thesis' experiments, is presented and the software, which was created to collect and analyze the experimental data, is described. In addition, this part presents some implementation difficulties, which arose during the research.

Part IV introduces the experimental scenario, evaluates the experimental data and shows how this knowledge can be used to detect social situations on mobile phones in a decentralized and infrastructure less manner. Furthermore, it discusses the system's limitations.

A conclusion of the research is given in part V and some possible future work in this research topic is outlined.

Part II.

Basics

2. Related Work

2.1. Social Signal Processing

A social interaction between people can be a variety of things. It can either be a conversation, eye contact or a physical interaction, e.g. helping somebody to carry a box. It is always some sort of contact, be it physical, verbal or mental. Such situations can be detected by computers in a variety of ways, utilizing a lot of different data sources. The interpersonal distance and relative body orientation are a very reliable indicator for social situations, thus they are used in *Interaction Geometry*.

2.1.1. Interpersonal Distance

One of the most significant characteristics in detecting social sitations are the interpersonal distances, which are related to socioemotional closeness of the persons involved: "The best way to learn the location of invisible boundaries is to keep walking until somebody complains. Personal space refers to an area with invisible boundaries surrounding a person's body into which intruders may not come" [Sommer, 1959]. Most commonly the interpersonal distances are categorized into *four concentric zones*, defining a relationship between the interacting people, based on the zone they interact in [Hall, 1959]. The following distances are typical and valid only for western cultures, such as Europe or the United States (cf. [Mehrabian, 1972] [Sundstrom and Altman, 1976] [Willis Jr., 1966]):

- *Intimate Zone* (< 0.5m): This zone is usually used by close relatives or intimate partners, only. Non-permitted entry into this zone is normally avoided, whenever possible (e.g. it might be impossible in public transport).
- *Casual-Personal Zone* (0.5m – 1.2m): This distance is normally used whilst interacting with familiar people, like friends or colleagues. It reflects sympathy between the interaction partners.
- *Socio-Consultive Zone* (1.2m – 2.0m): In this zone body contact is not possible any more and it is typically used by interaction partners with a formal and impersonal relationship, e.g. for business conversations.
- *Public Zone* (> 2.0m): The public zone is usually not used for any personal interaction, the only exceptions being speakers in front of an audience, theater actors or interactions disturbed by the presence of some physical barrier.

In addition to the social relationship between interaction partners, psychological aspects play a certain part in choosing interpersonal distances, too: Introvert people tend to extend the interpersonal distance slightly, whereas extrovert people behave to the contrary.

2.1.2. Body Orientation

Another factor in determining social interactions is the relative body orientation between two people. This orientation, in which one person is facing the other, is mainly based on the persons torso orientations, rather than the persons angles of view. Males tend to orientate themselves directly towards each other in hostile situations, as this provides better courtesy through direct eye-contact [Mehrabian, 1972] [Argyle and Schmidt, 1989]. Moreover, "facing in the opposite direction with respect to others is a clear sign of non-inclusion [...]. Face-to-face interactions are in general more active and engaging (the frontal position addresses the need of continuous mutual monitoring), while people sitting parallel to each other tend to be either buddies or less mutually interested" [Vinciarelli et al., 2009].

2.1.3. Interaction Geometry

The *Interaction Geometry* is a method of determining social situations between two people, based upon their relative geometric alignment towards each another. It is especially based on the relative distance between those people's body center (*Interpersonal Distance*) and the angles between their headings (*Body Orientation*). Considering several people in a group, those values are compared in pairs of two. The aforementioned relative body orientation, in conjunction with the current interpersonal distance between two people, can be used by interaction geometry models, to determine if the two people are currently having a social interaction, as proposed by Groh et al.: "We furthermore propose to consider only relative distance of body centers (projected onto the xy-plane) and angle of the relative rotation around the z-axis" [Groh et al., 2010].

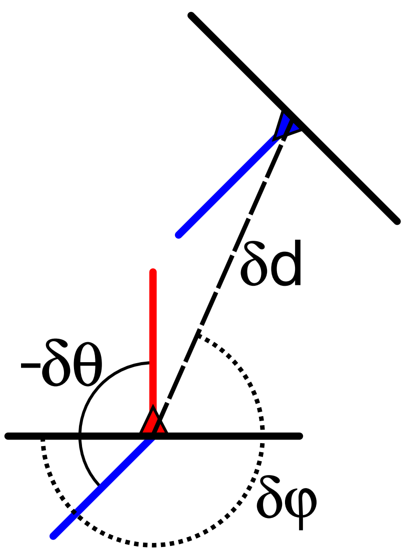

Figure 2.1.: Interaction Geometry Parameters

The relative distance between two people is called δd, while the angle of their relative body orientation is called $\delta\theta$. The angle $\delta\theta$ is measured between two people's heading and is ranging from $-\pi$ to $+\pi$, 0π meaning the interaction partners are facing into the same direction and $\pm\pi$ meaning that they are facing each another. Later on another angle $\delta\varphi$ was proposed, which describes the angle at which the interaction partner is positioned in respect to the own heading direction (cf. figure 2.1).

The data collected with the proposed ultrasound based tracking system is supposed to be fed into the *interaction geometry model*, which will then use machine learning methods based on a *Gaussian Mixture Model*, to determine if a social interaction is currently happening or not (also compare [Groh et al., 2010] [Groh et al., 2011a] [Groh and Lehmann, 2011]).

2.2. Distance Measurement with Mobile Devices

Several ways of how to determine the position of mobile devices and their relative distances to each another already exist, both in static environments, such as a fixed infrared tracking system, installed in a specific room and in dynamic environments, such as measuring with the mobile devices itself. This thesis' focus is on the dynamic environments, as an infrastructure less and decentralized system for distance measurements shall be researched. An overview of those properties of the presented system's is given in table 2.1.

2.2.1. GPS and WiFi Localization

The triangulation of GPS satellites is based on an infrastructure of geostationary satellites around the earth. Using GPS it is possible to determine the position of a device in outdoor situations, GPS is not able to directly determine the orientation of a device, though. Therefore a double differentiation of the measured positions would be needed, which implies the necessity of movement. Furthermore, the GPS standard specifies a root mean square error of 4m [US-DepartmentOfDefense, 2008], which in turn makes it unusable for determining the interpersonal distances, usually being below 2 meters. Triangulation of WiFi access points, in contrast to the GPS system, is also usable in indoor situations and provides a better positioning accuracy (provided a good enough access point infrastructure). But still it is not possible to directly determine the orientation of a device with this system and it is heavily dependent on a good infrastructure of access points of known location.

2.2.2. RSSI Analysis

The analysis of *Received Signal Strength Indication* (RSSI) is applicable on most receiving and transmitting radio techniques, such as WiFi, Bluetooth and FM. Additionally, it is not necessarily dependent on an infrastructure of access points, as it is conceivable that one mobile device could function as a sender, the other device as a receiver and vice versa. Banerjee et al. presented their *Virtual Compass* project [Banerjee et al., 2010], which is based on a combination of RSSI analysis of WiFi and Bluetooth signals. They were able to determine the relative distance between two devices with a median accuracy of 0.9m to 1.9m, but they have no solution for determining the relative body orientation, as needed by interaction geometry models. Matic et al. came up with a proposition to use WiFi RSSI analysis in combination with the orientation sensors, embedded into today's smartphones, to determine the relative body orientation as well as the relative distance between two devices

[Matic et al., 2012]. In addition they applied machine learning techniques to reach a median estimation error of 0.5m and 10°, which is already a pretty good performance. But it is still hard to correctly determine the current interpersonal distance between two interaction partners with this accuracy, especially in the corner cases.

2.2.3. Radio and Ultrasounds

In comparison to the aforementioned techniques, distance measurements using ultrasounds showed a pretty good accuracy in the past. The *BAT System* by Harter et al. [Harter et al., 2002] for example is a system, consisting of active, mobile ultrasound senders (BATs), which are carried by the people. They send out short ultrasound pulses frequently, which are received by several beacons, installed at the ceiling. These beacons are connected to a central computer, which can then calculate the exact position of every single BAT. This system has a privacy problem, though, as movement profiles of all users are analyzed on a central computer. This issue is solved in the MIT's *Cricket System*, as proposed by Priyantha et al. [Priyantha et al., 2000]. Their system consists of passive, mobile receivers and active beacons with known coordinates, installed at the ceiling. This way the devices can determine their position with a very high accuracy and without loosing control over their data. Still, both systems have no solution to the problem of determining the relative body orientation. That is why the *Cricket Compass* was proposed by Priyantha et al. [Priyantha et al., 2001]. The *Cricket Compass* combines the passive approach of the *Cricket System* and adds the capability to determine the current orientation of the mobile device. This is achieved by placing several ultrasonic receivers in a special spatial arrangement and calculating the time of flight of a signal send by an active beacon for each of the receivers. This way the *Cricket Compass* is able to determine the position of a device withing approximately 15cm and the rotation angle within ± 8 degrees. This is a very good accuracy, which could be used in combination with interaction geometry models to decide if a social interaction is taking place or not. But still, both the *BAT System* and the *Cricket Compass* are in need of a fixed infrastructure of beacons.

Table 2.1.: Overview of Measurement Techniques

	Ultrasound Based	Infrastructure Less	Accuracy
GPS and WiFi Localization	✗	✗	2–4 m
RSSI Analysis	✗	✓	50 cm
Cricket and BAT	✓	✗	15 cm

3. Conception

3.1. Decentralized Tracking with Ultrasounds

As shown in the related work (chapter 2), the best result in tracking people is achievable with ultrasound sensors. This is why this thesis' focus is on an ultrasound based tracking system. In contrast to the most systems mentioned above (except the RSSI analysis), the system proposed in this thesis is meant to be completely infrastructure less, so it can be used everywhere and is not limited to a specific geographical area. And in contrast to the RSSI analysis methods mentioned above it is based on ultrasound sensors, to achieve an as good as possible accuracy.

Using ultrasound sensors, there are two possible measurement techniques. The easier technique is the *echo mode*, where only a single ultrasound transceiver is needed, which sends out an ultrasonic pulse and counts the time of flight, until the echo is received. This way the distance to a unknown object can be determined in a very precise resolution. The problem is that the object is unknown, though. This is why in this thesis a *sender-and-receiver mode* is used. In this mode at least two ultrasonic transceivers are needed. The first transceiver is the sender and sends out an ultrasonic pulse. The second transceiver is the receiver and records the precise timestamp when the sender's ultrasonic pulse was received. Once the receiver gets to know the exact timestamp of when this pulse was send out by the sender, it can calculate the time of flight and thus the distance to the sender.

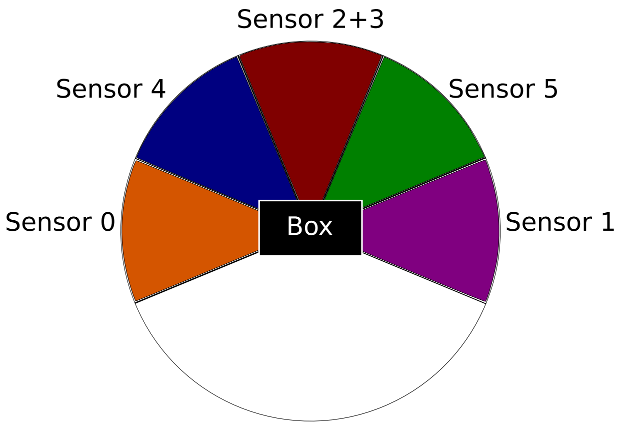

Figure 3.1.: Five sectors of the ultrasonic sensor array

For this thesis a sensor array consisting of six ultrasonic transceivers is used, which is connected to a Linux based smartphone. The ultrasonic sensor array consists of six ultrasonic transceivers, which are placed in a 10cm x 3.5cm x 7cm box, pointing to the outside in five different directions. They are arranged to cover an angle of 225°, divided up into five capturing sectors, whereas the front sector is redundantly captured by two sensors (see figure 3.1). It is positioned at the right side of the hip of the people participating in the experiments. This is a simplification over a real world scenario, where the interacting people would place their smartphones in their pockets in an arbitrary position and orientation. This is alright, though, as it is possible to determine the relative orientation of a smartphone, placed in a pocket in relation to its user's body orientation, using the gyroscope and accelerometer, which are usually build into today's smartphones, as shown by Shi et al. [Shi et al., 2011].

3.2. Determining Interaction Geometry with Ultrasound Sensors

The proposed ultrasound based tracking system is able to measure the relative distance and the relative body orientation between two people, carrying a mobile device (smartphone). The resulting data of this system is mean to be evaluated with a *Gaussian Mixture Model*, to conclude if a social interaction is currently happening between two people. In contrast to the aforementioned measurement systems, this system works in a fully decentralized and infrastructure less way. Furthermore, it is limited to measure the relative distance and relative body orientation between two people, instead of absolute coordinates, as most of the infrastructure dependent systems do.

The tracking hardware, as described in chapter 3.1, is used to conduct a series of experiments, where people in social situations (e.g. a conversation) are being tracked with both, the proposed ultrasound tracking system and a static, high precision infrared tracking system, for comparison. The data collected in those experiments is used to provide the relevant, measured performance and accuracy parameters the system is able to achieve.

In the evaluation the distance and orientation measurements of the proposed ultrasound system are compared to the data of the high precision infrared system, to conclude if the system is usable in the context of *Social Signal Processing*, to determine the interaction geometry of two interaction partners, when being combined with machine learning techniques and a *Gaussian Mixture Model*, as proposed by Groh et al. [Groh et al., 2010].

Part III.

Implementation

4. Implementation

In this chapter the technical soft- and hardware system, which was built to capture and analyze the data retrieved during my experiments, is presented. On the hardware it consists of sensor arrays of six ultrasonic transceivers each. Each of those sensor arrays is connected to a Linux based *Goldelico Letux 2804* smartphone to form an *agent*, which is carried by the test persons and servers as a controller and mobile capturing device.

4.1. Hardware

4.1.1. SRF02 Ultrasonic Transceiver

The SRF02 ultrasonic transceiver is a very power efficient and small ultrasonic measuring device by Robot Electronics. Its typical power consumption is just about 5mA at 5V and its size is just 24mm x 20mm x 17mm. It is controllable via an I2C bus or a serial line. This makes it a perfect fit for the mobile use case described in this thesis. The SRF02 is specified for measurements from 18cm up to 6m, with an accuracy of about two centimeters. In my verification measurements it turned out that the SRF02 does very accurate measurements of ±1cm in the range from 20cm to 500cm but has problems on bigger distances, starting at 5m (compare figure 4.1).

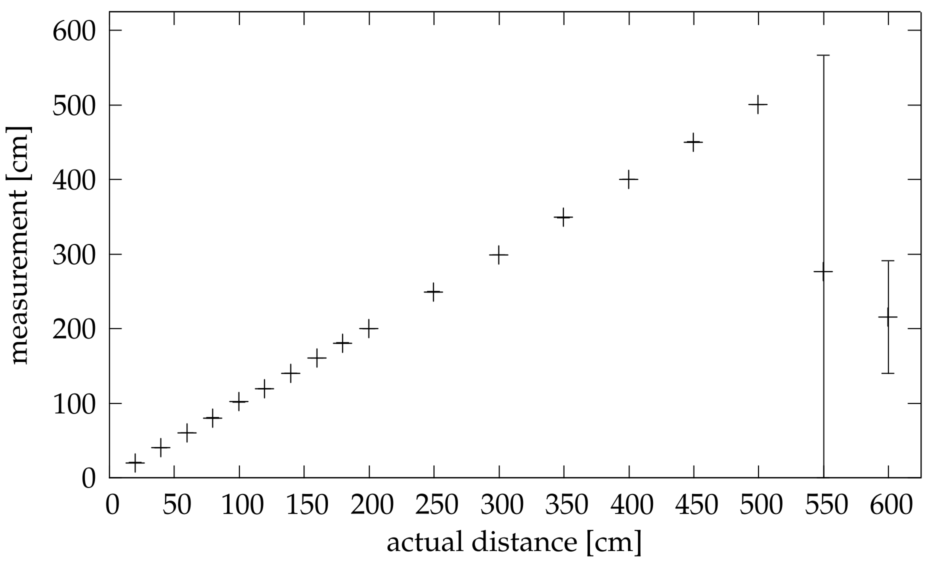

Figure 4.1.: SRF02 accuracy

The verification measurements were done using a single SRF02 ultrasound sensor in *echo mode* (compare chapter 3.1). Thus if two or more sensors are used together in *send-and-receive mode* (compare chapter 3.1), as done in the experiments, the maximal possible tracking distance is doubled to approximately ten meters, as the ultrasound pulse does not need to travel the way back to the sender, as necessary in the *echo mode*.

4.1.2. Ultrasonic Tracking Device

The ultrasonic tracking devices, used in the experiments, are build from an ultrasonic sensor array (see chapter 3.1 and figure 4.2), combined with a Letux 2804 smartphone. In the experiment six ultrasonic transceivers got connected with one of Robot Electronic's USB-I2C adapters [RobotElectronics, b], which got connected to the smartphone's USB-OTG port.

The OpenPhoenux GTA04, which is the base of the Goldelico Letux 2804 smartphone, is an open hardware platform, which provides its users with the ability to study, extend and modify the system's hardware and software, by providing free and open documentation, schematics and an unlocked bootloader [GoldenDeliciousComputers, 2011]. At its core it is a typical smartphone platform, providing an ARM Cortex-A8 powered OMAP3 SoC with an 800 MHz clock and 512 MiB of memory. In addition it provides a lot of peripherals like an UTMS modem, a 2.8" resistive touchscreen, WiFi, Bluetooth and a lot of sensors, like an accelerometer, a gyroscope and a digital compass. For external communication it provides two hardware buttons, a headset jack and an USB-OTG jack. All of this technology is put into a handy form factor of 12cm x 6cm x 2cm, which makes it a nice fit for a mobile tracking device.

Figure 4.2.: Ultrasonic sensor array, consisting of six sensors

4.2. Implementation Problems

The mutual measurements of the tracking devices are based upon time of flight calculations, so device A needs to know when device B sent out an ultrasonic burst. Only this way it is possible for device A to calculate the distance to device B, based on the time difference between when it received the ultrasonic burst and when device B sent out that burst. Therefore a common starting trigger for all devices is needed. After all devices are triggered synchronously, they can count the time independently, to stay in sync.

4.2.1. Common Trigger and Synchronization

The first attempt to implement a synchronous start on all devices was to use the Letux 2804's accelerometer as a trigger. The idea was to stack the smartphones, shake them all together at the same time and detect the acceleration peak on all devices, using the accelerometer, to start the measurements. It turned out that the accelerometer has a maximum sample rate of only 1200 Hz [BoschSensortec, 2010], though, which led to a resolution of about 830 microseconds. This is by far to inaccurate, as a 830 microseconds delay would already introduce an error of about 284 millimeters, assuming a speed of sound of $343\frac{m}{s}$.

As the accelerometer turned out to be to inaccurate, another approach was to use the ultrasonic signals itself to synchronize the tracking devices. One device was declared as *Master*, providing the others with its clock, by sending out an ultrasonic pulse every fixed interval. The other *Slave* devices, which were placed at a known distance to the master, periodically listened for pulses from the master and adjusted their listening interval, depending on the difference between the real (known) distance and the measured distance. After all devices were synchronized to the master's clock, they started the synchronous measurements. Using this approach it is possible to synchronize all tracking devices to the master's clock with an error of just ± 50 microseconds, which leads to a maximum misalignment of about 34 millimeters (assuming a speed of sound of $343\frac{m}{s}$).

4.2.2. Drifting Clocks

Another problem of synchronous measurements, besides a common starting trigger, are hardware clocks drifting away from each another: The different tracking devices' clocks drifted away from each other, not keeping the common time interval (tick) and thus loosing their synchronization. The usage of hardware based *High Resolution Timers* [Gleixner and Molnar, 2008], as provided by the Linux kernel and setting a real time priority to the measurement process, using the *chrt* command made the drift become a lot smaller, but the drift was still too big after those optimization. Two devices drifted away from one another to a point where they could not hear each other anymore after about 30 seconds.

4.2.3. Common External Clock

Drifting clocks made it impossible to collect any data in a real experiment of ten minutes length, as the devices completely lost their synchronization after just 30 seconds. An attempt to fix this issue, was to send a periodic broadcast via a wireless 802.11 channel to the tracking devices, which contains the information of which device is supposed to send out an ultrasonic burst at a specific point in time. Measuring the time a WiFi broadcast takes to reach its destinations showed, that it could vary by 2.5ms. This is already a big overhead, which leads to a positioning error of up to 85 cm (assuming a speed of sound of $343\frac{m}{s}$), presumably introduced by the sender's, receiver's and WiFi router's network stacks, which the signal needs to go through.

As a last attempt all tracking devices got connected to one external host computer, using an audio cable from the host to the smartphones' headset jacks. The external host computer now provides a single, common audio tick, which is interpreted as a common clock on all tracking devices. This way it is possible to keep all devices synchronized during a long time period, enabling them to calculate the correct times of flight and thus output the correct spatial distances to its neighbors. The maximal sample rate of 48kHz and the smallest possible period size of 8 samples are used, reducing the latency of the audio signal to a minimum. But still the signal is processed on every device with a slight delay: An accuracy test, where two devices were placed at a fixed distance of 60cm, resulted in an average measured distance of 59.88cm with a standard deviation of 5.78cm, which was expected, assuming the aforementioned parameters and a speed of sound of $343\frac{m}{s}$:

$$\frac{1}{48000}s \cdot 8 \cdot 343\frac{m}{s} \approx 5.72cm$$

This equation implies that once the synchronization problem is fully solved, the system's distance measurement accuracy will be improved by about 5.7cm.

4.3. Software

4.3.1. Data Capturing

The data capturing software for the experiments is written in *Python* and is tested to run on a *Debian GNU/Linux* system. To provide all its functionality it depends on the following components:

- Python $\geq$ 2.7.3
- PySerial $\geq$ 2.5
- PyAlsaaudio $\geq$ 0.5.

The application is split into four parts: The *SRF02* class is a hardware abstraction layer for the ultrasonic sensors. The *TokenRing* class controls which sensor of which device is allowed to send out an ultrasonic burst at a certain point in time. The *SSPtool* script is the main entry point of the application, which assures the synchronization and parallel start of all devices. Finally, the *Host Software* functions as an external clock to all agents, so they can stay synchronized.

Class: SRF02

The SRF02 class is a hardware abstraction layer for the ultrasonic sensors, providing a sane *Application Programming Interface* (API) to control the ultrasonic transceivers. This API follows the naming conventions of the SRF02's technical specification [RobotElectronics, a] very tightly and is meant to be used in conjunction with Robot Electronic's USB-I2C adapter.

The class uses the PySerial API to send bytes in a specified order to a specified SRF02 transceiver, in order to trigger the execution of a specified command. An object of this class should be initialized by passing the address of the desired sensor and the serial port of the USB-I2C adapter, which it is connected to, e.g.: *sensor = SRF02(addr='e0', port='/dev/ttyUSB0')*. Using this object it is for example possible to use the *range_usec()* call, to start a ranging with a microsecond resolution. In this case the application would send this sequence of bytes to the USB-I2C adapter: *0x55 0xE0 0x00 0x01 0x52*, which would trigger an ultrasonic ranging on sensor 0xE0. Afterwards, using the *get_result()* call, it is possible to send *0x55 0xE1 0x00 0x02 0x02*, which would return the ranging result in microseconds.

Class: TokenRing

The TokenRing class handles the sequence in which the sensors listen to or send out their bursts, so that they do not overlap. Therefore it provides a method named *handler*, which should be called by the main application in certain intervals. These intervals should not be shorter than 150 milliseconds, as this is the time the routine needs to process all the sensor data. To initialize an object of this class, the serial port of the device's USB-I2C adapter should be passed as an argument, e.g. *device = TokenRing(port='/dev/ttyUSB0')*.

During one run of the *handler* routine, the currently active sensor of the currently active agent sends out an ultrasonic 40 KHz burst and catches its echo. All sensors of currently passive agents just listen and capture the time at which the active sensor's ultrasonic burst reaches them. As all devices use a single external clock, the passive devices know when the burst had been started and thus its time of flight. Using the burst's time of flight and the speed of sound (about $343\frac{m}{s}$, at 20°C), the passive devices can calculate the distance between the active sensor and each of its passive sensors. Due to the synchronization they furthermore know which sensor of which agent is currently active and thus know exactly which device is at which distance at this point in time.

In addition to the distance, the passive agents can also determine the approximate position of the sender, as the agents' sensors are positioned in a way that every sensor points into another direction. So it is possible to determine at which sensor the burst arrived first and thus the position of the sender can be approximated in the direction this sensor is pointing to.

At the end of the *handler* routine, the object's internal counters are being set. Each object has a counter for the currently active ultrasonic transceiver and a counter for the currently active agent. Using those counters the agents form a token ring, which starts at agent 0, sensor 0. In each tick the token is passed on to the next agent. After the token is back at agent 0, the sensor counter is increased and the active sensor will be agent 0, sensor 1. In a group of three agents this builds the following sequence of active sensors (senders):

```
agent 0, sensor 0
agent 1, sensor 0
agent 2, sensor 0
agent 0, sensor 1
agent 1, sensor 1
agent 2, sensor 1
...
```

Once the token traversed all sensors of all agents, it is passed back to the beginning (agent 0, sensor 0) in a cyclic manner. Passing the token between agents several times a second, instead of passing it to all sensors of one agent first and then on to the next agent's sensors was chosen to be able to better capture dynamic situations, which could occur between interacting people.

Main Application: SSPtool

The application's main entry point is defined in *SSPtool.py*. To start capturing data on the mobile devices, it is necessary to start the SSPtool application on all devices in the group. To start the application at least two arguments are needed: The first being the ID of the device, the second defines the number of agents in the group. The third argument is optional and defines the serial port to use. To start measuring on the device with ID 2 in a group of four devices, SSPtool would be started as follows: *./SSPtool.py 2 4 /dev/ttyUSB0*.

Once SSPtool is started on all devices in the group it will wait for the external clock to start ticking. The external clock will always send six logical ones and afterwards a multiple of six logical zeros, depending on the number of agents in the group. In the case of four agents, the clock would look like this: *111111 000000 000000 000000*. Each device listens until it recognizes the six logical ones, then it sets its internal counters according to its ID and starts executing the handler on every tick, be it either a logical one or a logical zero. Due to the fact that all agents start after six logical ones were received, they are synchronized to a common interval.

Host Software

The host software is a rather simple tool, which runs on an external computer and sends out the aforementioned clock signal (e.g. *111111 000000 000000 000000*) to all agents in parallel. In addition to this, the host software captures the timestamp at which the synchronization was started (this is the point in time at which the host software got executed and all agents were already waiting for the synchronization signal to arrive) and the timestamp at which the experiment started, which will be used in the data analysis software later on.

To run the host software a single argument is necessary: The number of agents participating in the current experiment, e.g.: *./host.py 4*. To capture the second timestamp (start of the experiment), the supervisor has to press [Enter] while the host software is running.

4.3.2. Data Analysis

The data analysis software is written in Python as well and is meant to be run on a host computer. To provide all its functionality it depends on Python only:

- Python $\geq$ 2.7.3

This software is designed to consist of two parts: The *Normalizer* transforms the raw data, which was collected during the experiments on every single agent into a single CSV file of common, normalized format. The *Analyzer* evaluates the normalized ultrasound data in comparison to the normalized infrared data.

Normalize

The *Normalizer* merges and normalizes the raw data, which every single agent collected during the experiment, into a single CSV file of common form. For example in an experiment with three agents the following raw data files can be collected: *data_0.csv*, *data_1.csv*, *data_2.csv*. These raw data files contain the information of how far a single agent measured the others to be away from it and in which sector this agent received each specific signal. The *Normalizer* reads those raw data files and groups the measurements of all agents into frames, which consist of six measurements each (one measurement per sensor). Furthermore, each frame's start time (the time of the first measurement in the frame) and end time (the time of the last measurement in the frame) are calculated and normalized to be relative to the starting time of the experiment, which is defined to be zero.

To use this tool, three arguments should be passed, the first being the host's start timestamp (e.g. the time at which the synchronization started). The second argument has to be the timestamp at which the experiment started for real (this timestamp can be retrieved from the host software as well). The third argument defines the number of agents, participating in the experiment and the last argument describes the path to the raw data files. So to normalize experimental data from an experiment this command has to be executed: *./normalizer.py <sync_start> <real_start> <num_agents> <path_to/raw_data/folder/>*

The output of this command will be the normalized data of all agents, combined in a single CSV file of the following form:

```
start,end,dtheta,dphi,amplitude,receiver,sender
0.242, 4.741,  0.000000e+00, 5.497787e+00,  655, 0, 1
5.339, 9.838, -0.000000e+00, 2.356194e+00,  655, 1, 0
0.246, 4.745, -3.926991e-01, 2.748894e+00,  885, 2, 1
0.539, 5.039,  3.926991e-01, 5.497787e+00,  885, 1, 2
...
```

Analyze

The tool for analyzing the data of several experiments and comparing it to the reference data from the *ARTtrack2* system has to be fed with some metadata about the measurements, such as the number of participants, mapping between ultrasound sensor IDs and infrared beacon IDs and the starting time. This metadata is defined at the beginning of the *analyze.py* script. The tool uses the *Normalizer* to create a normalized file for each defined metadata-record (experiment) and compares this file to the corresponding data of the infrared tracking system. It loops over all defined records and calculates the *mean distance error*, its *standard deviation* and its *median*. These results can then be evaluated and used to judge on the performance and accuracy of the system. The analysis of our experiments yields the following output:

```
GLOBAL EVALUATION:
amplitude (cm),dtheta (deg),dphi (deg)
25.6972765317,34.0617975139,36.8779800703 (mean)
8.64532071483,17.9651949776,31.263806372  (stddev)
24.4021265608,29.6173328084,20.2825099259 (median)
```

Part IV.

Evaluation

5. Evaluation

5.1. Experiment

To collect the data needed for this thesis, an experiment was conducted, exploring the interaction geometry of people. The experiment was inspired by a similar experiment as described by Groh et al. [Groh et al., 2010]. That experiment was reproduced and extended with the intend to gather additional information from the ultrasound sensors and to explore the interaction geometry of smaller groups of about two to four people, whereas the aforementioned experiment explored lager groups of about ten people.

5.1.1. Participants

About 50 people participated in the experiment, most of them being students of mathematics, informatics and mechanical engineering. This is due to the fact that the experiment took place at the Garching campus of the *Technical University of Munich*, where those faculties are located. That is also why the majority of the participants were aged between 20 and 30 years. Throughout the experiment there was a ratio of about 40% women to 60% men. Focusing on the interaction geometry of smaller groups, the participants got divided into groups of two, three and four people. In the end a very homogeneous distribution of about 30% dyads, 40% triads and 30% groups of four was reached .

5.1.2. Setting

An eight camera infrared marker based tracking system by Advanced Realtime Tracking GmbH was used, consisting of the *D-Track* Software (Version 1.24.3) and eight *ARTtrack2* infrared cameras. The cameras, which were mounted fifty-fifty at the floor and the ceiling of the room, sent out and received IR signals that were reflected by little reflective balls, which were located on small plastic markers. Each such marker (about the size of a human hand) was fitted by a number of such balls in a special spatial arrangement, in order to define the axes of this marker. The used *ARTtrack2* IR tracking system is able to track the spatial location and orientation of a marker with an accuracy of $<$ 1mm and $< 1°$ [AdvancedRealtimeTracking, 2007]. By attaching one marker to every participant's right shoulder, it was possible to precisely track their movement in the area covered by the infrared cameras during the experiment. In addition to the infrared cameras, a ceiling mounted video camera was installed for visual control of the experiment. The room where the experiment took place was about $15m^2$ in size and the IR tracking system was installed in the center of the room, providing a 3m x 3m area where the participants had to execute their tasks (cf. figure 5.1). The experiments took place in a time frame of about three weeks, one group of people at a time.

5.1.3. Approach

The procedure chosen for the experiments was as follows: After a group of participants entered the room, they got a short briefing of what they are supposed to do, which actually was talking about whatever they wanted to, while being tracked by the infrared and ultrasound tracking systems. To be able to track them in a very precise manner, an infrared marker was placed on the right shoulder of every participant and they were made to a belt, to attach the ultrasound sensor devices to. The participants were then asked to have a conversation of about ten minutes length, while feeling free to move in the 3m x 3m area. Some generic questions to ask one another were provided if they did not know what to talk about.

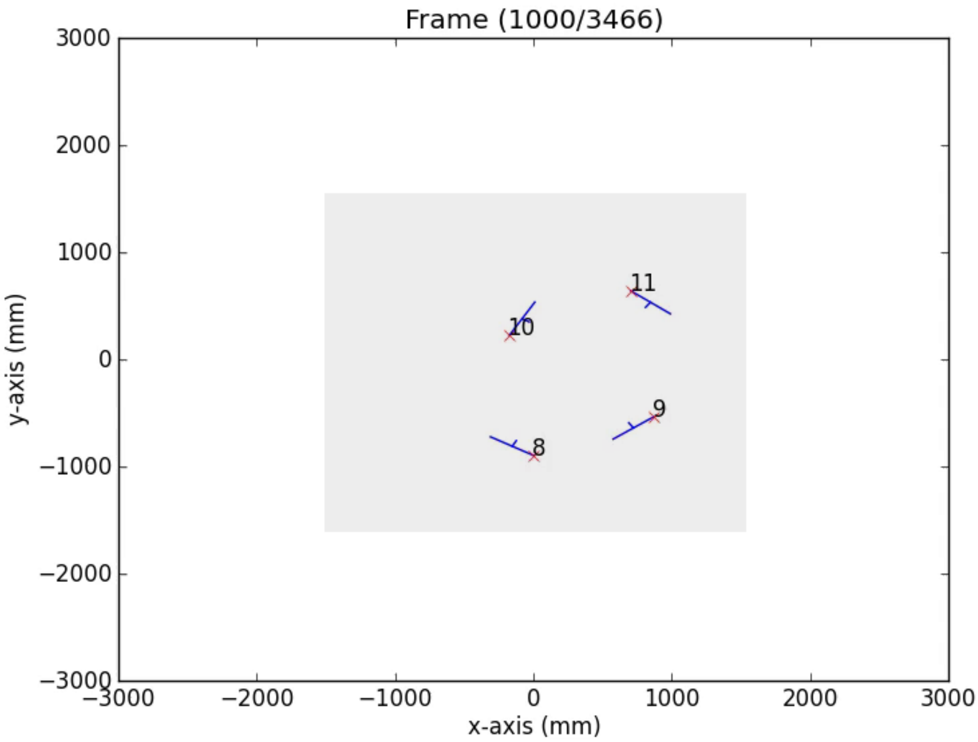

Figure 5.1.: Typical experiment scene

5.2. Comparison with ARTtrack2

This section evaluates the data, collected by the ultrasound sensors during the experiments, in a way that the final and normalized values of the relative distances between a pair of people δd, the relative body orientations $\delta\theta$ of each pair and the angles $\delta\varphi$, describing the direction at which the interaction partner is positioned get compared to the same set of reference values, measured by the infrared tracking system. The results are summarized in table 5.1.

5.2.1. Direction Angle $\delta\varphi$

To evaluate the angle $\delta\varphi$, which describes the direction in which the interaction partner is located in, it was necessary find out under which rotation angle the sensor arrays were attached to the participants, during the experiments. Roughly, they were attached to the right side of the participants hip, which implies that the tracking devices were not pointing into the peoples heading direction, but were rather rotated a little bit to the right side. As the sensor arrays were always placed just above the participants' right trouser pockets, the average angle between a person's heading and the sensor array's heading should be about 45°.

To verify this assumption, all data sets are computed with different device rotation angles starting from 0° and going up to 90°. Afterwards the results are compared, to find out at which device rotation angle the *Direction Angle* $\delta\varphi$ shows the minimal mean error. To no surprise the device rotation yielding the best results is at 47°, where $\delta\varphi$ shows a mean error of about 37° and a standard deviation of about 31° (see figure 5.2).

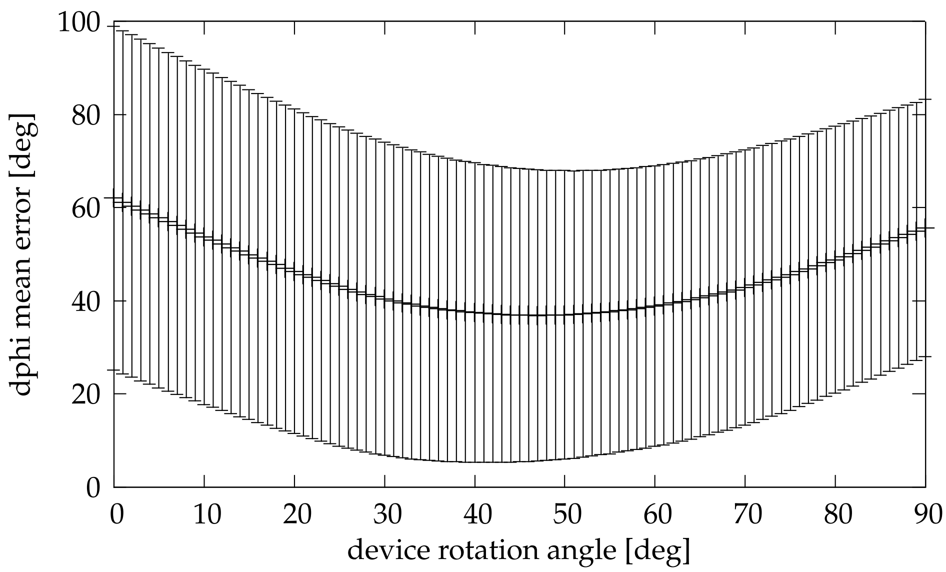

Figure 5.2.: Agent Rotation

Knowing the best agent rotation angle, it is used as a base for further evaluations. As the aim is to compare the ultrasound system's performance to the infrared system's performance, the mean error and standard deviation of all data sets are computed, to compare the ultrasonic tracking system's $\delta\varphi$ values to the corresponding values of the *ARTtrack2* infrared tracking system, using the *analyize.py* tool and assuming an agent rotation angle of 47°. The results of those computations show that the ultrasonic tracking system is accurate up to a *mean error of 37°* at a *standard deviation of 31°* and a *median of 20°*, when determining the direction angle $\delta\varphi$ at which the interaction parter is located.

5.2.2. Relative Body Orientation $\delta\theta$

The evaluation of the relative body orientation $\delta\theta$ is needed, as it is, together with the relative distance between two interaction partners, the most important parameter for determining if a social interaction is currently happening. Unfortunately the ultrasound tracking system is not able to capture this parameter directly, but it has to be calculated from the $\delta\varphi$-values of a pair of records.

This calculation is done in the normalization stage, where the system has two corresponding data records available, e.g. one of agent A listening to agent B and the other from agent B listening to agent A, with just a very small time delay in between. The system is designed to recognize and assign an incoming signal to one of nine sectors on a semicircle around the sensor array (numbered from left to right: 0, 1, ..., 8), depending on the sensor/sensors the signal is first recognized on. At the normalization stage the system knows in which sector s_A agent A received agent B's signal and vice versa (s_B) and due to the fact that both devices are attached to a person at a fixed and approximately know position, it is possible to calculate the persons' *Relative Body Orientation* $\delta\theta$, which is between $-\pi$ and $+\pi$, as follows:

$$sgn(x) := \begin{cases} +1 & \text{für } x \geq 0 \\ -1 & \text{für } x < 0 \end{cases}$$

$$\delta\theta_A = -sgn(s_A - s_B) \cdot (1 - \frac{|s_A - s_B|}{8}) \cdot \pi$$

$$\delta\theta_B = -sgn(s_B - s_A) \cdot (1 - \frac{|s_A - s_B|}{8}) \cdot \pi$$

Having the ultrasound system's $\delta\theta$ values calculated in the normalization stage, it is possible to compare them to ARTtrack2's corresponding values. The results show that the ultrasonic tracking system is accurate up to a *mean error of 34°* at a *standard deviation of 18°* and a *median of 30°*, when determining the angle $\delta\theta$, which represents the relative body orientation between two interaction partners.

5.2.3. Relative Distance δd

The relative distance δd between two interaction partners is the most important parameter in determining interaction geometry. The distance between two people is measured, using the time of flight an ultrasonic signal needs, to reach a receiver. This time of flight can be transformed into a distance, using the speed of sound in air (in $\frac{m}{s}$), which can be approximated as follows [Kim and Kim, 2011], T_c being the current temperature in °C.

$$v_{sound} = 20.05 \cdot \sqrt{T_c + 273.16}$$

Due to the missing ability to precisely measure the temperature on the mobile tracking devices a speed of sound of $343\frac{m}{s}$ is assumed for this thesis' experiments, which corresponds to a temperature T_c of about 20°C. With the described technique it is possible for a receiving agent to calculate its distance to the currently sending agent. And due to the implemented *Token Ring* (cf. chapter 4.3.1), there is always only one sending agent at a time and all listeners are aware of its ID. This makes is possible for all listening agents to calculate their pairwise distances to the currently sending agent in parallel.

After an agent has collected distance measurements to a specific sender for each of its six sensors, it calculates the relative distance to this other agent by calculating the mean distance of all six measurements, ignoring the minimum and the maximum values. To determine the ultrasonic tracking system's distance measurement accuracy, two different techniques in comparing the ultrasound data to *ARTtrack2*'s corresponding reference data are evaluated. First, the distance measured with the ultrasound sensors is compared to the average distance the infrared system measured in the few seconds time frame between measuring on sensor 0 and measuring on sensor 5. This evaluation of δd leads to a *mean error of 25.63 cm* at a *standard deviation of 8.3 cm*. Using this technique all samples captured by the infrared system in this few seconds time frame get blurred and depending on how much the person carrying the agent moved in this time frame the result might be very vague. That is why another technique is evaluated, where the distance measured by the ultrasound system is compared to the median sample of the infrared system's reference data in that specific time frame. Surprisingly, the second evaluation method has only a very minor impact on the evaluated distance values, as the mean error with this method is at *25.7 cm* at a *standard deviation of 8.65 cm* and a *median of 24.4 cm*. The mean error value of $\delta\theta$ improved by about 4°, though. In the end the ultrasound based tracking system is able to determine the relative distance δd between two interaction partners with an accuracy of about 26 cm.

Table 5.1.: Accuracy of the ultrasonic tracking system

	Mean Error	Standard Deviation	Mean Value
Direction Angle $\delta\varphi$	37°	31°	20°
Relative Body Orientation $\delta\theta$	34°	18°	30°
Relative Distance δd	26 cm	9 cm	24 cm

5.3. Detecting Social Situations

In order to evaluate the social situation detection accuracy, a *Gaussian Mixture Model* (GMM) is applied, considering the data retrieved from the analysis stage. This is done by layering Gaussian noise with the previously analyzed parameters onto the baseline data from the infrared tracking system and checking the result with a GMM analysis tool by A. Lehmann, using a 10-fold stratified cross-validation for classifiers with 5, 10, 25 and 50 gaussians.

The data is evaluated in two different ways: First, the mean error is defined to be zero, considering only the standard deviation, as the mean error could be seen as a systematic error, which could be avoided, e.g. by a perfect synchronization. This way the maximum theoretical accuracy in determining interaction geometry can be computed. As table 5.2 shows, the theoretical maximum accuracy of 79.32% is reached, when considering $\delta\theta$ and δd, using a classifier with 25 gaussians. An even higher accuracy can be reached if only δd is being considered, but this would ignore the body orientation, which is not desirable.

Table 5.2.: Accuracy in detecting social situations (zero mean)

Number of Gaussians	$\delta\theta$ & $\delta\varphi$ & δd	$\delta\theta$ & δd	$\delta\varphi$ & δd	δd	Baseline (IR)
5	71.42%	79.19%	72.05%	80.03%	80.24%
10	70.60%	78.68%	71.51%	79.95%	80.17%
25	71.90%	79.32%	72.82%	80.64%	81.03%
50	71.40%	79.09%	72.31%	81.08%	82.06%

In the second evaluation method the systematic mean error was considered, too. Table 5.3 shows that the ultrasonic tracking system is able to achieve a practical social situation detection accuracy of 68.87%, when $\delta\varphi$ and δd are considered, using a classifier with 25 gaussians. Here again, the computed accuracy is better if only δd is being considered, which would ignore the body orientation, though.

Table 5.3.: Accuracy in detecting social situations

Number of Gaussians	$\delta\theta$ & $\delta\varphi$ & δd	$\delta\theta$ & δd	$\delta\varphi$ & δd	δd	Baseline (IR)
5	53.64%	53.09%	67.33%	77.69%	80.24%
10	53.87%	53.63%	67.91%	77.92%	80.17%
25	50.79%	52.34%	68.87%	78.60%	81.03%
50	47.45%	47.13%	67.37%	77.49%	82.06%

5.4. Limitations

The following paragraphs discuss some limitations the current system has. Those limitations exist both due to the system's current implementation as well as due to some experimental constraints, which were not to avoid.

5.4.1. Experimental Limitations

The first limiting factor of the experiments is, that they took place in a rather small room and the area where the participants could have their interaction in was even smaller: It was a 3m x 3m area in the center of the room. This constraint was needed, as the infrared tracking system, which was necessary as a reference, was installed in a fixed manner, to only capture this area. It might be possible that some people have felt cramped and would have behaved differently in a real social situation.

Another constraint is related to the cables, which had to be connected from a host computer to every ultrasonic tracking device and in such to every participating person, during the experiments. The cables were installed in an as unobtrusive way as possible, attached to the backside of the peoples' belts. But still, the participants had to take care not to step onto their or the other participants cables, which limited them in their mobility and might have led to distortions.

The last experimental constraint is that due to the small groups of two to four people social situations were basically happening all the time between them. In larger groups the participants might have switched their interaction partners from time to time.

5.4.2. Varying Attachment Angles

The angles in which the ultrasonic sensor arrays were attached to the participants' belts were not stable during all experiments. This is due to the peoples' different body shape and due to not having a precise definition of how the sensor arrays have to be attached, except that they had to be attached to the right side of the participants' hips, with the USB cable pointing downward. This fact led to the situation that every tracking device measured a little different $\delta\varphi$ values and thus a the $\delta\varphi$ result is distorted, as it is a mean value of those measurements.

5.4.3. Scalability of the System

In the experiments the *ultrasonic tracking system* was used in groups of two to four people. As the time delay for each sensor is 300 milliseconds, a tracking device needs 3.3 seconds (in a dyad) to 6.3 seconds (in a group of four), to capture a whole frame (one measurement on each sensor). The 6.3 seconds in a group of four make it hard to capture dynamic situations with a lot of movement, as the measurements are spread across a relatively long time period. This leads to different measurements on each sensor and thus an increased mean error. It can improved by the approach described in chapter 7.2. That way the delay for each sensor could be reduced to 65 milliseconds, which leads to a frame length of 1.365 seconds in a group of four and thus provides a much better accuracy, especially in dynamic situations.

The same applies for tracking in larger groups. In its current implementation the tracking system is not meant to be used in groups larger than four people. If the sensors' delay time would be reduced to 65 milliseconds, though, it could be used in lager groups of up to 15 people as well, while still staying below a frame length of under five seconds.

Part V.

Closure

6. Conclusion

At the end of this thesis a conclusion on the performance and accuracy the ultrasound based tracking system is able to provide in determining the relative distance δd, the relative body orientation $\delta\theta$ between two interaction partners and the direction angle $\delta\varphi$ to a counterpart is given.

6.1. Performance in Determining Interaction Geometry

To sum up the performance properties of the ultrasonic tracking system, it can be said that the system provides a rather good accuracy in determining the relative distance δd between two interaction partners, with an average error of just about 26 cm, when compared to the very precise infrared reference tracking system *ARTtrack2*. The angles $\delta\theta$ (relative body orientation) and $\delta\varphi$ (direction), needed to determine interaction geometry, cannot be captured in a very high precision, though.

The relative body orientation $\delta\theta$ can be determined with an average error of about 34°, whereas the direction angle $\delta\varphi$ shows an average error of about 37°, which is most probably due to the fact that the sensor arrays are positioned only roughly at the same position and with only roughly the same rotation to the people carrying them. This situation could probably be improved, if the data would be combined with sensors like a digital compass, an accelerometer and a gyroscope, built into today's smartphones.

Despite the synchronization problems at the beginning (cf. chapter 4.2.1), it was possible to present and evaluate a prototype of a mobile and decentralized tracking system, based on ultrasound sensors, which can be used to gather interaction geometry data for determining social situations.

For the experiments of this thesis the data of the ultrasonic tracking system was checked with a GMM analysis tool, using a 10-fold stratified cross-validation. The results of this evaluation showed that the system is able to recognize social situations with an overall accuracy of 68.87%.

6.2. Application of the System

For a real world application, the system is meant to be integrated into tomorrow's smartphones, as "Mobile phones are now available to the majority of people [...], making them the fastest technology adoption in mankind's history. And the potential functionality of this ubiquitous infrastructure of mobile devices is dramatically increasing" [Eagle, 2005]. Up to now it is not possible to use the proposed ultrasound tracking system on a larger scale, as self built ultrasonic sensor arrays are needed. But once ultrasound transceiver get integrated into modern mobile phones, people are able to use the system on a day to day basis. And in fact big smartphone component manufacturers like Qualcomm are already heading into the direction of including ultrasound sensors [Lomas, 2013].

Once the hardware is available to a bigger amount of people, the system can be used in a wide range of use cases. It is designed to be infrastructure less and decentralized, so it is possible to be used wherever the user goes. It is usable in indoor scenarios as well as in outdoor scenarios. Once the captured data gets combined with usual smartphone sensor technologies, the accuracy can be improved and the data can be shared via a data channel like an UMTS/LTE connection between the interaction partners and a *Gaussian Mixture Model* (GMM) can be applied, to determine their interaction geometry [Groh et al., 2010].

7. Future Work

7.1. Real Time Linux

As described in chapter 4.1.1 the clocks of the different tracking devices are drifting away from each other if they are run independently, making the system loose its synchronization. If a real time system, e.g. *RT Linux*, would be used, providing hard real time constraints, it should be possible to keep the system synchronized, even though every tracking device would be running independently from one another. Once the system could stay synchronized, it would be possible to detach the cables to the host, as no external clock is needed any more. To have a common starting trigger, the devices could use the ultrasonic synchronization as described in chapter 4.2.1. This way it would be possible to have a completely infrastructure less ultrasonic tracking system.

7.2. Hardwired Ultrasound Sensors

Another drawback of the current implementation is the USB-I2C adapter, which connects six ultrasound sensors to one smartphone via an USB-OTG port. This I2C adapter turned out the be a bottleneck, as it first receives a command via a serial line and sends it out on its internal I2C bus afterwards. If something goes wrong, the adapter times out after 500 milliseconds, which is a rather long time period and thus brings a huge overhead. The current implementation uses a 300 milliseconds time frame per sensor, before the token is passed on. As this time frame does not allow to wait for the adapter's timeout, an advanced error checking was needed. To improve this situation the ultrasound sensors could be directly connected to the smartphone's I2C bus, which would enable the sensors to scan at their full speed of 65 milliseconds per sensor [RobotElectronics, a]. Being able to scan at a faster speed rate would be especially useful to track dynamic situations and larger groups.

7.3. Improving System Performance

Another worthwhile topic for further research would be the improvement of the system's overall performance. One part which could be reconsidered, is the token ring's static sequence of measurements. It could improve the overall accuracy of the system, if a randomized sequence of measurements would be used, as systematic errors (e.g. echos captured after a specific time delay) could be avoided. A simple protocol which would probably be a good starting point is *ALOHA*, where one sensor listens to the medium and sends out its burst once it cannot receive signals from any other sensor. After sending out a signal, the sensor would wait for a random time interval and start over.

In order to improve the angle measurement accuracy, it would be possible to consider the time delay between a signal received on one sensor of an agent and the same signal received on another sensor of that very agent. Using this method it could be possible not to just determine an angle by matching it to the sector where the signal was first received, but to interpolate an angle between two or more sectors, depending on the delay values. Of course it would also be a good idea to use further sensor technologies, provided by modern smartphones.

7.4. Transmitting Ultrasonic Waves Through Clothes

In the experiments the tracking devices were attached on top of the participants' clothes, which allowed the ultrasound transceivers to broadcast directly into the space around them. In a real world scenario, where the system is integrated into a smartphone, the sensors would probably be covered by clothes, e.g. because a user has put his mobile phone into his pocket. Thus it is necessary to find out the impact of different clothes on the behavior of ultrasonic waves. This knowledge can then be used to build improved ultrasound transceivers which are able to treat such circumstances appropriately.

7.5. Decentralized Synchronization of Mobile Devices

A very important topic for the ultrasound tracking system is to research methods for better synchronization between the agents. On one hand a working synchronization would improve the distance measurement accuracy by about 6 cm (cf. chapter 4.2.3), on the other hand it would make the system really usable in real world situations. As mentioned before a concrete solution to this problem could be the usage of a real time kernel like *RT Linux*, but consumer devices like smartphones usually do not have a real time kernel installed. A feature modern smartphones usually provide, though, is a multicore processor. With this feature available, it could be possible to configure the operating system's scheduler in a way that it does not touch one of the available cores, so that this core can be dedicated to keep the system synchronized. In Linux this could be achieved, using *preempt_disable()* and *raw_local_irq_save(flags)*, as described by Stiller [Stiller, 2013].

7.6. Considering Device and Sensor Positions

In the experiments self built sensor arrays were used, which were externally connected to a smartphone. In a real world situation the ultrasound sensors are meant to be integrated into the smartphone's housing, though. It would be interesting to research how many ultrasound sensors are really needed to determine interaction geometry and what the best spots to place them in a smartphone are. In this topic it should be considered how and where a smartphone is placed, which can be determined, using the build in sensors like accelerometer and gyroscope [Shi et al., 2011]. Furthermore, it would be possible to approximate the distance from the smartphone/sensors to the user's body central this way, which could then be considered in determining the distance between two interaction partners.

References

References

[AdvancedRealtimeTracking, 2007] AdvancedRealtimeTracking (2007). *ARTtrack & DTrack User-Manual v1.24.3*. Germany.

[Argyle and Schmidt, 1989] Argyle, M. and Schmidt, C. (1989). *Körpersprache & Kommunikation*. Junfermann.

[Banerjee et al., 2010] Banerjee, N., Agarwal, S., Bahl, P., Chandra, R., Wolman, A., and Corner, M. (2010). Virtual compass: relative positioning to sense mobile social interactions. In *Pervasive computing*, pages 1–21. Springer.

[BoschSensortec, 2010] BoschSensortec (2010). *BMA180 Digital, triaxial acceleration sensor – Data sheet v2.5*. Germany.

[Eagle, 2005] Eagle, N. N. (2005). *Machine perception and learning of complex social systems*. PhD thesis, Massachusetts Institute of Technology.

[Gleixner and Molnar, 2008] Gleixner, T. and Molnar, I. (2008). *hrtimers - subsystem for high-resolution kernel timers*. https://www.kernel.org/doc/Documentation/timers/hrtimers.txt.

[GoldenDeliciousComputers, 2011] GoldenDeliciousComputers (2011). *Openmoko Phoenux (GTA04) – System Manual v3.1*. Germany.

[Groh et al., 2011a] Groh, G., Fuchs, C., and Lehmann, A. (2011a). Combining evidence for social situation detection. In *Privacy, security, risk and trust (passat), 2011 ieee third international conference on and 2011 ieee third international conference on social computing (socialcom)*, pages 742–747. IEEE.

[Groh and Lehmann, 2011] Groh, G. and Lehmann, A. (2011). Deducing evidence for social situations from dynamic geometric interaction data. *International Journal of Social Computing and Cyber-Physical Systems*, 1(2):206–222.

[Groh et al., 2011b] Groh, G., Lehmann, A., and de Souza, M. (2011b). Mobile detection of social situations with turn-taking patterns.

[Groh et al., 2010] Groh, G., Lehmann, A., Reimers, J., Friess, M. R., and Schwarz, L. (2010). Detecting social situations from interaction geometry. In *SocialCom/PASSAT*, pages 1–8.

[Hall, 1959] Hall, E. T. (1959). *The Silent Language*. Doubleday.

[Harter et al., 2002] Harter, A., Hopper, A., Steggles, P., Ward, A., and Webster, P. (2002). The anatomy of a context-aware application. *Wireless Networks*, 8(2/3):187–197.

[Kim and Kim, 2011] Kim, S. J. and Kim, B. K. (2011). Accurate hybrid global self-localization algorithm for indoor mobile robots with two-dimensional isotropic ultrasonic receivers. *Instrumentation and Measurement, IEEE Transactions on*, 60(10):3391–3404.

[Lomas, 2013] Lomas, N. (2013). Why qualcomm wants to bring ultrasound transmitters to smartphones and tablets. http://techcrunch.com/2013/02/27/qualcomm-epos-ultrasound.

[Matic et al., 2012] Matic, A., Osmani, V., and Mayora-Ibarra, O. (2012). Analysis of social interactions through mobile phones. *MONET*, 17(6):808–819.

[Mehrabian, 1972] Mehrabian, A. (1972). *Nonverbal Communication*.

[Priyantha et al., 2000] Priyantha, N. B., Chakraborty, A., and Balakrishnan, H. (2000). The cricket location-support system. In *MOBICOM*, pages 32–43.

[Priyantha et al., 2001] Priyantha, N. B., Miu, A. K., Balakrishnan, H., and Teller, S. (2001). The cricket compass for context-aware mobile applications. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 1–14. ACM.

[RobotElectronics, a] RobotElectronics. *SRF02 Ultrasonic range finder – Technical Specification*.

[RobotElectronics, b] RobotElectronics. *USB-I2C USB to I2C Communications Module – Technical Specification*.

[Shi et al., 2011] Shi, Y., Shi, Y., and Liu, J. (2011). A rotation based method for detecting on-body positions of mobile devices. In Landay, J. A., Shi, Y., Patterson, D. J., Rogers, Y., and Xie, X., editors, *Ubicomp*, pages 559–560. ACM.

[Sommer, 1959] Sommer, R. (1959). *Personal space*. Prentice-Hall.

[Stiller, 2013] Stiller, A. (2013). Spezialkommando – intrinsics für präzise zeitmessung und cache-flushs. In *c't 2013, Heft 17*, pages 172–174. Heise Zeitschriften Verlag GmbH & Co. KG.

[Sundstrom and Altman, 1976] Sundstrom, E. and Altman, I. (1976). Interpersonal relationships and personal space: Research review and theoretical model. *Human Ecology*, 4(1):47–67.

[US-DepartmentOfDefense, 2008] US-DepartmentOfDefense (2008). *Global Positioning System Standard Positioning Service Performance Sstandard v4*. Washington, DC.

[Vinciarelli et al., 2009] Vinciarelli, A., Pantic, M., and Bourlard, H. (2009). Social signal processing: Survey of an emerging domain. *Image and Vision Computing*, 27(12):1743–1759.

[Willis Jr., 1966] Willis Jr., F. N. (1966). Initial speaking distance as a function of the speakers' relationship. *Psychonomic Science*, 5:221–222.