Thomas Hasler, BSc

Estimating Relative RFID-Tag Locations based on Time-Differences in Read Events

Master Thesis

to achieve the university degree of

Master of Science

Master's degree programme: Computer Science

submitted to

Graz University of Technology

Supervisor

Assoc.Prof. Dipl.-Ing. Dr.techn. Denis Helic

Institute of Interactive Systems and Data Science
Head: Univ.-Prof. Dipl.-Inf. Dr. Stefanie Lindstaedt

Murfeld, May 2018

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master‘s thesis.

Date

Signature

Abstract

Over the course of recent years, *Radio Frequency Identification* (RFID) has been used in several business domains to solve, for example, the problem of monitoring stock and tracing items along the product supply chain. However, keeping track of the exact locations of items is still an open problem. Solving this challenge is particularly desirable in logistics, retail, as well as warehouses, as it can drastically increase efficiency of several business processes. In this work, I tackle the problem of estimating relative distances between passive RFID tags. I present novel approaches which leverage information about time and signal strength of read events using off-the-shelf RFID readers and tags, without the requirement of additional hardware. I qualitatively and quantitatively evaluate my proposed approaches by inferring RFID tag coordinates and comparing them to ground truth data. Results of my experiments suggest that it is possible to derive relative locations based on time-differences in read events for several layouts of RFID tags. I see this work as a first step towards new approaches, which leverage time-differences in read events to infer relative tag distances.

Contents

List of Figures

1 Introduction

The *Radio Frequency Identification* or RFID technology is widely adopted and used in a variety of different applications in retail [Wu+09], logistics [Wei05], security [GR06], and health care [NRW06; DKA07], among others. One of the essential features of RFID systems is being able to track and identify objects without a direct line-of-sight. Ultra-High-Frequency (UHF) passive RFID systems make this possible over a range of several meters, while relying on cost-effective hardware. Especially these properties make such systems very interesting for retailers. However, as RFID read performance is influenced by several factors, such as metallic surfaces reflecting RFID signals or fluids absorbing them, tagging (i.e., attaching RFID tags to items) is not viable for all types of products.

In the fashion industry, items are often tagged during the manufacturing process, which makes it possible to track an object from its creation until it is sold, sometimes even further. However, for a store, tracking and managing items that are in stock is a very crucial process. An accurate inventory is a fundamental prerequisite for a store to perform automatic reordering of goods, make smart and quick replenishments from backroom to salesfloor, and thus, reach high on-floor availability of goods. Ultimately, a higher inventory accuracy results in more profit, hence increasing inventory accuracy is one of the primary goals of RFID solutions for fashion retailers.

1.1 Problem

Retailers already implement such RFID solutions that support many of their business processes including inventory accuracy. However, Raman, DeHoratius, and Ton [RDT01] show that besides inventory accuracy, one of the main problems of a store is misplaced items. They report that customers could not find 16% of all items because they were misplaced instead of out-of-stock. In this context misplaced means either the item is in the backroom instead of the salesfloor or just not at its dedicated place on the sales floor. There is a variety of different reasons for misplaced items such as mistakes by the store staff or customers who did not return the item to its assigned place correctly. Typically, RFID solutions recognize items on the back room that should be on the sales floor and give replenishment advice to the staff for those items. However, locating misplaced items on the salesfloor is not possible at the moment, but urgently needed.

1.2 Motivation

In this work, I infer geospatial information about RFID tags based on the information that is collected during stocktakes. Store employees perform stocktakes using a mobile RFID reader which they use to scan all tagged items in their store. For each scanned tag, the software stores the identifier, a timestamp and the received signal strength. Without additional information, exact localization of tags is not possible, but the goal of this work is to provide a solution without introducing additional hardware or infrastructure while only using already available information. As a result, instead of the exact positions of the tags, I try to obtain relative geospatial information such as the relative distances between the tags.

Knowing those relative distances between RFID tags is often enough to answer interesting questions. For example, to find a misplaced item, it is often enough to know which other items are close to it. More specifically, it is easier to find a misplaced white T-shirt if I know that a couple of blue jeans are next to it, rather than having to search the whole store. This is especially relevant for stores when they have to account for missing items after stocktake. Missing items are either stolen or gone for some other reason, or the item was not scanned during the stocktake. The RFID system that stores information about scanned RFID tags can provide information about the last known location (i.e., the last neighboring items) and therefore reduce the time to search for the item before it is removed from stock.

The search for missing items is not the only process that can be improved using the knowledge about the relative distances. For example, the system can group replenishment requests for items that are located close to each other in the back room, optimizing walking distances of store staff.

A different application would be recommendations for the positioning of items. Consider a T-shirt that is a top-seller in store A, but achieves below average sales figures in store B. Based on the relative distances, the RFID system can analyze and recommend a better positioning of the T-shirt. For example: ”In store A, this item is a top-seller when put next to items X and Y, consider a rearrangement of your items.”

1.3 Outline

This work is structured as follows. In Chapter 2 I provide a brief introduction to the *Radio Frequency Identification* technology, explaining concepts and properties that are important for the reader to be familiar with, in order to fully understand the challenges of this work. I discuss related work in Chapter 3, which mainly consists about work that has been done in the field of indoor location systems. In Chapter 4 I present the methodology I propose to infer relative tag locations, including three different approaches for estimating distances between RFID tags. In Chapter 5 I introduce the hardware I use for my experiments and outline the experimental setup in detail, before I present and discuss results in Chapter 6. I finish with a conclusion and talk about future work in Chapter 7.

2 Background

In this chapter I provide background information on the problem of automatic identification of objects in Section 2.1, which is a fundamental requirement in the retail industry. I discuss technologies that are used to tackle this problem and provide a first introduction into the technology that I use in the context of this work in Section 2.2, the Radio Frequency Identification (RFID). In Section 2.2.1 I discuss RFID tags, which represent the main determinant for categorizing RFID systems. Subsequently, I concentrate on passive RFID systems and explain how energy is provided to passive tags (Section 2.2.2). In Section 2.2.3 I outline how a reader can read multiple tags at the same time. Finally, I talk about the frequencies at which RFID systems typically operate (Section 2.2.4) and conclude with a brief description on what RFID system I use in the context of this work (Section 2.2.6).

2.1 Automatic Identification of Objects

Automatic identification (Auto-ID) of objects is vital for many industries. Typical applications include tracking of products along the supply chain or manufacturing process, access control in security systems and identification of products at the point-of-sale. There are many technologies used for Auto-ID such as optical character recognition, which is used to read identifying strings printed or written on an object or document. Other techniques include the usage of biometrics such as iris and facial recognition to identify a person, as well as the usage of cards with magnetic stripes (e.g. credit cards) for access control.

One of the most prominent examples of a technology that is used for Auto-ID are bar codes, as they are almost ubiquitous in today's commercial landscape in the form of the universal product code (UPC)[1]. An example UPC can be seen in Figure 2.1. Such a bar code encodes information in the form of parallel lines with varying width and varying spacing in between them. A scanner can read the code by traversing the pattern perpendicular to the lines.

Especially in the retail industry there is a bar code on almost every product, either printed directly on it or on a label sticking on it to identify the product at the point-of-sale.

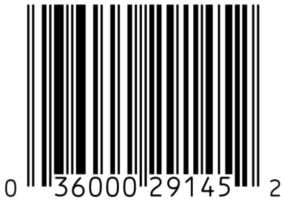


Figure 2.1: **Example of a bar code in the form of a universal product code (UPC).** The reader needs a direct *line of sight* to read the code, as it needs to traverse the pattern perpendicular to the bars.

The UPC is widely adopted because it is cheap and easy to implement, but it also has several drawbacks, such as the limited amount of information it can store. Increasing the storage can either be done by using thinner bars or by increasing the size of the bar code to fit more lines. While the former approach is limited by the readability of the code, the latter is often not possible due to limitations in space on the object. As a result, the ID space of bar codes is hardly large enough to identify single items, but they are instead used to identify the product classes (e.g, instead of a unique black T-shirt in size M, all black T-shirts of all sizes).

Bar codes cannot be extended or modified and have problems if there is dirt, ink or something else covering parts of the code. The main drawback of the UPC applies to all optical methods for Auto-ID, which is the necessity of direct visibility of the object or code that holds the identifying information (*line of sight*). However, there are many applications of Auto-ID where it is a requirement that the identification of objects can be done without a line of

[1] https://www.gs1.org/standards/barcodes/ean-upc

sight, for example in the fashion retail industry, where stores want to identify items during a stock take, to account for missing items.

For this use case, it is important to be able to identify the article even if other clothes cover it or it just is not in the expected location, because for example a customer did not return it to its dedicated place. This is just one application in which, although they are cheap and versatile, a different approach to bar codes is needed.

A modern alternative to bar codes is Radio Frequency Identification (RFID), which uses radio communication to identify physical objects. The main part of this work is to tackle the problem of geospatial localization of such RFID tags; hence in Section 2.2 I am going to explain the basic principles of this technology while outlining the motivation and highlighting the challenges of this work.

2.2 Radio Frequency Identification (RFID)

Identification via RFID technology has several advantages over bar codes and other optical identification mechanisms. However, the main benefit is that there is no need for a direct line of sight. In turn, this implicates that goods can be identified even if they are covered by other goods or are just not directly visible by the RFID reader.

This leads to a vast simplification of the daily work of the employees in retail while taking inventory, as they do not have to pick up and find the bar code for each item individually. As a result, employees can perform stock takes faster and more often, while also increasing the stock accuracy as it is less likely that the staff misses items that are in stock. Other advantages of RFID over optical solutions are that reading performance is not influenced by the lighting condition, as well as dirt or stains on the tag.

The principles of RFID technology were first used in World War II, when the British implemented a system to identify aircraft as friend or foe [Bow85]. However, until a few years ago, the technology was simply too expensive for large scale retail applications. Tagging, which refers to assigning unique RFID tags to goods of a retailer, is economically viable if the tags are very cheap. In accordance to Moores Law, the size and therefore the cost of the tags could be reduced drastically over the cause of the last years. Nowadays companies use standards introduced by GS1[2] to tag their goods and identify them using a 96-bit Electronic Product Code (EPC) [3].

In the remainder of this section I explain the RFID technology in more detail. More precisely, I am going to discuss different categorizations of RFID systems according to Want [Wan06] and highlight the designs that are of particular interest for applications in retail.

Note that an RFID system does not describe a single technology, but a combination of different technologies and hardware. The most basic RFID system consists of a *reader*, often also called *interrogator*, a *transponder* or *tag* and *antennas* used for communication. We will use the terms *reader* and *tag* for the remainder of this work to refer to RFID readers and RFID tags respectively.

2.2.1 RFID Tags

A tag consists of three parts, (i) the antenna, (ii) a semiconductor chip and (iii) a medium that encapsulates other parts such as an adhesive plastic to attach it to a label or goods. Typically these tags are only a few centimeters in size and very thin (see Figure 2.2 for examples).

RFID systems are mainly categorized by the way power is supplied to the tags. In particular, we distinguish between systems that use *active* and *passive* tags.

Active RFID Systems
In an active RFID system, tags have their own power supply or battery and transmitter to send messages to the reader. However, for retail, such a solution for identifying objects would not be feasible, as active tags are too expensive and the handling of these tags is too complex due to the active power requirements of all tags. Even if small batteries were used to power the tags, maintenance costs would be high, as someone would need to check for low batteries and exchange them.

[2] https://www.gs1.org/

[3] https://www.gs1.org/standards/epc-rfid

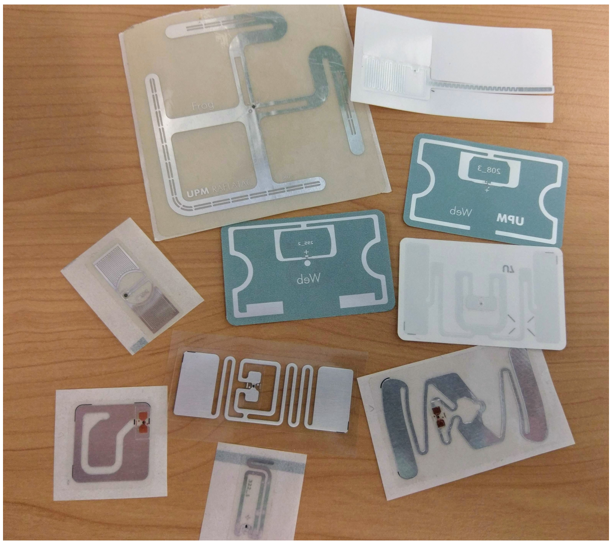


Figure 2.2: Examples of passive RFID tags.

Passive RFID Systems
Hence, to identify objects in real world retail applications, passive tags are the preferred solution. Passive tags neither have their own power supply nor a dedicated transmitter and are therefore much cheaper, easier to handle and need almost no maintenance. They power their integrated circuit and antenna through radio waves they receive from the reader. Such a transfer of energy from one medium to another is called *coupling* and the performance of this process mainly depends on the distance between tag and reader.

2.2.2 Coupling

According to Want [Wan06], there are two different categories of *coupling*. He refers to the near-field, as the electromagnetic field that is created in the vicinity of a coil when an alternating current passes through it. Some waves of this field radiate into space, creating the far-field which is the base for radio transmission. Consequently, we distinguish between near-field and far-field coupling.

Near-Field Coupling

Near-field or inductive coupling is the transfer of energy in a shared magnetic field. The energy of the magnetic field gives rise to a small current in the tag, which is enough to power its integrated circuit. The tag changes the resistance over time, for example by switching a resistor on and off, which results in small variations in voltage, which can be observed by the reader. This way the tag can encode its ID as a pattern of changes in resistance, e.g. a change in resistance corresponds to 1, while the normal resistance means 0. However, this approach exhibits physical limitations. The maximum range for near-field coupling is approximately $\frac{c}{2\pi f}$, where c is the constant speed of light and f is the frequency used to send the signal. The higher the frequency (i.e., the more data has to be transmitted), the smaller the range in which near-field coupling works. Hence, when multiple tags at the same location have to be read and distinguished, high data rates and thus high frequencies are needed. Nevertheless, in theory, one could still build a system that works over several kilometers, however, the power requirements of the reader and the size of the antennas limit the practical implementations. In practice, systems that use near-field coupling achieve a range of up to one meter, most of the times only a few centimeters. The need for high frequencies and larger distances between reader and tag led to far-field coupling, which is also used in retail RFID applications and in the experiments presented in this thesis.

Far-Field Coupling

In systems that use far-field coupling, the tags use the waves that radiate away from the electromagnetic field created by the reader to power their integrated circuit. The amount of energy that arrives at the tag is considerably smaller than when using near-field coupling, and a different solution for transporting information is needed. The antenna of tags for far-field coupling is designed to absorb most of the energy that the tag receives from the reader. However, the tag can dynamically change the antenna, so that it reflects the signal. By alternating between absorbtion and reflection of the signal over time, the tag can create a pattern that corresponds to the ID or, in the context of retail applications, the EPC of the product. The reader recognizes the reflections and can therefore identify the tag. This process is called *backscattering*. The range of a far-field system depends on the amount of energy that arrives from the reader at the tag and on the sensitivity of the reader for backscattered signals. The reflected signal that the reader receives from the tag is typically very small, as both, (i) the signal from the reader as well as (ii) the backscattered signal, are inversely proportional to the square of the distance. As a result, the returning energy is proportional to $\frac{1}{r^4}$, where r is the distance between the tag and the reader. Using far-field coupling, a range of several meters can be achieved in practice. However, there are many factors that influence read performance and the maximum range of a passive RFID system that uses far-field coupling. For example, water is a strong absorber of energy. Hence, the human body, which consists to more than 50% of water [WWB80], can shield electromagnetic waves from reaching tags. On the other hand, metals reflect energy, which can also negatively affect the read performance.

Received Signal Strength Indicator (RSSI)

Regardless of which kind of coupling an RFID system uses, the reader can determine a useful indicator of how strong the received signal is. However, the RSSI is not standardized; hence, each manufacturer of RFID hardware can have an own definition of the RSSI, regarding the unit, range, and accuracy of the RSSI. As a result, RSSI values of different manufacturers of tags and readers can not be compared to each other. Even RSSI values from different devices of the same manufacturer vary [Lui+11]. Nevertheless, if the fabricator is the same for all tags, and a single type of reader device is used, the RSSI can provide useful information for many different applications.

In the context of this thesis, the signal strength is measured in *dezibel* with reference to *one milliwatt* (*dBm*). Any power P can be converted to the corresponding value x in dBm using this formula: $x = 10 \log_{10} \frac{P}{1mW}$. Using this formula we can derive that $30dBm$ equals 1W. Passive RFID systems deal with much lower power levels, for example, the response of a passive tag is typically only a few μW or nW which yields RSSI values in the range from -80 to -30 in my experiments.

Intuitively, the signal strength is a good indicator of how far away or nearby an object is situated to the reader. In theory, the signal strength should be higher if an object is closer, and lower if an object exhibits a larger distance to the reader. However, in practice, the signal strength is affected by a variety of different factors, such as the orientation of the tag (i.e. its antennas), if the tag is covered by something, if it is surrounded by metals or other reflecting/shielding substances, and many more. Parameswaran et al. [PHU+09] even showed in an experimental study, that the RSSI is not a reliable indicator of distance, even under ideal conditions within a controlled environment.

The hypothesis of my work is that, although the RSSI is not a reliable indicator of distance, using subsequent reads of a tag, we can filter out noisy RSSI values, derive the denoised signal strength and infer the actual distance of a tag to a reader.

2.2.3 Reading Multiple Tags at the Same Location

Identifying many tags at almost the same time is a pressing issue in todays retail stores. Wether store employees want to scan all items in a shopping cart, or all articles lying on a shelf, readers have to process the answers of potentially hundreds of tags at once. This can lead to collisions of the signals of the tags, which the reader recognizes, but can not reconstruct, leading to information loss as the corresponding tags are not read. Hence, it is important to understand that a reader can only communicate with one tag at a time.

An early solution for this problem was to implement a random waiting time in the tags to avoid collisions. Although this improves the process, collisions can still happen, and probabilities increase with larger batches of tags. When a reader detects a collision, it would simply try again to read the tags until no collisions happen anymore.

Modern RFID systems that implement the standards defined by GS1[4] use special algorithms, such as the *Query Tree Protocol* [Wan06] to handle collisions. According to this protocol, the reader queries for tags whos identity start with a specific bitmask. If only one tag is found using this bitmask, it will answer with the next bit in its identity and the reader will use this bit to extend the bitmask to continue its search. If more than one tag answers, the reader detects a collision and in the next step, it has to query for all possible bitmasks. Figure 2.3 shows an example with three collocated tags. Using the Query Tree Protocol, modern

[4] https://www.gs1.org

readers can handle several hundreds of tags per second. The reader we use in our experiments can read more than 700 tags per second according to the manufacturers specification. All the equipment we use is described in Section 5.2 in more detail.

2.2.4 Frequencies

RFID systems use a wide range of frequencies starting from around 100 kHz to over 5 GHz. However, systems mostly use reserved bands for communication. There are common bands used by RFID systems in low-frequency (LF) range from 125 to 135 kHz, high-frequency (HF) at 13.56 MHz and ultra-high-frequency (UHF) at 2.4 GHz, and the reader and tags have to be selected accordingly. LF RFID systems are typically used for access control or livestock tracking. HF RFID applications include ticketing and payment applications, for example the well-known *Near Field Communication* standard is based on HF RFID. In the retail domain, higher frequencies are preferred, due to the higher data rates, allowing readers to process a larger number of tags per second. Hence most RFID systems used by retailers operate at ultra-high frequency.

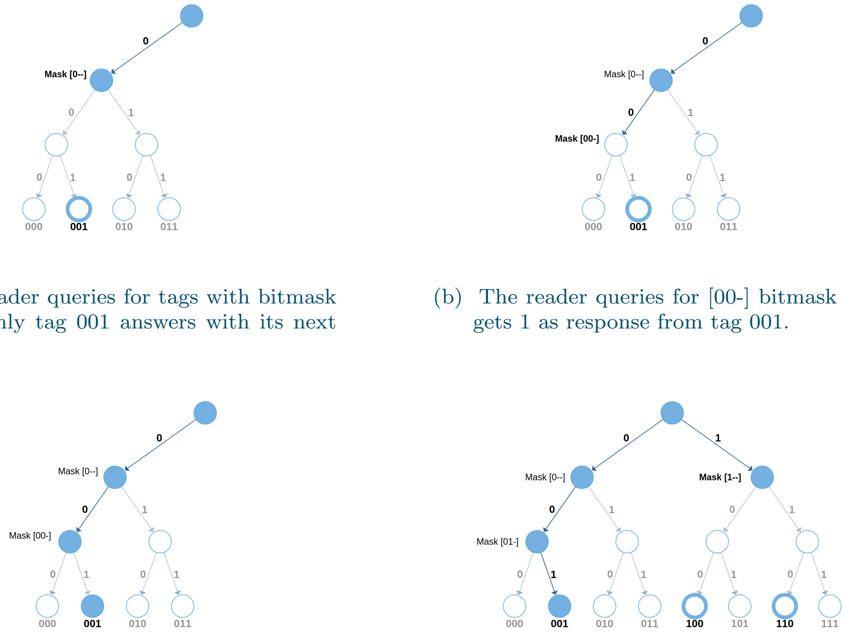
2.2.5 Sessions of EPC Gen2 Tags

The *EPC UHF Gen2 Air Interface Protocol*[5] defines a standardized behavior for tags and readers. For this work, the concept of *sessions* is important to understand, and as a prerequisite readers of this work must know about the concepts *state* and *persistence*.

An EPC Gen2 compliant tag provides an *inventoried flag* or *state* for each of the sessions, which has either the value A or B. Typical behavior for a reader is to read all tags with state A and change their state to B. The time how long a tag stays in state B before it automatically switches back to state A is called *persistence*.

The GS1 protocol defines four sessions (S0, S1, S2, S3), which differ mainly in the defined persistence times. A tag usually is in state A when it first enters the field of the reader, except it already was in the field before and its persistence

[5] https://www.gs1.org/standards/epc-rfid/uhf-air-interface-protocol/2-0-1



(a) The reader queries for tags with bitmask [0–]. Only tag 001 answers with its next bit 0.

(b) The reader queries for [00-] bitmask and gets 1 as response from tag 001.

(c) Tag 001 is identfied when the reader queries for pattern [001].

(d) When querying for bitmask [1–] the reader detects a collision, because both tag 100 and 110 answer.

Figure 2.3: Example of the Query Tree Protocol [Wan06] using three tags with IDs 001, 100, 110. From top to bottom, the reader queries for tags whos identity starts with a certain bitmask and each tag answers with the next bit in its identity. In (a) the reader queries for tags with bitmask [0–] and only tag 001 answers with the next bit in its identity. In the next step (b), the reader only queries for bitmask [00-] and can identify the tag in (c). If multiple tags answer, the reader recognizes a collision as in (d), where two tags (100 and 110) answer. Due to the collision, the reader has to query for both bitmasks, [10-] and [11-] and can identify the tags as before.

time is not over yet. In session 0, tags stay in state B for as long as they are in the field of the reader and switch back to state A when the field no longer energizes them. Sessions 2 and 3 are similar to session 0; however, these two sessions define persistence for the time after the tag leaves the field of the reader. As a result, in session 2 and 3, the tag stays in state B after the tag leaves the field, and switches back to state A only after the corresponding persistence time is over. In session 1, tags stay in state B for a defined time regardless if they are in or out of the field. Figure 2.4 shows a schematic of how sessions work.

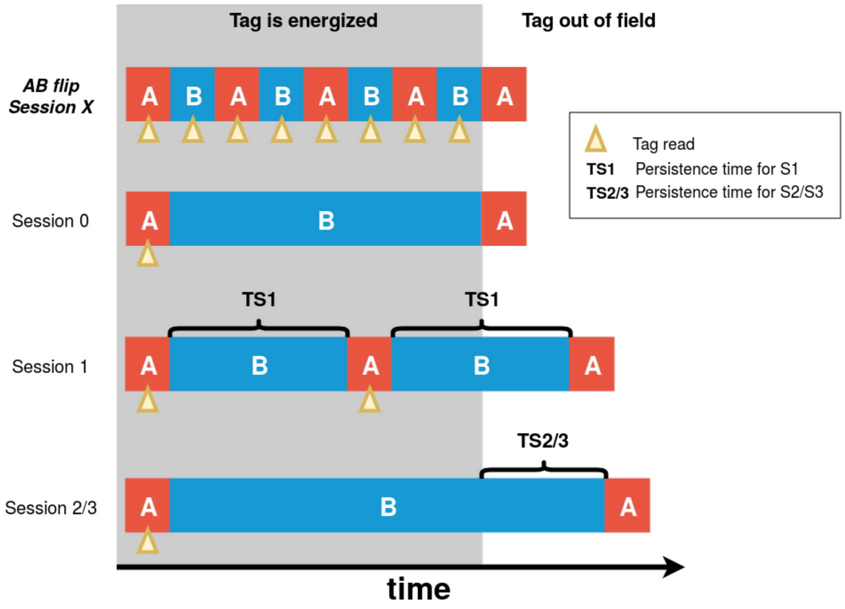


Figure 2.4: **Sessions of EPC Gen2 tags.** The red and blue boxes show the state of a tag over time, both when the tag is in the field of the reader (grey background), and when it is out of the field. In session 0, tags stay in state B for as long as they are in the field. In session 1, tags remain in state B for the time TS1. In session 2 and 3, tags remain in session B for as long as they are in the field of the reader; however, when they leave the field of the reader, they remain in state B for the corresponding times TS2 and TS3. The reader I use for my experiments implements *AB-flip* mode, which reads tags all the time, alternating the query state. Hence *AB-flip* mode generates the most data.

For each session, the figure shows how the tags change their state over time, in and out of the field of the reader. In addition to the sessions, the figure also shows how the *AB-flip* mode works that is implemented by the reader I use in my experiments (see Section 5.2 for details about the hardware). In *AB-flip* mode, the device automatically alternates queries for tags in state A and B. As a result, using this mode, the reader generates the most read-events per tag.

2.2.6 Practical Applications of RFID in Retail

There are many design decisions users of RFID systems have to make. The main deciding factor of current RFID systems is the price of the tags, due to the large upfront costs in tagging every single item of a retailer. Hence, mainly passive tags are used. To allow for fast and accurate stocktakes, the read range is also important, which is why most real world applications use far-field coupling. Finally, due to large ID spaces and the need to read many collocated tags within short periods of time, high data rates and therefore a high frequency is needed. As a result, many retail applications use Ultra-High Frequency Passive RFID systems.

Almost all applications of RFID need software to support processes such as a stocktake. The software stores information such as the target list (i.e., the list of expected items in stock), the sold items, reserved items and potentially several additional types of complementing information about the store. Using this data, RFID systems already solve many different problems for retailers.

Due to item level tagging the system can provide smart refilling advices, as it exactly knows how many items of a specific article are available on the salesfloor. For example, if there is only one instance left of a top-selling T-shirt, the system may advise the staff to refill sold items with units from backroom.

Product recommendations can also be provided by analyzing cross-selling articles, which means articles that are sold together. Due to RFID tagging the system knows which items are in stock at any time and can recommend only items that are actually available in the store.

However, at the moment there is usually no geospatial information about the items available, as existing RFID systems do not provide any features to capture this information. Furthermore, it is infeasible for the staff to provide information about the location of every single item to the system. As a result, the system is not aware of the location of the items inside a store.

In this thesis, I set out to tackle the problem of assigning geospatial information to tags based on read events of tags during stocktakes. Spatial information about the tags can drastically improve an RFID system, as not only new features can be developed, but also existing features can be improved. For example when searching for a missing item, the RFID system can recommend a location where to search for the item. Refilling advices can be improved by grouping advices based on location, thus minimizing the distance an employee has to walk. Furthermore, the location of the items in different stores can be analyzed in order to optimize product placement.

3 Related Work

A lot of work has been done in the field of localization. For outdoor applications, the Global Positioning System (GPS) is the primary system used for localization. However, for indoor localization, the GPS is not suited, as a direct line of sight between the satellites and the receiver device is needed. Therefore, I concentrate on indoor localization, as most fashion retailers are located indoors. In this chapter, I first discuss the algorithms that are used in related work to determine a location in Section 3.1, before I focus on the hardware and technology of location systems in Section 3.2.

3.1 Positioning Algorithms

Liu et al. [Liu+07] analyze three different schemes for location estimation, (i) triangulation/trilateration, (ii) scene analysis, and (iii) proximity. Regardless of the technology and hardware that is used in a location system, typically one of these three schemes is used for the estimation of the position of an object.

3.1.1 Triangulation / Trilateration

Both triangulation and trilateration are methods to position an object using three reference points. While the former uses the angles between the object and the reference points to estimate the position, the latter takes the distances between the object and the reference points into account.

The distances for the lateration approach are typically derived from measurements such as time-of-arrival (TOA, i.e. the one-way propagation time of a signal to a receiver) [Pat+03], [SHS01], or received signal strength (RSS) [AP04] and [Pat+03]. Figure 3.1 shows a schematic of how basic trilateration works, where the dashed lines represent an arbitrary distance approximation. For each of the reference points, the distance approximation creates a circle with potential locations of the tag. A basic trilateration approach would be to calculate the intersection of the circles to estimate the location of the tag. In contrast to trilateration, triangulation uses the angle-of-arrival (AOA, i.e., the angle from which the signal arrives at the receiver) instead of the distances to calculate the positions [NN03].

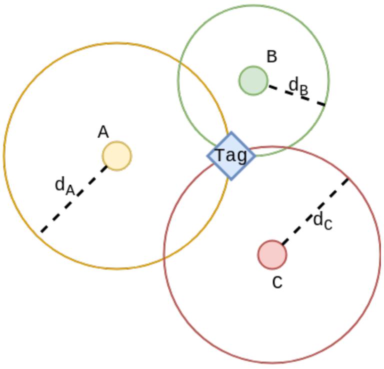
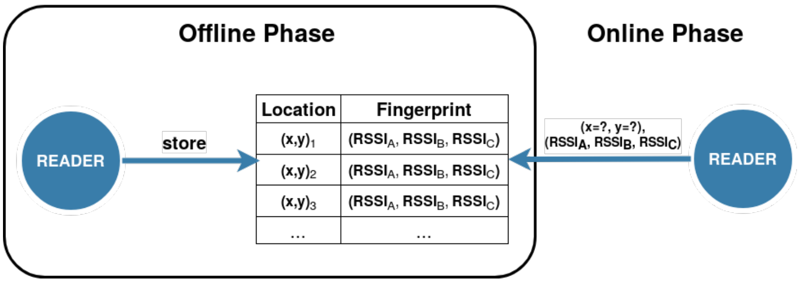


Figure 3.1: **Schematic of a basic trilateration approach.** For each read point, the corresponding distance measurement or approximation (d_A, d_B, d_C) creates a circle of possible locations for the tag. Here, the dashed lines correspond to an arbitrary distance approximation. A basic trilateration approach is to calculate the intersection of the circles to estimate the position of the tag.

For this work, I consider only one RFID reader, scanning many passive tags; thus the reader is the only active device in my setup. As a result, triangulation or trilateration methods are not applicable to my thesis.

3.1.2 Scene Analysis

The basis of scene analysis approaches is an initial *offline* or *training* phase, during which data is collected that is used to calculate *fingerprints* of locations. An example of information encapsulated in such a fingerprint is the RSSI to known reference tags or other antennas. The result of this process is a database of *location fingerprints*, where each fingerprint captures some characteristic or pattern in the signal at its corresponding location. This database can then be used in the *real-time* or *online* phase to infer the location of the user by finding the fingerprint that is the closest match. Figure 3.2 shows a schematic of how scene analysis works, given fingerprints that consist of RSSI values of reference points A, B, and C.



Location	Fingerprint
$(x,y)_1$	($RSSI_A$, $RSSI_B$, $RSSI_C$)
$(x,y)_2$	($RSSI_A$, $RSSI_B$, $RSSI_C$)
$(x,y)_3$	($RSSI_A$, $RSSI_B$, $RSSI_C$)
...	...

Figure 3.2: **Schematic of a scene analysis approach.** The reader creates a database of *location fingerprints* in the *offline* phase. In this example, a fingerprint consists of three RSSI values to reference points A,B, and C. When the reader wants to determine its position in the *online* phase, it searches for matching or close fingerprints in the database.

One of the most well-known systems that use scene analysis is an indoor pedestrian tracking system by *Microsoft* called RADAR [BP00]. In the *offline* phase, they record information about the radio signal as a function of the position of the user. During this step, fingerprints of known reference points are calculated and stored. During the *online* phase, they calculate fingerprints of the current location of the user and match them with the previously stored fingerprints to derive the position of the user.

Other methods used for scene analysis include probabilistic methods [Kon+04], which learn $p(L_i|s)$, which is the probability of location L_i given a signal s, during the learning phase. The system can then predict the location with the highest probability or use the probability to calculate a weighted average position over all locations in the second step. A different method would be to use the k-nearest-neighbor (kNN) [Ni+03] algorithm, where the system chooses the k nearest locations in signal space and then calculates the position based on them. There are also other methods used such as neural networks or support vector machines; however, the principles are the same for them.

For my thesis, a scene analysis approach is not feasible at the moment, as the data collection step goes beyond reading items in a store. The task of analyzing a scene is not feasible for the store staff that uses the same software I use in my thesis, due to the required financial and operational efforts. Moreover, at the moment, no additional hardware is a requirement under which my proposed localization approach should work.

3.1.3 Proximity

Proximity methods usually are implemented with a dense grid of antennas. If a tag is in the range of an antenna (i.e., the reader detects the tag), it is considered to be collocated to this antenna. If multiple readers detect the tag, the antenna with the highest signal strength defines the location of the tag. As a result, this localization approach provides a symbolic, relative location of the tags.

Researchers used proximity based algorithms to locate materials on construction sites with 20 reference tags arranged in a dense grid [SHC07]. However, they do localization outdoors and can therefore determine the position of the reader using GPS. Other location systems that use proximity methods are proposed in [SS02] and [He+05].

This kind of localization algorithm is not possible for my work, because I only use a single, mobile RFID reader. However, I use a form of a proximity method for my experiments, as the goal of my work is to estimate the relative distances between tags.

3.2 Indoor Location Systems

Hightower et al. [HB01] give a brief introduction into the problem of indoor localization and provide a taxonomy of location systems. Further, they define and discuss the properties of location systems such as *accuracy*, *scale* and the *cost* of such systems. According to Hightower et al. [HB01] a localization system can provide *physical* or *symbolic* location. A physical location is an exact definition of where an object is, such as the coordinates of a building. A symbolic location is an abstract description of the position, e.g., "*in the kitchen*", or "*next to X*".

Moreover, a location can be *absolute* or *relative*. Absolute information uniquely identifies a location, for example, the coordinates in a map. A relative location, on the other hand, is always relative to a reference point; hence it depends on the point of view.

The location systems mentioned by Hightower et al. [HB01] use different techniques for localization, such as infrared (*Active Badge* [Wan+92]), ultrasound (*Active Bat* [Har+02] and *Cricket* [PCB00]) and computer vision (*Easy Living* [Kru+00]). Modern location systems rely more and more on RFID technology.

3.2.1 RFID-based Location Systems

The *SpotON* system by Hightower et al. [HWB00] was one of the first location systems relying on RFID technology. However, instead of using conventional tags they use custom active tags specifically designed for localization, with the primary requirement of being able to provide an exact signal strength measurement. In addition to the custom tags, they use multiple base stations (RFID readers) distributed over space that report signal strength of detected tags to a central server, where the values are used to triangulate the position of the tag.

Alippi et al. [ACV06] propose a statistical approach that also requires multiple readers to cover the area under investigation. However, they locate off-the-shelf passive RFID tags using their system. There are several other approaches that also require multiple readers or antennas ([Hek+10], [Sav+14] and [BP09]).

The LANDMARC system by Ni et al. [Ni+03] uses a similar method, but instead of expensive readers as base stations, they work with active RFID tags as reference tags. How many reference tags to use and where to position them has to be decided based on the environment for the localization. In their experiments, they used a dense grid of reference tags. The position of an object is calculated as a weighted sum of the coordinates of the k nearest reference tags.

Saab and Nakad [SN11] try to estimate the position of a vehicle or a person in an indoor environment. More specifically, they use RFID technology and try to track an RFID reader along a path that has passive RFID tags next to it. They store the positions of these tags in a database and estimate the distances between the reader and the tags using the RSSI of the backscattered signals. Subsequently, they use the estimated distances to calculate the position of the reader via trilateration.

Joho et al. [JPB09] propose a probabilistic sensor model that can be trained using unsupervised learning. The sensor model specifies the likelihood to get a measurement $\mathbf{z}$ given the pose $\mathbf{x}$ of the antenna and the location $\mathbf{l_g}$ of tag with ID g. This probabilistic model $p(\mathbf{z}|\mathbf{x}, \mathbf{l_g})$ can be learned from data, and using Bayes' rule the posterior $p(\mathbf{l_g}|\mathbf{x}, \mathbf{z})$ can later be evaluated. However, due to the enormous amount of combinations between antenna poses and locations of the tag, this calculation is infeasible. Therefore, Joho et al. [JPB09] use locations relative to the antenna rather than absolute locations. As a result, to be able to localize tags using this model, the position of the reader has to be determined at all times. In their setup, they drive a shopping cart that has the RFID reader and a laser range scanner mounted on it. They use laser-based FastSLAM [GSB07] to localize the antenna and estimate the distances of the tags afterward.

The RFID reader I use in the context of my work is a mobile handheld unit and does not have additional hardware to determine its position. Moreover, the mobile reader allows for fast and agile movements while performing a stocktake, which would make it difficult to localize the reader even if for example laser sensors were available.

4 Methodology

This chapter focuses on the methodology that I introduce in this work, which consists of several steps. First, I describe the data preprocessing step (see Section 4.1), in which I filter and align the data before I perform the distance estimation (see Section 4.2). Then I use Multidimensional Scaling (MDS) to calculate coordinates for the tags based on the estimated distances (see Section 4.3). Finally, I run several experiments which I evaluate on manually constructed ground truth data (see Section 4.4). The steps of this methodology are highlighted in Table 4.1.

Table 4.1: **Methodology**. I use this methodology to estimate tag distances solely based on time and signal strength information of recorded tag read-events. After preprocessing of the data, I estimate distances usiing different approaches and calculate coordinates based on the approximations using Multidimensional Scaling (MDS). Finally, I evaluate the results using (i) Pearson correlation and (ii) Mean Absolute Error.

	Step	Description	Section
I.	**Preprocessing**	Filter unwanted data and shift timestamps.	4.1
II.	**Estimate Distances**	Estimate tag distances using	4.2
		• Temporal Distance	4.2.1
		• RSSI Peaks	4.2.2
		• Weighted Sum of Time and RSSI	4.2.3
III.	**Calculate Coordinates**	Determine coordinates based on estimated distances using MDS.	4.3
IV.	**Evaluation**	Calculate the error of the estimation.	4.4

4.1 Data Preprocessing

The reader receives a response from all tags that its query reaches. It is possible that during my experiments I receive and log read events from tags that are not part of my tag set. In a first preprocessing step, I remove read events that belong to these unknown tags.

As mentioned in the Section 5.3, only the timestamp and RSSI value are captured. I scale the timestamp so that the first read-event happens at timestamp 0. Each following entry corresponds to the number of milliseconds after the beginning of the experiment.

4.2 Estimating Relative Tag Distances

For the estimation of tag distances, I evaluate several different approaches, which vary in the amount of information I extract from raw data and how I process it to estimate distances. In the remainder of this section I first describe an approach to approximate tag distances based on time-differences in read events of tags (see Section 4.2.1). Afterwards, I show methods that try to estimate the distances by leveraging the relationship between the timestamps and signal strengths of the read events of the tags in sections 4.2.2 and 4.2.3.

4.2.1 Temporal Distance

The basic idea of this approach is that tags that are scanned together or within a short period of time, are also located close to each other. Whereas if there is more time between read events of two tags, there should also be more distance between them.

For example, Figure 4.1 shows read events of several tags over time. I can easily see that tag 1 (blue) and tag 2 (orange) have read events almost at the same time; hence I would estimate a very small distance between them. Time-differences between tag 2 (orange) and tag 3 (green) on the other hand are larger, therefore the distance estimation would also be larger.

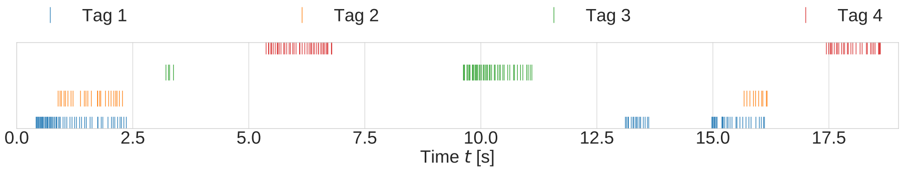


Figure 4.1: **Read events over time**. The figure shows read events of different tags that happen during one of my experiments. Each bar represents a single read event and time-differences between tags clearly vary, for example, read events of tag 1 (blue) and tag 2 (orange) happen almost at the same time, while there is more time between read events of tag 2 (red) and tag 3 (green).

In my first approach, I want to leverage the information in time-differences of read events to estimate distances between tags. Distance is a function of speed and time and if the former is constant, there is a direct relationship where distance equals time scaled by the constant speed.

More formally, assuming speed v is constant, it can be written as $v = s/t$, where s denotes the distance and t the duration; hence, $s \propto t$. As a result, time-differences of read events of tags are proportional to the distance of the tags to the reader, and therefore also between the tags themselves.

Figure 4.1 highlights that for each tag I get a vast number of read-events. For the distance estimation of tags I consider the minimum difference between read events; hence, my distance approximation $\hat{d}$ can be written as follows:

$$\hat{d}_{ab} = min(|t_{a_n} - t_{b_m}|) \quad \forall n \in N_a, \forall m \in N_b$$

where N_x is the number of read events for tag x, and t_{x_i} corresponds to the timestamp of its i-th read event.

This approach ignores that changes in direction have a great impact on the actual distance between two locations. For example if I walk in a circle, I travel a certain amount of time and distance, but end up at the same position as before. To address this limitation, I use the minimum time-difference between read events and assume that read events at the beginning and the end of the circle overlap and hence yield realistic distance approximations.

In a simplified setting, where I read one tag at a time and move with constant speed in a single direction, this approach can recover the distance up to a scaling factor. However, in reality, I read multiple tags at almost the same time and get responses from tags with varying distance and angle to the reader at any time. Hence, I hypothesize that using only the time information in the experiments, should not work very well for tests that go beyond the outlined simplified settings.

4.2.2 RSSI Peaks

For this approach, I add information about the received signal strength (RSSI) to the time-differences. Although I can not derive distances directly from RSSI as it is influenced by a variety of factors such as surrounding objects or the orientation of the tag (see Chapter 2 for a detailed explanation), there is still information encoded in the signal strength information.

Figure 4.2 shows the RSSI of several tags over time, as they enter and later leave the field of the reader. In this example all tags are scanned at least twice. The RSSI value starts low as the tag gets in the focus of the reader and increases until it reaches a peak; and it decreases again until it leaves the focus. At the peak, the reader points exactly in the direction of the tag.

Note that the range, and therefore also the peak values of the RSSI differ for each tag. Several factors can influence the maximum values of these peaks such as the distance to the reader, the orientation of the tag as well as surrounding tags or objects to name but a few. Hence, I do not use the RSSI value itself, but rather use the peak values to identify the moment in time when the tag was best readable for the reader.

To obtain these RSSI peaks, I partition the sequence of read events into segments that correspond to a tag entering and leaving the field of the reader. In other words, whenever a tag enters the field of the reader, I want to create a segment containing all the read events of the tag until it leaves the field.

Typically, read events belonging to the same segment happen in a very short period of time, while the gap between read events of different segments is higher. To find the right threshold, I inspect the distribution of time-differences between consecutive read events of a tag in Figure 4.3.

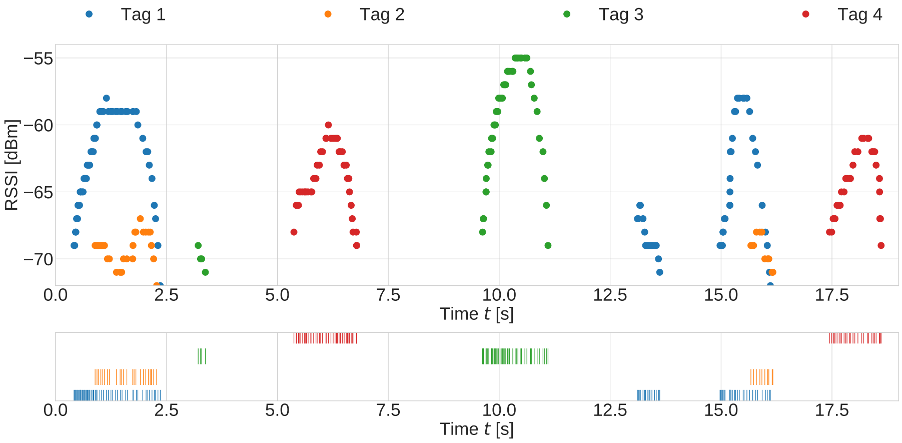


Figure 4.2: **The RSSI over time.** The figure depicts the RSSI over time during one of my experiments. Each dot represents a single read event and the bell-shaped segments model the RSSI of a tag entering and leaving the focus of the reader. Each peak corresponds to the moment when the reader points in the direction of the tag.

The distribution shows that the time-difference between read events is small for the majority of the read events (i.e., below 1,500 milliseconds). By setting the threshold to 1,000 milliseconds, about 5% of the read events generate new segments.

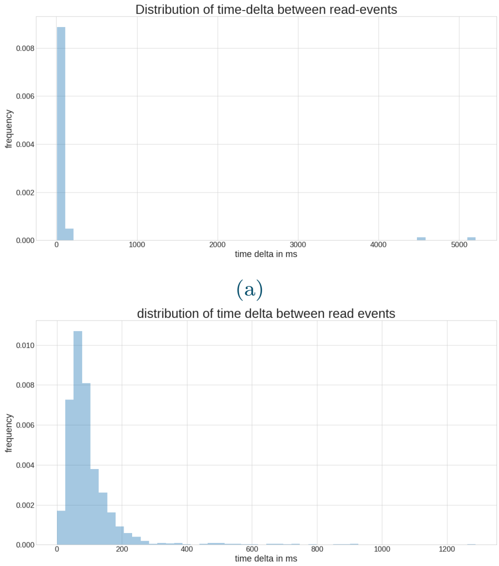
For each of the segments, I calculate the mean timestamp of the read events with the highest RSSI. More formally, let E_i be the set of read events of the i-th segment, and $P_i \subseteq E_i$ be the subset of read events with the highest RSSI in segment i. Then I define the time $\bar{t}_i$ of the RSSI peak of the i-th segment as the mean of the elapsed milliseconds:

$$\bar{t}_i = mean\{t(p_1), t(p_2), ..., t(p_n)\} \quad \forall p_i \in P_i \tag{4.1}$$

where $t(p_i)$ denotes the timestamp of read-event p_i.

With this I define the distance estimation between tags a and b for the RSSI peaks approach as follows:

$$\hat{d}_{ab} = min(|\bar{t}_{a_n} - \bar{t}_{b_m}|) \quad \forall n \in S_a, \forall m \in S_b$$



(a)

(b)

Figure 4.3: **The distribution of time differences between two successive reads of the same RFID tag.** (a) Shows the histogram for all read events, while (b) highlights the distribution of time differences smaller than 1500 ms.

where S_x is the number of segments for tag x, and t_{x_i} corresponds to the mean timestamp of the RSSI peaks in the i-th segment as shown in Equation 4.1.

4.2.3 Weighted Sum of Time and RSSI

In the first two approaches (see Section 4.2.1) I used only time-differences to estimate distances, while in the second approach (see Section 4.2.2) I utilized RSSI values to define the moment in time of the RSSI peaks, however, I did not use the RSSI value itself for the estimation.

For my third approach I now use both, time-differences and differences in signal strength of read events. Therefore I calculate a weighted sum of time and RSSI of read events as follows. Let $w_{x,y}$ be the weighted sum between two read events x and y as follows:

$$w_{x,y} = |t_x - t_y|\alpha + |r_x - r_y|\beta + R\lambda \tag{4.2}$$

Where α is the weight for the time-difference, β the weight for the difference in RSSI, and $R = \frac{2}{r_x + r_y}$ is a regularization factor to penalize small RSSI values.

I calculate pairwise distances of the tags by comparing all read events and use the minimum distance between two read events as distance approximation. More formally, I calculate the distance estimation $\hat{d}_{ab}$ between tags a and b in this approach as follows:

$$\hat{d}_{ab} = min(w_{x,y}), \forall x \in N_a, \forall y \in N_b$$

where N_a and N_b are the read events for tags a and b, and $w_{x,y}$ is the weighted sum of read events x and y.

The calculation using Equation 4.2 is a computationally intensive task, as for n read events I need to compute n^2 values. Furthermore, to find the distance between two tags I need to compute all combinations of their read events to find the minimum distance. Hence, to reduce complexity I propose to use a RSSI-threshold t_{RSSI} that defines which read events to use for the estimation. For each tag, I perform MinMax scaling on the RSSI value, and filter out read events with RSSI values lower than t_{RSSI}.

With this I not only limit the computational complexity, but also filter out measurements with low RSSI values that are likely to be reflected signals from tags that are far away from the reader.

I performed a grid search for the hyperparameters α, β, λ and t_{RSSI} and found that the configuration depicted in Table 4.2 performs best.

4.3 Calculating Coordinates

The result of the estimation step described in Section 4.2 is a distance matrix with the approximated relative distances of the tags. I use Multidimensional Scaling (MDS) to find a representation in 2D, that matches my estimated distances. In other words, using the MDS algorithm I calculate coordinates for

Table 4.2: **Hyperparameters for the Weighted Sum of Time and RSSI.** I performed a grid-search over the range from 0 to 1 in increments of 0.1 for all parameters. The threshold $t_{RSSI} = 0.2$ means that RSSI values lower than 20% of the maximum value are ommitted. The values indicate that RSSI values are not that important, as the weight for the RSSI β, as well as the regularization factor λ for small RSSI values are small (0.1), compared to the weight α for the time-difference.

α	β	λ	t_{RSSI}
0.9	0.1	0.1	0.2

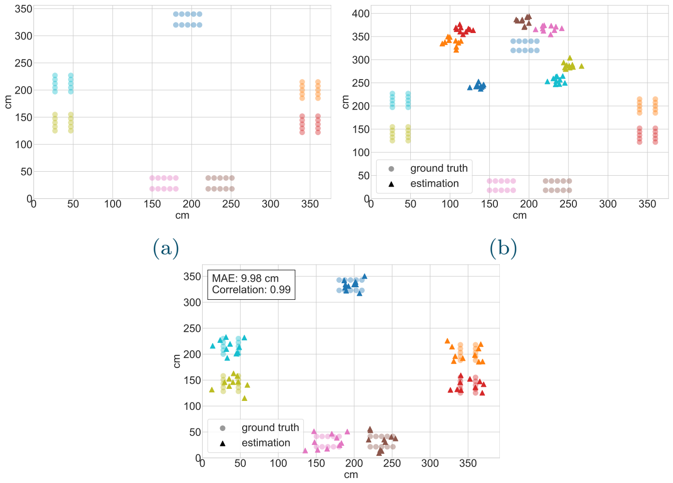
tags, so that the distances between the tags match the given distances as close as possible.

However, the result of MDS is invariant to scale, translation, rotation and reflection. Figure 4.4b shows one possible solution provided by MDS. In this example, coordinates are scaled, translated and rotated. Hence, as a preprocessing step for the evaluation, I have to find the transformations that map the estimation to the ground truth, which is called the Procrustes problem. In Section 4.3.2 I discusses the methods I use for solving it, to finally arrive at a solution that could look like in the example in Figure 4.4c.

4.3.1 Multidimensional Scaling

Multidimensional Scaling (MDS) is a method for visualizing data based on some similarity or dissimilarity measurements called *proximities*. Given the proximities of n points, MDS finds a geometrical representation of these points in a way that the pairwise distances match the dissimilarity measurements as close as possible.

MDS requires that the proximity data is provided, which can be any score, rating or other measures of similarity. In my case, I use the estimated relative distances as my proximities, and I seek a geometrical representation in 2D space. There are many variants of MDS, which differ based on the provided proximities. In this work, the dissimilarity values have metric properties and satisfy the triangle-inequation, hence, I can use basic *metric MDS*.



(a)

(b)

(c)

Figure 4.4: **Methodology.** The figures depict different steps of my methodology. First, I generate a ground truth dataset for each setup, (Figure 4.4a). Using MDS on the estimated distances, I obtain results invariant to scale, translation, reflection and rotation (Figure 4.4b). After applying Procrustes analysis, the final estimated coordinates are shown in Figure 4.4c.

To get an intuition of how MDS works, I provide an example. When looking at a map, it is easy to derive the distances between cities. I can use a ruler and measure the distances (in a specific scale), or I can use the known coordinates of the cities and calculate them. However, the inverse problem is not as easy to solve. Given just the distances, I want to find the coordinates of the cities.

Table 4.3 shows a distance matrix of European cities, which I obtained by manual measurements in Google Maps. Therefore, the measurements are admittedly not very precise, but precise enough to demonstrate MDS. By only looking at the table, it is hard to get the overall picture of where each city is located. Even for such a small example, the visualization improves our understanding of the data a lot.

Figure 4.5 shows the result of the MDS algorithm if I provide the distance matrix as similarity measure. The picture looks familiar, as the cities are arranged in a similar fashion that we usually see when we look at a map of Europe.

However, MDS only optimizes the relative distances of the cities, and there are infinitely many solutions that satisfy this requirement as we can for example scale the true distance by any natural number while still preserving the relative distances. The result can be translated (shifted in x- and y-direction), scaled, rotated and reflected and would still not change the solution, i.e., the relative distances would not be different. For this example, I picked one solution by hand that looked good to me, but in general, I do not have a ground truth available to compare the MDS result.

The result of the MDS algorithm is called a *configuration*, which consists of

Table 4.3: **Distance matrix of seven European cities**. The distances were obtained through measurements done with Google Maps.

	Vienna	**Berlin**	**London**	**Paris**	**Rome**	**Amsterdam**	**Brussels**
Vienna	0	523	1235	1036	763	938	915
Berlin	523	0	933	878	1186	580	655
London	1235	933	0	348	1435	358	318
Paris	1036	878	348	0	1103	427	260
Rome	763	1186	1435	1103	0	1302	1178
Amsterdam	938	580	358	427	1302	0	175
Brussels	915	655	318	260	1178	175	0

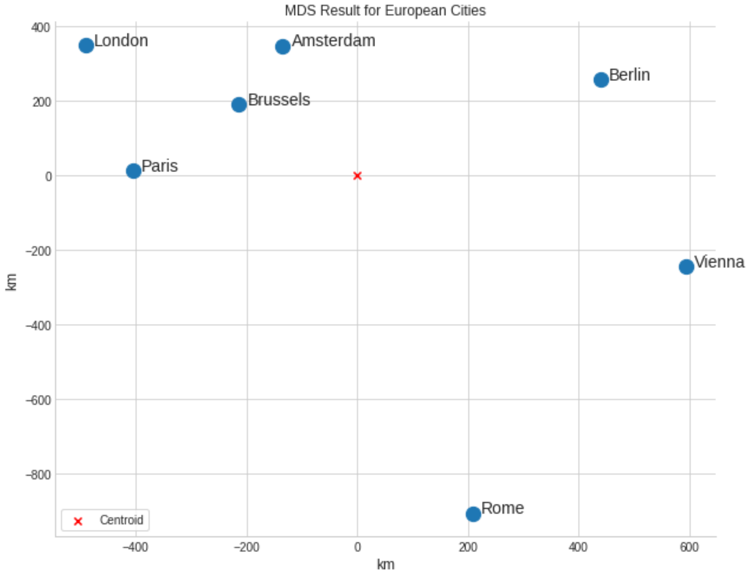


Figure 4.5: **MDS result of European cities.** The algorithm yields a result with the point coordinates relative to the centroid in origin $(0, 0)$. Any solution that is rotated or reflected is equally valid, i.e., the relative distances would also be the same. However, I picked this solution by hand, as it looks similar to the view we are used to in a world map.

the coordinates of the points corresponding to the elements of the similarity matrix. To evaluate a configuration Kruskal [**Kruskal1964**] defines the *stress* as measure of goodness-of-fit, which is the square root of the normalized sum of squared errors:

$$stress = \sqrt{\frac{\sum_{i<j}^{n} (d_{ij} - \hat{d}_{ij})^2}{\sum_{i<j}^{n} d_{ij}^2}}$$

I want to find a configuration that minimizes the stress, i.e., the configuration with the least squared error. This is an optimization problem which can be solved by many algorithms such as Gradient Descent or Simulated Annealing. For my implementation in Python I use the package `sklearn.manifold.MDS`[1]

[1] http://scikit-learn.org/stable/modules/generated/sklearn.manifold.MDS.html

which already implements this optimization.

4.3.2 Procrustes Problem

By the definition of the stress (which is optimized in MDS) there are many possible solutions, because the distances are invariant of scale, rotation, reflection, and translation. To compare the result of the MDS algorithm directly to the ground truth, I need to find the correct transformations.

The problem of finding the right rotation and reflection is called orthogonal Procrustes problem which I solve by using the Python package `scipy.linalg.orthogonal_procrustes` [2]. It uses the work of [Sch66], who showed that given two matrices P and Q, one can find the orthogonal transformation matrix R, that maps P onto Q using the cross correlation matrix $M = QP^T$. The singular value decomposition of M equals $U\Sigma V^T$, and it can be shown that one can use parts of this decomposition to write R as $R = UV^T$. The result is a 2×2 transformation matrix R which I can use to transform the coordinates of our MDS result.

I shift both, the ground truth coordinates, as well as the points I got from MDS so that the corresponding centroid is in the origin, by subtracting the mean point of each dataset. This way I can eliminate all differences caused by translation.

The last transformation I have to care for is scale. I normalize the coordinates of both datasets so that they are between 0 and 1 with this formula:

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}$$

where z_i is the new normalized value. I use denominator that I used for normalizing the ground truth coordinates to rescale all coordinates to the actual scale. The result of the process can be seen by examing Figure 5.5.

[2] https://docs.scipy.org/doc/scipy/reference/generated/scipy.linalg.orthogonal_procrustes.html

4.4 Evaluation

For my experiments I constructed ground truth datasets that consist of the coordinates of all tags, relative to tag 0.

I use the ground truth and the transformed estimated coordinates (see Section 4.3) to calculate (i) the Pearson correlation coefficient (see Section 4.4.1), and (ii) the Mean Absolute Error (see Section 4.4.2).

To understand how an approach works and why it performs good or bad it is often not enough to only get the accuracy of the method as a single value. Visualizing the estimated distances can help to interpret the results through manual inspection. Besides, in real world there is no ground truth available, and visualization can help to evaluate the outcome. For example, the store staff could visually check on sample basis if the estimated distances correspond to the real distances. Hence, I use visualizations of the result as a third metric to evaluate the results of MDS.

4.4.1 Pearson correlation coefficient of distances

In Section 4.3.2 I discussed how I transform the estimated coordinates so that they match the ground truth as close as possible. I calculate distance matrices $\mathbf{D}$ based on the ground truth coordinates and $\widehat{\mathbf{D}}$ based on the transformed estimated coordinates. I evaluate the result by computing the Pearson correlation coefficient between distances in $\mathbf{D}$ and $\widehat{\mathbf{D}}$.

Let $\mathbf{D}$ be an $n \times n$ distance matrix based on the ground truth coordinates of n tags, $\widehat{\mathbf{D}}$ be the matrix of distances derived from the predicted coordinates with the same dimensions. I transform the upper triangular matrix of $\mathbf{D}$ to a vector $\mathbf{x}$ and $\widehat{\mathbf{D}}$ to $\mathbf{y}$ respectively, by stacking the corresponding row vectors on top of each other. As a result, $\mathbf{x}$ and $\mathbf{y}$ contain the ground truth distances and the predicted distances respectively in the same ordering. One way to calculate the *Pearson* correlation coefficient would be using this formula:

$$r_{xy} = \frac{\sum_i^n (x_i - m_x)(y_i - m_y)}{\sqrt{\sum_i^n (x_i - m_x)^2 (y_i - m_y)^2}}$$

where m_x is the mean of vector $\mathbf{x}$ and m_y is the mean of vector $\mathbf{y}$. I use the implementation of the Python package *scipy*[3].

The Pearson correlation coefficient has a range from -1 to 1. A value of 1 means total correlation between $\mathbf{x}$ and $\mathbf{y}$, which implies that a linear equation perfectly describes the relationshipt between the two. A value of -1 describes an inverse relationship, while a value of 0 means no relationship at all.

4.4.2 Mean Absolute Error of the coordinates

After transforming the predicted coordinates as described in Section 4.3.2 I can also calculate the Mean Absolute Error (MAE) of the coordinates. I use the euclidean distance between the ground truth coordinates and the predicted coordinates, and calculate MAE as follows.

Let $\mathbf{x}_i$ be the ground truth coordinate vector of the i-th tag and $\mathbf{y}_i$ be the predicted coordinates. For n tags, I calucate the MAE of the coordinates using this formula:

$$MAE = \frac{1}{n}\sum_{i}^{n}(\|\mathbf{x}_i - \mathbf{y}_i\|) \text{ where } \|\cdot\| \text{ is the euclidean distance}$$

In this chapter I discussed the methodology I use to estimate relative RFID tag locations based on time-differences in read events. To that end, I described the data preprocessing steps I perform and present three different approaches to estimate relative tag distances based on read events.

Moreover, I discussed how I derived coordinates from the distance estimations using Multidimensional Scaling. Addionally, I describe how I perform Procrustes analysis to find the rotation, scale, and transformation that maps the approximated coordinates as close as possible to the ground truth. Finally, this chapter discussed the metrics I use to evaluate my results.

The next chapter describes the setup for all the experiments I conducted in detail.

[3] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.pearsonr.html

5 Experimental Setup

In this chapter I discuss the location and environment in which I conducted my experiments in Section 5.1. Then I introduce the equipment of the RFID system I use for all the experiments. Specifically, in Section 5.2, I first describe the reader and its configuration, followed by a description of the tags I use in my experiments. Afterward, I explain what kind of data I collect during my experiments in Section 5.3. In Section 5.4 I discuss the motivation and design decisions regarding the framework I created and then describe the three different setups I constructed. Finally, in Section 5.5 I present the experiments I conducted.

5.1 Facilities

I conducted all experiments in the workshop at Detego, which is a large room containing work tables, toolboxes, and racks, full of things such as tools and RFID hardware. Photographs of the facilities can be seen in Figure 5.4, which shows the different setups I construced.

The facilities contained several metal constructions including a conveyor belt, hence, the environment was not perfect for RFID read performance. However, in real-world, an RFID application might have to deal with similar situations, as many fixtures inside a store are also manufactured using metals.

Detego is continuously testing RFID hardware in their workshop. As a result, RFID tags other than the ones I used for my experiments were spread across the room from previous tests and projects. These tags further affect the read performance in my experiments. However, as I only had a small set of tags for my experiments, more tags only made my setup more realistic.

5.2 RFID Hardware

5.2.1 Reader

The RFID reader device I used for the experiments was a *ZEBRA RFD8500 Handheld* which is shown in Figure 5.1 and from here on referred to as *reader*. The device is configured to use *Session 0* in *AB flip* mode, meaning that the reader reads all tags with the inventory flag set to A, and changes their flag to B. Afterwards the reader automatically queries all tags with inventory flag set to B and changes their flag back to A (see Section 2.2.5 for a detailed description about sessions). This process is repeated until the user decides to stop. As a result, the reader reads all the tags all the time limited only by the performance of the reader itself.

I set the power level of the reader to the maximum value of 30.0 dBm which equals 1W for all our experiments. I connect a mobile device running the Android operating system via the integrated Bluetooth interface and store the collected data on the phones internal storage.

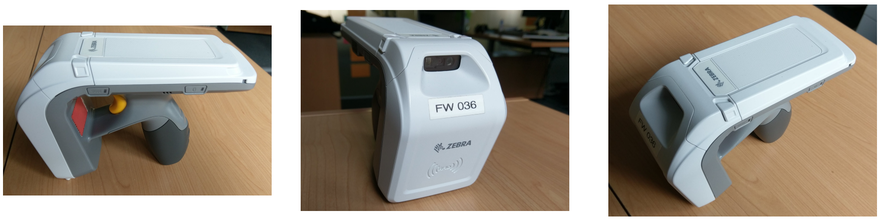


Figure 5.1: **The RFID reader I used for my experiments.** It is a *ZEBRA RFD8500 Handheld* with maximum power of 30 dBm, which is equal to 1W. I operate the reader at *Session 0* in *AB flip* mode.

5.2.2 Tags

For my experiments, I selected 70 tags of the model *UPM Web 208_3*, which is a model commonly used in retail applications (see Figure 5.2). Note that the tags were not new but already used; hence, the performance of the tags might differ under similar experimental setups. However, I made sure that all tags tags could be detected by the reader, and thus work as intended.

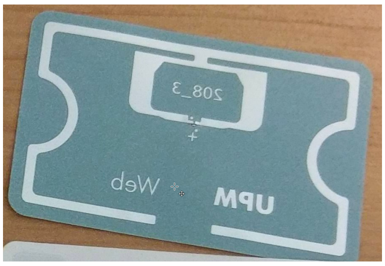


Figure 5.2: **Example of the tags I used** in the experiments. Such tags (*UPM Web 208_3*) are commonly used in retail applications.

To be able to distinguish the tags, I encoded each of them with a dummy EPC with a serial number ranging from 0 to 69. For the remainder of this thesis, whenever I refer to the tag encoded with an EPC with serial number x, I will just call it *tag* x.

5.3 Dataset

The dataset I use solely consists of the data that I collect during my own experiments. Whenever the reader receives a response from a tag, in other words, for each read-event, a log entry is written. For each read-event I store (i) the Unix timestamp in milliseconds, (ii) the serial number of the EPC encoded on the tag and (iii) the RSSI in dBm of the received signal. See Table 5.1 for examples.

Table 5.1: **Example data of read-events.** The values are the timestamp in milliseconds, tag number and the RSSI value in dBm.

Timestamp	Tag	RSSI
1510236744359	12	-63
1510236744375	8	-62
1510236744376	10	-59
1510236744377	6	-64
1510236744379	16	-61

The amount of data I collect varies between experiments and depends on how long the experiment lasted and on the amount and frequency of tags read during the experiment. I mention the exact number of captured read events for each experiment separately in the next Section.

5.4 Framework

Fashion retailers are creative in the way how they present their goods on the sales floor. Stores are constantly reorganizing and moving their equipment to provide a fresh and modern look at their clothes. However, despite all the efforts they put into store design, the core elements still include shelves, racks, hanging rails, and tables. Hence, for my experiments, I used tables as the basic building blocks to construct setups that might occur in real-world stores.

Unfortunately, I did not have tagged apparel and therefore had to use only the tags for my experiments, which had several implications. In a real-world scenario, the clothes often cover the tags, and sometimes tags are lying on top of each other. Moreover, tags attached to clothes typically do not have the same rotation, which has a strong influence on RFID read performance. Furthermore, when clothes lie in stacks on a table, the tags are located in different heights (e.g., the tag on the t-shirt at the bottom of a stack is lower than the tag belonging to the t-shirt on top of the stack). Due to the lack of tagged apparel, I could not simulate these factors in my experiments.

I attached the tags in groups of ten on polystyrene panels, where each panel simulates a pile of clothes of the same product class (e.g. one panel represents ten black T-shirts in different sizes and another panel corresponds to blue jeans). As they do not influence the read performance negatively, I use the polystyrene panels for more comfortable handling during the setup of the experiments. Figure 5.3 shows an example of a polystyrene panel. On each panel I arranged the tags in two rows, where the front row contained the tags with lower serial numbers and the back row tags with the higher serial numbers (e.g., panel 1 has tags 10 to 14 in the front and tags 15 to 19 in the back).

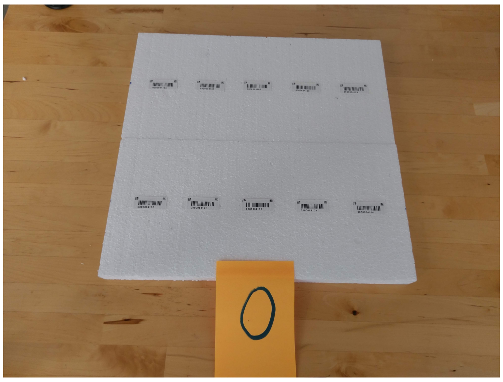


Figure 5.3: **RFID tags sticked on a polystyrene panel.** Panel 0 contains the tags with serial number 0 to 4 from left to right at the bottom and the tags 5 to 9 from left to right at the top. The other panels have their corresponding tags on them in the same ordering.

I constructed three different setups to simulate real-world scenarios in a store. I describe the first and most basic setup which I call *2D Setup* in Section 5.4.1. This simple arrangement was part of a small feasibility study and is the basis for the other two setups that I call *2D / asymmetric Setup* (see Section 5.4.2 and *3D Setup* (see Section 5.4.3).

For each of the three setups, I measured the distances between the tags to construct a ground truth dataset. The dataset consists of the measured coordinates of each tag, which I use to calculate the ground truth distance matrix. With the ground truth coordinates and distance matrix I can evaluate my results using MAE and *Pearson* correlation coefficients (see Section 4.4). Photographs as well as plots of the ground truth datasets can be seen in Figure 5.4.

The following sections discuss the three setups in detail and describe all the experiments I conducted.

5.4.1 2D Setup

I arranged four tables in a rectangular shape, and put seven panels (i.e., 70 tags) on them. Except for one table which had only one plate on it, each table contained two panels which equals 20 tags. Figure 5.4a shows a photograph of the setup on the left, and the corresponding ground truth dataset on the right.

With this setup I intended to simulate a small booth on a sales floor in which the clothes are arranged in a rectangular way. I expect that all proposed algorithms work reasonably well in this setup.

5.4.2 2D / asymmetric Setup

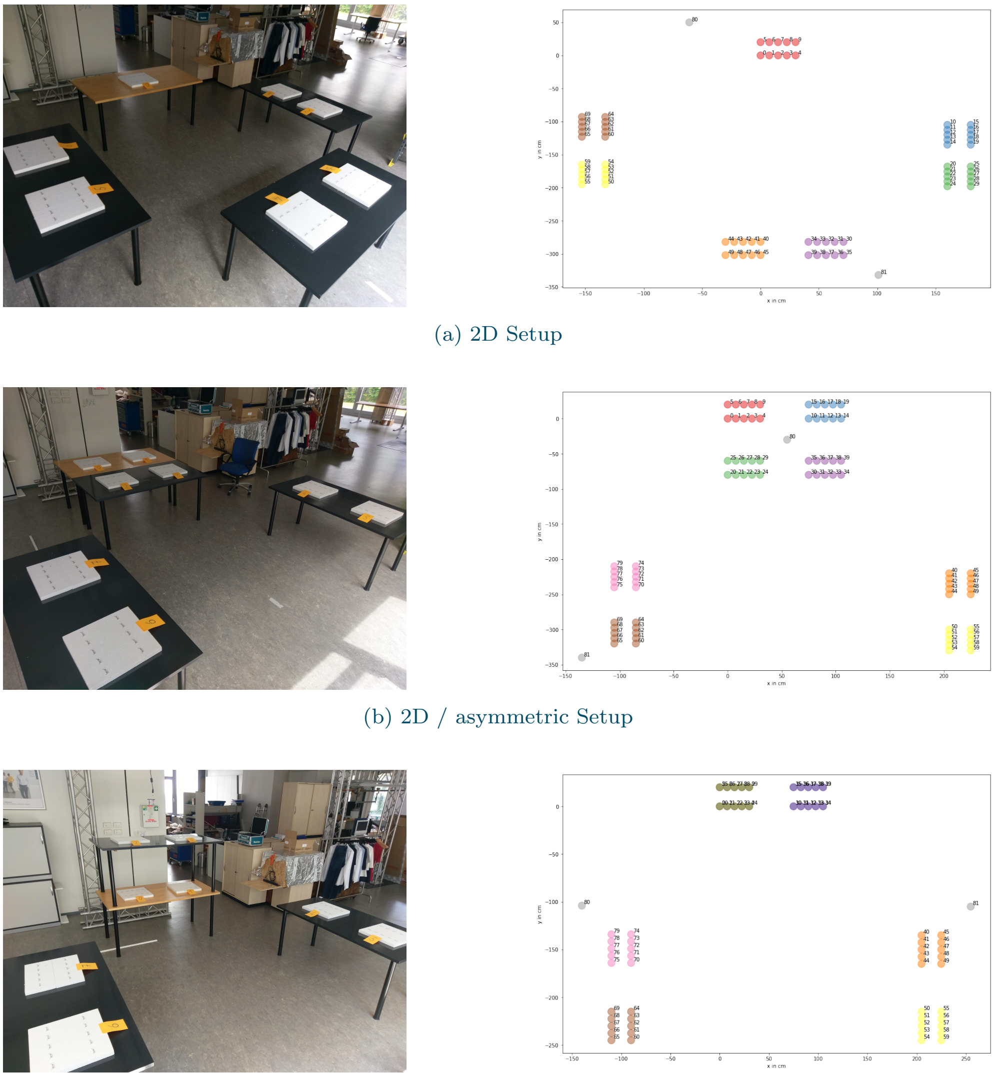
This setup is an extension to the *2D Setup* (see Section 5.4.1). On one table I put two panels on the front, and two panels at the back as shown in Figure 5.4b. As a result, to estimate the distances accurately, an algorithm has to recover depth information from the data.

Fashion retail stores can be very large, meaning they can potentially have tens to hundreds of thousand items on the sales floor. Moreover, very often the arrangement of items on a sales floor is very dense, in other words, there are often a lot of items on small space. RFID readers get a vast number of responses of tags at the same time. These tags can be far or close to the reader, hence, the ability of recovering depth is important for the estimation of the distances.

5.4.3 3D Setup

This setup aims to simulate a shelf in which clothes are located on top of each other in different boxes. Shelves are often used in a store, hence I wanted to try how well my algorithms work, even if I only provide coordinates and a visualization in two dimensions. After all, a 2D visualization can still be useful. Items on top of each other would have the same coordinates in 2D space, and if store staff is searching for an item, I can at least provide information on which shelf the item is located.

For the simulation, I put two tables with two plates on them on top of each other (see Figure 5.4c). As some tags are located directly on top of each other, in the plot of the 2D ground truth dataset, the lower tags are covered by the tags above them.

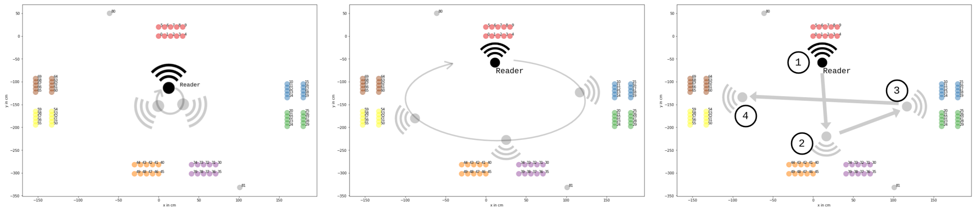


(a) 2D Setup

(b) 2D / asymmetric Setup

(c) 3D Setup

Figure 5.4: **Photographs and ground truth datasets of all setups.** Each row corresponds to a different setup, showing a photograph on the left and the ground truth dataset on the right. Tags are drawn as circles and are colored differently for each styrofoam plate. Note, that in Figure 5.4c the tags 0 to 19 are covered by tags 20 to 39 because they are directly above them on the upper table.



(a) (b) (c)

Figure 5.5: **Schematic of the three different experiments I conducted.** For the first experiment shown in Figure 5.5a, I read all the tags from a fixed point in the center of the setup. Figure 5.5b shows the *Circular Walk* experiment, where I walk clockwise from table to table. The *Random Walk* is similiar, but here I choose a random table sequence.

5.5 Experiments

For every setup, I performed mutliple experiments which I describe exemplary for the 2D Setup (see Section 5.4.1). The experiments are the same for the other setups.

I repeated all experiments three times, doing (i) one iteration, (ii) two iterations and (iii) four iterations, where one iteration means that I pointed the reader in the direction of each tag at least once. The idea is to do multiple iterations to collect more data and test if an algorithm can take advantage of that.

Note, that I use the phrase "*the reader points in the direction of each tag*", because I cannot guarantee that the tag is actually in the focus of the reader (i.e., the tag is powered by the field of the reader). As discussed in Section 2.2, RFID read performance is affected by a variety of factors such as the rotation of a tag, and metals that reflect signals. For that reason, in most of my experiments there were some tags that I did not detect with the reader.

I also repeated the experiments for each iteration 3 times.

5.5.1 Fixed Read Point

The first experiment was reading all tags from a fixed point in the center of the setup. As schematically shown in Figure 5.5a I turned for 360 degrees to try to have each tag once in focus for one iteration. For two and four iterations I turned for 360 degrees twice and four times correspondingly.

The motivation for this experiment is that a store employee might stand in the middle of multiple tables and try to scan the clothes wihout approaching the tables any further. In this setup the amount of tags that could not be read is potentially high, due to the distance between reader and tags, and tags possibly shielding other tags. Table 5.2 shows the data I collected during the 2D setup.

Table 5.2: **Datasets for the Fixed Read Point Experiments.** The table describes the data I collected during the Fixed Read Point experiments on the 2D Setup (see Section 5.4.1). For each number of iterations, I show the number of read events, median number of read events per tag, the duration of the experiment, and how many tags I read during the experiment.

	# iter	# Events	# Events per Tag	Duration	Tags Read
Fixed Read Point	1	1,742	26	16s	56 (80%)
	2	2,882	39	25s	59 (84%)
	4	4,124	57	49s	54 (77%)

5.5.2 Circular Walk

Read performance drastically improves as the distance between reader and tags decreases. Therefore, in most cases store staff walk from table to table when doing stocktake. In this experiment I want to simulate this behaviour, by walking clockwise from plane to plane starting with plane 0. Figure 5.5b shows a schematic of the Circular Walk.

Table 5.3 shows the collected datasets per setup and number of iterations. As I repeated each experiment 3 times, the table shows the median values of the number of recorded read events, number of read events per tag, the duration of the experiment and the number of unique read tags. Naturally, all these values increase linearly with the number of iterations.

Table 5.3: **Datasets for the Circular Walk Experiments.** On each setup I try a different number of iterations (i.e., how often I try to read all tags). I repeat each number of iterations 3 times and show the median number of read events collected as well as the median number of read events by tag and the median duration of the experiment. Due to the nature of RFID, I am not able to read all the tags in every experiment, hence I show the median number of read RFID tags in the last column.

	# iter	# Events	# Events per Tag	Duration	Tags Read
2D Setup	1	2,769	41	20s	66 (94%)
	2	6,010	85	46s	68 (97%)
	4	8,841	126	86s	70 (100%)
2D / asymmetric Setup	1	1,486	26	22s	53 (66%)
	2	3,710	46	41s	58 (72%)
	4	4,718	62	80s	61 (76%)
3D Setup	1	2,334	36	22s	65 (81%)
	2	5,054	63	48s	75 (93%)

5.5.3 Random Walk

I expect that in real-world, store employees follow a certain path when walking through the sales floor; however, they will not always walk clockwise from table to table. In many cases this would not even be possible.

As a result, in my last experiment, I perform a Random Walk from table to table. This means that starting from plane 0, I randomly choose which table I visit next. However, when I do multiple iterations, I keep the same sequence of tables. I did this experiment only on the 2D Setup (see Section 5.4.1), a schematic of the random walk is depicted in Figure 5.5c. The data I collected during my experiments is summarized in Table 5.4. I could read almost all tags in the experiments (from 92% to 98%) and collected between 34 and 86 read events per tag.

Table 5.4: **Datasets for the Random Walk Experiments.** I only performed the random walk experiment on the 2D Setup (see Section 5.4.1). I repeated the experiment for each number of iterations 3 times and in the table I show only the median values. The table presents the number of collected read events, read events per tag, the duration of the experiment in seconds, and the number of unique tags read.

	# iter	# Events	# Events per Tag	Duration	Tags Read
Random Walk	1	2,637	34	19s	65 (92%)
	2	5,008	67	40s	69 (98%)
	4	5,808	86	80s	68 (97%)

6 Results and Discussion

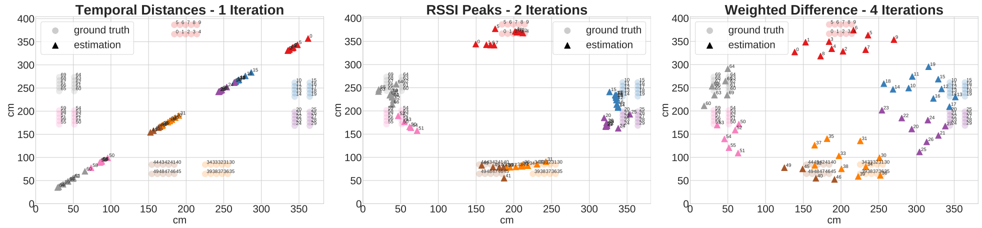
In this chapter, I discuss the results I obtained in my experiments. For each experiment, I compare MAE and correlation coefficients of my algorithms for the different setups, and show visualizations of the obtained outcomes.

6.1 2D Setup

Results vary considerably with the number of iterations performed during the experiments. However, the Weighted Sum algorithm achieved consistently worse outcomes than the other two algorithms (see Table 6.1).

Table 6.1: **Results on the 2D Setup.** In this Table, I describe results I obtained performing experiments on the 2D Setup. I show for each experiment the number of iterations, the MAE and the correlation coefficient r for each algorithm. I calculate the MAE between the estimated and the ground truth coordinates, and r between the pairwise tag distances in the ground truth and the estimation.

	# iter	**Temporal Distance**		**RSSI Peaks**		**Weighted Sum**	
		MAE	r	MAE	r	MAE	r
Fixed Read Point	1	68.12	0.84	57.88	0.85	52.65	0.85
	2	30.29	0.92	24.91	0.96	37.26	0.91
	4	98.19	0.50	92.06	0.56	95.72	0.59
Circular Walk	1	126.80	0.75	101.95	0.73	83.60	0.72
	2	28.36	0.93	36.07	0.92	43.52	0.88
	4	28.65	0.94	29.18	0.94	50.17	0.84
Random Walk	1	122.52	0.50	119.41	0.51	137.25	0.49
	2	120.73	0.49	116.01	0.52	113.58	0.58
	4	116.33	0.60	114.34	0.58	126.42	0.49



(a)

(b)

(c)

Figure 6.1: **Results of the Fixed Read Point experiment on the 2D Setup.** In this Figure, I show a visual representation of the results I obtained on the dataset with a fixed read point. Figure 6.1a shows that for a single iteration, my approach maximises the distance between the first and the last read tag (the results of single iterations for the other algorithms look similar). In Figure 6.1b I present the best result I achieved, using the RSSI Peaks algorithm. Figure 6.1c shows the result of the Weighted Sum algorithm using data from 4 iterations. Distances between tags of the same cluster are consistently higher using this algorithm compared to the others.

Single Read Occurrences

My results indicate, that for a single iteration (i.e., one round, without recurring reads of any of the first tags at the end of the experiment), my approach infers the maximum distance between the first and the last read tag in the experiment. This is true for all my proposed algorithms. As a result the visualization is inaccurate and for all approaches just a diagonal line (see Figure 6.1a). The MAE is relatively high, ranging from 52.65 cm to 68.12 cm for the Fixed Read Point experiment and from 83.60 cm to 126.80 cm for the Circular Walk (see corresponding rows with one iteration in Table 6.1).

However, while the estimated coordinates do not necessarily reflect the ground truth, the tags are still grouped correctly (see Figure 6.1a). Note, that tags on the same panel (i.e., tags in the same color) are grouped together and distances to tags from other panels are higher than to tags in the same cluster. Furthermore, tags on panels on the same table (e.g., blue and purple tags) are closer to each other than to tags on a different table (e.g., red tags). The correlation coefficient also supports this observation with high values from 0.84 to 0.85.

Recurring Reads
As soon as the dataset contains recurring reads of tags, the results improve drastically. Especially recurring reads of starting tags at the end of the experiment lead to estimated coordinates close to the actual coordinates of the tags. All algorithms can take advantage of data collected from a second iteration. In Table 6.1 I show that the results improve compared to results with only 1 iteration (see rows for 2 iterations). Both, MAE as well as the correlation coefficient reflect this improvement.

Multiple Walk-Throughs
The implications of 4 iterations (i.e., four rounds, with four time windows with several read events for each tag) are different, depending on the experiment and the algorithm. In case of the Fixed Read Point algorithm, MAE and correlation coefficient r are consistently worse than with 2 iterations. Specifically, MAE is about 3 times as high, and r is also considerably smaller with values between 0.5 and 0.59 compared to 0.91 - 0.96 for 2 iterations. These results indicate that more iterations result in more potential for noisy data which influences the results.

I could not keep the speed of my movement consistent enough and produced more noisy data with .

On the other hand, for the Circular Walk experiment, there is little improvement to observe for the Temporal Distance and RSSI Peaks algorithms, while the results for the Weighted Sum algorithm are worse.

Overall, I achieved the best result for the Fixed Read Point experiment with the RSSI Peaks algorithm with the MAE being 24.91 cm and the correlation coefficient 0.96 (see Figure 6.1b). For the Circular Walk experiment, the lowest MAE was 28.36 cm with a correlation coefficient of 0.93, achieved with the Temporal Distance algorithm.

Discussion

Recurring Reads
For all my approaches, the main feature is the time when a tag was read during the experiment. The RSSI value is the only additional information I can use. Due to this limited amount of information, it is vital for all my approaches

that recurring reads of tags happen during the experiment. The visualization in Figure 6.1a depicts how my algorithms struggle without recurring reads. The lack of recurring reads of tags on the first panel (i.e., red tags), lead to missing distance information between the first and the last panel. As a result, tags are positioned along a line.

Nevertheless, even without recurring reads, I am able to clearly separate tags on different panels and tables (see colors of the tags in Figure 6.1a). Note that the distances between panels on different tables are higher. For example, the distance between red tags and blue tags is higher than the distance between blue tags and purple tags, as the latter are located on the same table.

Constant Speed
Results show that movement with constant speed is an important requirement. In Table 5.2 I showed the durations of the Fixed Read Point experiments depending on the number of iterations. One potential explanation for the decrease in localization accuracy might be changes in speed. The numbers indicate that I could not maintain a constant speed well enough, it seems that I became faster with more iterations. 1 iteration took 16 seconds, while 4 iterations took 49 seconds, which is less than 4 times 16 seconds. On the other hand, the durations for the Circular Walk experiments correspond almost perfectly (see Table 5.3).

Data for 4 iterations improved the results for the Circular Walk experiment, in which I could maintain constant speed. Regarding the Fixed Read Point experiments in which I became faster with more iterations, results became worse.

As a result, my observations support the hypothesis that constant speed is very important and a basic requirement for my approaches. However, the Fixed Read Point experiments are especially vulnerable even to small variations in speed due to the sharp angle between reader and tags, as small movement can change the entire focus of the reader. This is not the case for the Circular Walk experiment where a speed change of walking typically does not change the entire focus of the reader.

In future work I could tackle this problem by using additional information such as step detection of other smartphone sensor data.

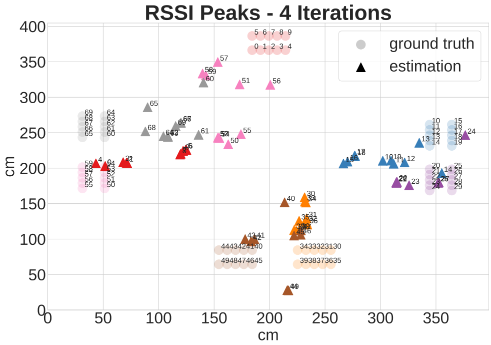


Figure 6.2: **Random Walk results on 2D Setup**. This Figure depicts the result of the RSSI Peaks approach on the Random Walk experiment. I am not able to recover the true locations of the tags; however, distances between brown & orange, and blue & purple tags are reasonable.

Circular Sequence

In both experiments, Fixed Read Point and Circular Walk, the panels are read sequentially in a circular read pattern. I always start with panel 1, panel 2, panel 3, and so on, until the last panel. All results presented in this Section are subject to this limitation. In the next Section, I am covering how my approaches react to a random reading pattern.

6.2 2D Setup - Random Walk

For my Random Walk experiment I chose the sequence as shown in a schematic in Figure 5.5c. Note that I did not change the sequence for multiple iterations, and I used the same sequence for all experiments. Overall, Table 6.2 shows that when performing a random walk, results, in terms of MAE and r, get worse. With MAE of over 110 cm and correlation coefficient only up to 0.60, results are noticeably worse than for all other experiments. However, the visual representation of one of the results in Figure 6.4 shows, that even in this case, the panels are still distinguishable and the general placement of the tags appears to be correct although not as good as in the other experiments. Distances between brown & orange tags and blue & purple tags look reasonable.

Discussion

Given my only features are the time and RSSI of a read event, it is not surprising that my approaches struggle in this experiment. As distance is proportional to time differences in read events, the read sequence must correspond to the distances between the tags in order to accurately localize all tags. However, Table 6.2 indicates that at least for the Temporal Distance and RSSI Peaks algorithms, results tend to improve with more iterations. The MAE decreases, while the correlation coefficient increases. I hypothesize that in the case of the Random Walk experiment, more iterations with different reading sequences might improve results here. To verify this hypothesis, more experiments are needed, which I leave open for future work.

6.3 2D / asymmetric Setup

The goal for the *2D / asymmetric* setup was to test if my approaches are able to reconstruct depth information. In this setup I only read 66% to 76% of all the tags. Looking at the results, I found that most missing tags were located in the back of the table with four panels (see red & blue tags in Figure 6.3).

Results for this setup highlight again the fact that a single iteration of read events is not enough, as we observe high MAE values (95cm to 111cm) and low correlation coefficients ranging from 0.62 to 0.64 (see Table 6.3). As with previous experiments, two iterations improve the results considerably, with the best result in terms of correlation coefficient using RSSI Peaks approach

Table 6.2: **Random Walk Results.** In this Table I list MAE and the correlation coefficient r for my Random Walk experiments (# iter) on the 2D Setup. I calculate MAE between the coordinates of our ground truth and estimated RFID tag coordinates, and the correlation coefficient between the temporal distances and the distances in our ground truth.

	# iter	**Temporal Distance**		**RSSI Peaks**		**Weighted Sum**	
		MAE	r	**MAE**	r	**MAE**	r
Random Walk	1	122.52	0.50	119.40	0.51	137.25	0.49
	2	121.05	0.53	117.67	0.54	113.58	0.58
	4	116.33	0.60	114.35	0.58	126.42	0.49

with $r = 0.77$. The lowest MAE could be achieved by the Weighted Sum approach with 77.86 cm. Four iterations lead to worse results compared to two iterations.

Visual representations again show that tags on the same panel are grouped together. Figure 6.3a shows the best result using the RSSI Peaks approach, where tags on the same panels are dense clusters of tags. Tags in yellow, brown, pink and gray are visualized in corresponding clusters which are easily separable from each other. Due to the small space between them in the ground truth, the other tags form a single cluster in which little depth differences can be observed as red and blue tags are more in the back and purple and green tags more in the front.

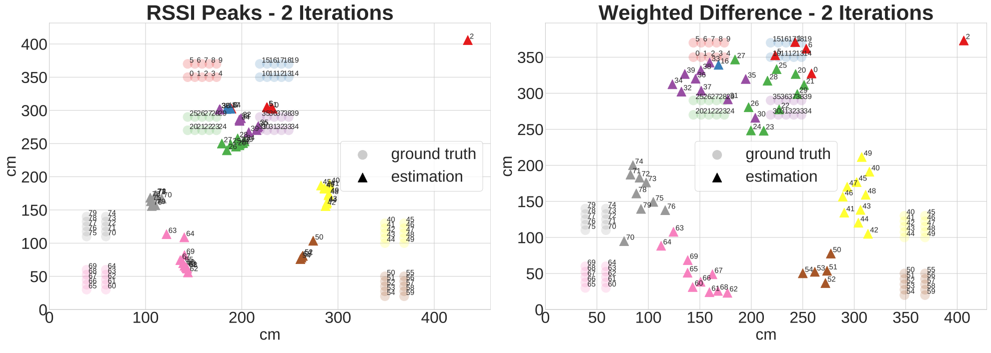
Table 6.3: **Results on the 2D / asymmetric Setup.** In this Table, I describe results I obtained performing experiments on the 2D Setup. I show for each experiment the number of iterations, the MAE and the correlation coefficient r for each algorithm. I calculate the MAE between the estimated and the ground truth coordinates, and r between the pairwise tag distances in the ground truth and the estimation.

	# iter	**Temporal Distance**		**RSSI Peaks**		**Weighted Difference**	
		MAE	r	MAE	r	MAE	r
2D / asymmetric	1	111.19	0.62	105.71	0.64	95.45	0.64
	2	88.31	0.76	86.17	0.77	77.86	0.76
	4	89.98	0.62	93.73	0.62	91.15	0.61
3D	1	122.14	0.71	105.21	0.68	94.55	0.58
	2	57.49	0.87	55.02	0.85	78.18	0.68

The visualization of the estimated tag locations by the Weighted Sum approch in Figure 6.3b looks similar, however, MAE is higher meaning the tags are more spread accross the space. Distances between tags in the same panel are much higher in comparison to the distances in Figure6.3a.

Discussion

Readability of tags is heavily influenced by metals (other tags), as they shield and reflect electromagnetic waves (see Section 2.2.2 for details). As a result, RSSI values are noisy and it is challenging to reconstruct depth information in such a dense configuration of tags, as in this setup. Unfortunately, many tags



(a)

(b)

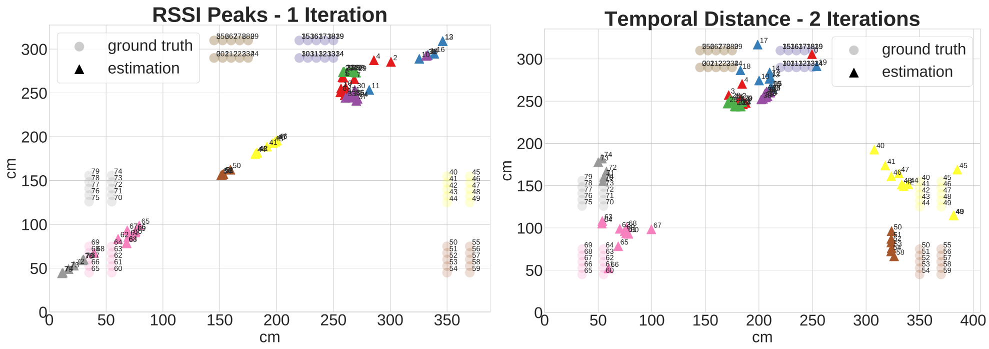
Figure 6.3: **Results on the 2D / asymmetric setup.** Figure 6.3a depicts the estimation using RSSI Peaks, which yields dense clusters of tags. Figure 6.3b show the result of the Weighted Sum algorithm, where distances between tags of the same panel (same color) are higher. In both cases little depth information is recovered, as green and purple tags tend to be in front of the blue and red tags. Both results contain an outlier (Tag 2) at the top right corner. Due to the dense configuration of the tags, only a few of the tags in the back (red and blue tags) could be read.

in the back (red and blue tags) could not be read during the experiment as they were shielded by other tags. Nevertheless, I see signs that depth can be reconstructed as green and purple tags are further in the front, while the blue and red tags appear to be further in the back. This is true for all approaches and is depicted for the RSSI Peaks and Weighted Sum approaches in Figure 6.3.

Note that in both cases (Figures 6.3a and 6.3b), the positions of blue and red tags are mirrored. In Figure 6.3b, the same is true for purple and green tags. The reason for this is that during the experiment I moved the reader in a circular shape, meaning that I read the tags in the sequence red, green, purple, blue, and then in the same sequence backwards. This observation proves that it is very challenging to recover the actual locations in dense tag configurations, as small reader movements have a potentially large impact on the results.

The fact that all approaches produce an extreme outlier (see Tag 2 in top right corners in Figure 6.3) is a little bit discouraging for real-world applications. The estimated location is more than 1 m away from all other tags.

Nevertheless, I think for finding misplaced items the results are still useful. My approaches can give information of other nearby items and by that, guide store



(a)

(b)

Figure 6.4: **Circular Walk results on 3D Setup.** This Figure depicts the result of the RSSI Peaks approach on the Random Walk experiment. I am not able to recover the true locations of the tags; however, distances between brown & orange, and blue & purple tags are reasonable.

staff in the right direction.

6.4 3D Setup

For the 3D setup, I achieve an MAE of 57.49 cm and a correlation coefficient $r = 0.87$ using the Temporal Distance approach. When I look at the rows corresponding to the 3D setup in Table 6.3, I can observe a similar pattern as in previous experiments. A single iteration of read events is not sufficient to recover the shape of the ground truth tag positions (see Figure 6.4a). However, Figure 6.4b as well as the results in Table 6.3 show that two iterations substantially improve the results.

Discussion

Ground truth distances between tags that are directly on top of each other are 0, as they are calculated based on 2D coordinates. These distances can not be 0 in my experiments, as it takes time to move the reader from the tags at the bottom to the tags at the top. As a result, an MAE of 0 cm is not possible in this setup.

Throughout all experiments, the results of the Temporal Distance approach consistently underestimated distances of tags on the same panel. On the other hand, the Weighted Sum approach always overestimated these distances, while the RSSI Peaks approach yields results roughly between the other two. Hence, it is no surprise that I get the best results using the Temporal Distance approach in this setup, as distances of tags that are on top of each other are closer to 0.

All my approaches put tags that are on the shelf like construction (i.e., panels on tables standing on each other), very close to each other. As a result, in real-world applications my approaches could tell on which shelf an item is located.

7 Conclusion and Future Work

In this chapter, I summarize the main contributions of this thesis and give an outline of future work in the field of localizing passive RFID tags in the context of fashion retail applications.

7.1 Conclusion

In this work I presented a novel methodology to infer relative distances between RFID tags, leveraging time-based differences in read events. I proposed three algorithms to estimate distances between RFID tags which use differences in time and RSSI between read events of tags. Furthermore, I evaluated the results against a ground truth using three different measures, using i) the MAE of the absolute positions of the tags, ii) the pearson correlation coefficient of the pairwise relative distances between tags, and iii) by inspecting a visualization of the results.

Results show that I am able to infer positions of the RFID tags, depending on the layout of my experiments with an MAE of up to 28.36 cm and a correlation coefficient of up to $r = 0.95$. Further, the results of all my experiments show that, using my approaches, it is possible to identify clusters of products, i.e., tags on the same panel. This translates to articles that are in close proximity, for example, on the salesfloor in a fashion retail store. Even in cases where the estimated positions of the tags do not reflect very well the ground truth, relative positions of such product clusters are still reasonable when looking at the visualization. Using algorithms that also leverage differences in RSSI values, I was able to achieve similar, for some experiments even better results.

Limitations

Despite promising results in localizing RFID tags based on time-differences in read events, my approaches have several limitations. While I can observe that results improve when redundant reads of RFID tags are available, adding more data tends to produce worse results. I hypothesize that there is too much noise in the RSSI value, which is influenced by a variety of factors, such as the rotation of the tag and surrounding metals. Moving with constant speed proved to be a requirement for my approaches to work.

Another limitation is that due to how I estimate distances between RFID tags, the sequence in which RFID tags are read is decisive for the distance estimation. It is important to create overlaps between tag reads, especially it is required that the tags that were read at the beginning of the experiment are again read at the end. Results of random walk experiments highlight this fact, as the error between estimated positions and ground truth is relatively high.

Nevertheless, as demonstrated in this work, my suggested approach yields promising first results, warranting further investigations to evaluate its performance in real-world retail applications.

7.2 Future Work

I will evaluate my approaches on real-world data collected from stocktakes of customers of Detego. Although there is no ground truth to test the estimations against, I can still inspect the visualization to see if I can recognize the structure of the store and clusters of articles. Examining closely clustered items for outliers (i.e., items of different articles) could represent a first starting point for detecting and minimizing misplaced items.

Due to the fact that I do not know the position of the RFID reader, my approaches are highly dependent on the path through the store that the staff takes while reading the tags.

In future work I would try to tackle this shortcoming by estimating the position of the reader, before refining the distance estimations of the tags. There is a variety of additional information that can be included for the localization of the RFID reader such as, signal strength of wifi networks, and the data of inertial sensors of the smartphone. Many retailers have wifi networks in their stores. When there are multiple wifi networks available (e.g., from neighboring stores) it is possible to perform trilateration to estimate the position of the reader. Using the data from sensors of the smartphone such as, gyroscope, compass, and the camera, could also help with localizing the reader as well as the tags.

Finally, using the data of several consecutive stocktakes could also increase the accuracy of the distance estimations. Items that were close to each other yesterday, are more likely to be close together today as well.

Bibliography

[ACV06] C. Alippi, D. Cogliati, and G. Vanini. “A statistical approach to localize passive RFIDs.” In: *2006 IEEE International Symposium on Circuits and Systems*. May 2006, 4 pp.-. DOI: 10.1109/ISCAS.2006.1692717 (cit. on p. 21).

[AP04] Joshua Ash and Lee Potter. “Sensor network localization via received signal strength measurements with directional antennas.” In: *Proceedings of the 2004 Allerton Conference on Communication, Control, and Computing*. 2004, pp. 1861–1870 (cit. on p. 17).

[Bow85] Lord Bowden. “The story of IFF (Identification Friend or Foe).” In: *IEE Proceedings A - Physical Science, Measurement and Instrumentation, Management and Education - Reviews* 132.6 (Oct. 1985), pp. 435–437. ISSN: 0143-702X. DOI: 10.1049/ip-a-1.1985.0079 (cit. on p. 6).

[BP00] Paramvir Bahl and Venkata N Padmanabhan. “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*. Vol. 2. Ieee. 2000, pp. 775–784 (cit. on p. 19).

[BP09] M. Bouet and G. Pujolle. “L-VIRT: Range-free 3-D localization of RFID tags based on topological constraints.” In: *Computer Communications* 32.13 (2009), pp. 1485–1494. ISSN: 0140-3664. DOI: https://doi.org/10.1016/j.comcom.2009.05.007. URL: http://www.sciencedirect.com/science/article/pii/S014036640900111X (cit. on p. 21).

[DKA07] Konstantinos Domdouzis, Bimal Kumar, and Chimay Anumba. “Radio-Frequency Identification (RFID) applications: A brief introduction.” In: *Advanced Engineering Informatics* 21.4 (2007), pp. 350–355 (cit. on p. 1).

[GR06] Simon Garfinkel and Beth Rosenberg. *RFID: Applications, security, and privacy.* Pearson Education India, 2006 (cit. on p. 1).

[GSB07] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. “Improved techniques for grid mapping with rao-blackwellized particle filters.” In: *IEEE transactions on Robotics* 23.1 (2007), pp. 34–46 (cit. on p. 22).

[Har+02] Andy Harter et al. “The anatomy of a context-aware application.” In: *Wireless Networks* 8.2/3 (2002), pp. 187–197 (cit. on p. 21).

[HB01] Jeffrey Hightower and Gaetano Borriello. “Location systems for ubiquitous computing.” In: *Computer* 34.8 (2001), pp. 57–66 (cit. on pp. 20, 21).

[He+05] Tian He et al. “Range-free localization and its impact on large scale sensor networks.” In: *ACM Transactions on Embedded Computing Systems (TECS)* 4.4 (2005), pp. 877–906 (cit. on p. 20).

[Hek+10] C. Hekimian-Williams et al. “Accurate localization of RFID tags using phase difference.” In: *2010 IEEE International Conference on RFID (IEEE RFID 2010).* Apr. 2010, pp. 89–96. DOI: 10.1109/RFID.2010.5467268 (cit. on p. 21).

[HWB00] Jeffrey Hightower, Roy Want, and Gaetano Borriello. “SpotON: An indoor 3D location sensing technology based on RF signal strength.” In: (2000) (cit. on p. 21).

[JPB09] Dominik Joho, Christian Plagemann, and Wolfram Burgard. “Modeling RFID signal strength and tag detection for localization and mapping.” In: *Robotics and Automation, 2009. ICRA’09. IEEE International Conference on.* IEEE. 2009, pp. 3160–3165 (cit. on p. 22).

[Kon+04] Petri Kontkanen et al. “Topics in probabilistic location estimation in wireless networks.” In: *Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. 15th IEEE International Symposium on.* Vol. 2. IEEE. 2004, pp. 1052–1056 (cit. on p. 19).

[Kru+00] John Krumm et al. “Multi-camera multi-person tracking for easyliving.” In: *Visual Surveillance, 2000. Proceedings. Third IEEE International Workshop on.* IEEE. 2000, pp. 3–10 (cit. on p. 21).

[Liu+07] H. Liu et al. “Survey of Wireless Indoor Positioning Techniques and Systems.” In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 37.6 (Nov. 2007), pp. 1067–1080. ISSN: 1094-6977. DOI: 10.1109/TSMCC.2007.905750 (cit. on p. 17).

[Lui+11] G. Lui et al. “Differences in RSSI readings made by different Wi-Fi chipsets: A limitation of WLAN localization.” In: *2011 International Conference on Localization and GNSS (ICL-GNSS)*. June 2011, pp. 53–57. DOI: 10.1109/ICL-GNSS.2011.5955283 (cit. on p. 10).

[Ni+03] Lionel M Ni et al. “LANDMARC: indoor location sensing using active RFID.” In: *Pervasive Computing and Communications, 2003.(PerCom 2003). Proceedings of the First IEEE International Conference on*. IEEE. 2003, pp. 407–415 (cit. on pp. 19, 21).

[NN03] Dragos Niculescu and Badri Nath. “Ad hoc positioning system (APS) using AOA.” In: *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*. Vol. 3. Ieee. 2003, pp. 1734–1743 (cit. on p. 18).

[NRW06] Badri Nath, Franklin Reynolds, and Roy Want. “RFID technology and applications.” In: *IEEE Pervasive Computing* 1 (2006), pp. 22–24 (cit. on p. 1).

[Pat+03] Neal Patwari et al. “Relative location estimation in wireless sensor networks.” In: *IEEE Transactions on signal processing* 51.8 (2003), pp. 2137–2148 (cit. on p. 17).

[PCB00] Nissanka B Priyantha, Anit Chakraborty, and Hari Balakrishnan. “The cricket location-support system.” In: *Proceedings of the 6th annual international conference on Mobile computing and networking*. ACM. 2000, pp. 32–43 (cit. on p. 21).

[PHU+09] Ambili Thottam Parameswaran, Mohammad Iftekhar Husain, Shambhu Upadhyaya, et al. “Is RSSI a reliable parameter in sensor localization algorithms: An experimental study.” In: *Field failure data analysis workshop (F2DA09)*. Vol. 5. IEEE. 2009 (cit. on p. 11).

[RDT01] Ananth Raman, Nicole DeHoratius, and Zeynep Ton. “Execution: The Missing Link in Retail Operations.” In: *California Management Review* 43.3 (2001), pp. 136–152. DOI: 10.2307/41166093. eprint: https://doi.org/10.2307/41166093. URL: https://doi.org/10.2307/41166093 (cit. on p. 1).

[Sav+14] A. A. Savochkin et al. “Passive RFID system for 2D indoor positioning.” In: *2014 20th International Conference on Microwaves, Radar and Wireless Communications (MIKON)*. June 2014, pp. 1–3. DOI: 10.1109/MIKON.2014.6899990 (cit. on p. 21).

[Sch66] Peter H Schönemann. “A generalized solution of the orthogonal procrustes problem.” In: *Psychometrika* 31.1 (1966), pp. 1–10 (cit. on p. 34).

[SHC07] Jongchul Song, Carl T Haas, and Carlos H Caldas. “A proximity-based method for locating RFID tagged objects.” In: *Advanced Engineering Informatics* 21.4 (2007), pp. 367–376 (cit. on p. 20).

[SHS01] Andreas Savvides, Chih-Chieh Han, and Mani B. Strivastava. “Dynamic Fine-grained Localization in Ad-Hoc Networks of Sensors.” In: *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*. MobiCom ’01. Rome, Italy: ACM, 2001, pp. 166–179. ISBN: 1-58113-422-3. DOI: 10.1145/381677.381693. URL: http://doi.acm.org/10.1145/381677.381693 (cit. on p. 17).

[SN11] S. S. Saab and Z. S. Nakad. “A Standalone RFID Indoor Positioning System Using Passive Tags.” In: *IEEE Transactions on Industrial Electronics* 58.5 (May 2011), pp. 1961–1970. ISSN: 0278-0046. DOI: 10.1109/TIE.2010.2055774 (cit. on p. 21).

[SS02] SN Simic and Shankar Sastry. *Distributed localization in wireless ad hoc networks*. Tech. rep. Technical Report UCB/ERL, 2002 (cit. on p. 20).

[Wan+92] Roy Want et al. “The active badge location system.” In: *ACM Transactions on Information Systems (TOIS)* 10.1 (1992), pp. 91–102 (cit. on p. 21).

[Wan06] Roy Want. *RFID Explained:A Primer on Radio Frequency Identification Technologies*. Morgan & Claypool, 2006, pp. 94–. ISBN: 9781598291094. DOI: 10.2200/S00040ED1V01200607MPC001. URL: https://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6813362 (cit. on pp. 7, 8, 11, 13).

[Wei05] Ron Weinstein. "RFID: a technical overview and its application to the enterprise." In: *IT professional* 7.3 (2005), pp. 27–33 (cit. on p. 1).

[Wu+09] Dong-Liang Wu et al. "A brief survey on current RFID applications." In: *Machine Learning and Cybernetics, 2009 International Conference on*. Vol. 4. IEEE. 2009, pp. 2330–2335 (cit. on p. 1).

[WWB80] P E Watson, I D Watson, and R D Batt. "Total body water volumes for adult males and females estimated from simple anthropometric measurements." In: *The American Journal of Clinical Nutrition* 33.1 (1980), pp. 27–39. DOI: 10.1093/ajcn/33.1.27. eprint: /oup/backfile/content_public/journal/ajcn/33/1/10.1093_ajcn_33.1.27/4/27.pdf. URL: http://dx.doi.org/10.1093/ajcn/33.1.27 (cit. on p. 10).