



**University of
Zurich^{UZH}**

Ultra-high Frequency Passive RFID Identification and Visualization

*Yang Menz
Zürich, Switzerland
Student ID: 187-232-70*

Supervisor: Prof. Dr. Burkhard Stiller, Dr. Bruno Rodrigues,
Simon Tuck, Dr. Eder Scheid
Date of Submission: Oct 10, 2021

Abstract

Abstract As people spend most of their time indoors, Indoor Positioning Systems (IPS) have extended their use from object tracking to human motion detection. Among a variety of IPS technologies, Radio Frequency Identification (RFID), a non-contact electromagnetic signal for automatic object identification, is often chosen due to its inexpensive nature and ability to promptly identify a number of distinct tags. However, unlike objects under traditional tracking setups, people typically move and act arbitrarily, which leads to difficulties in localization, especially as traditional measures, such as Received Signal Strength (RSS), are noisy. In this thesis, a novel data collection method, which integrates a 3-D camera and a static RFID reader to refine reading accuracy and obtain RFID tag measurements, is introduced. In addition, Kalman Filtering is proposed to smooth a tag's location estimation. Moreover, this thesis elevates current declarative visualization of RFID-marked objects with defined metrics and key performance indicators (KPIs) and integrates them in a prototype dashboard for enhanced understanding in a practical context. The upgraded data collection, simple localization algorithm and additional visualizations aim to bridge semantic gaps for a RFID indoor application and offer campaign marketers and health measure administrators an analytical tool to effectively leverage collected measurements.

Keywords: Indoor Positioning Systems (IPS), RFID hybrid system, passive UHF RFID, offline customer behavioral analysis, real-time customer insights, offline customer success KPIs & metrics, privacy preserving tracking, visual performance dashboard

Da die Menschen die meiste Zeit in Innenräumen verbringen, haben Indoor Positioning Systems (IPS) ihre Anwendung von der Objektverfolgung auf die Erkennung menschlicher Bewegungen erweitert. Unter einer Vielzahl von IPS-Technologien wird die Radiofrequenz-Identifikation (RFID), ein berührungsloses elektromagnetisches Signal zur automatischen Identifizierung von Objekten, häufig gewählt, da sie kostengünstig ist und eine Reihe von verschiedenen Tags rasch identifizieren kann. Im Gegensatz zu Objekten in herkömmlichen Tracking-Konfigurationen bewegen sich Menschen jedoch typischerweise willkürlich, was zu Schwierigkeiten bei der Lokalisierung führt, zumal herkömmliche Messgrößen wie die empfangene Signalstärke (RSS) von Messrauschen betroffen sind. In dieser Arbeit wird eine neuartige Datenerfassungsmethode vorgestellt, die eine 3-D-Kamera und ein statisches RFID-Lesegerät kombiniert, um die Lesegenauigkeit zu verbessern und RFID-Tag-Messungen zu erhalten. Darüber hinaus wird eine Kalman-Filterung vorgeschlagen, um die Schätzung der Position eines Tags zu glätten. Des Weiteren wird in dieser Arbeit die derzeitige deklarative Visualisierung von RFID-gekennzeichneten Objekten mit definierten Metriken und Leistungsindikatoren (Key Performance Indicators, KPIs) erweitert und in ein prototypisches Dashboard integriert, um das Verständnis in einem praktischen Kontext zu verbessern. Die verbesserte Datenerfassung, der einfache Lokisierungsalgorithmus und die zusätzlichen Visualisierungen zielen darauf ab, semantische Lücken für eine RFID-Indoor-Anwendung zu schliessen und den Vermarktern von Kampagnen und den Verwaltern von Gesundheitsmassnahmen ein Analysewerkzeug zu bieten, um die gesammelten Messungen effektiv zu nutzen.

Acknowledgments

I want to thank Prof Dr. Burkhard Stiller for offering this unique research topic and allowing me to work at the Communication Systems Group Lab.

I also wish to express appreciation to my supervisor Dr. Bruno Rodrigues, for his patience and dedication in providing advice and support over the course of the master thesis. His guidance was always helpful whenever the code does not run or the elaboration of a concept is not to the point.

I am particularly grateful to Mr Simon Tuck from Livealytics. His contribution was significant to success of the thesis. On behalf of Livealytics, he provided the user scenarios, hardware and a testing space.

My appreciation goes out to my colleague Ile Cepilov who I worked together with and helped me during the thesis. Finally, I would like to thank my families for being there for me throughout all my master studies. I could never have done it without you.

Yang Menz

Contents

Abstract	i
Acknowledgments	iii
1 Introduction	1
1.1 Motivation	2
1.2 Thesis Goals	3
1.3 Description of Work	4
1.4 Thesis Outline	5
2 Background & Related Work	6
2.1 Background	6
2.2 Related Work	9
2.2.1 System Design	9
2.2.2 Localization Methods	10
2.2.3 Localization Algorithm	12
2.2.4 Data Visualization	13
2.2.5 Existing Projects	14
3 System Overview	15
3.1 Design of System Architecture	15
3.2 RFID Reader Configuration	16
3.3 Implementation of System Architecture	18

4	Data Management	21
4.1	Data Collection	21
4.2	Localization Algorithm	24
4.2.1	Kalman Filter	24
4.2.2	Interpolation	25
4.2.3	Specification and Parameterization	27
4.2.4	Results	31
4.3	Performance Measures	32
4.4	Data Visualization	35
4.5	Data Privacy	37
5	Evaluation	41
5.1	Kalman Filtering	41
5.2	Usability of Prototype	41
5.3	Application Integration	42
6	Final Considerations	44
6.1	Future Work	45
	Bibliography	46
	Glossary	53
	Appendices	53
A	Installation Guidelines for Dash Dashboard	54

B	Contributory Facts	55
B.1	Contents of The Hand-in	55
B.2	Additional Tables	55
B.3	Codes	56
B.3.1	Data Preprocessing	56
B.3.2	Data Processing	57
B.3.3	Kalman Filtering	61
B.3.4	Dash Dashboard	64
B.4	Simulation Description	69

List of Figures

2.1	RFID mechanism [27]	9
3.1	System architecture	16
3.2	Impinj RF modes flow chart [50]	17
3.3	Tags layout self-illustration	18
3.4	Tag detection heatmap under the defined reader modes	19
3.5	Tag detection heatmap under the remaining defined reader modes	19
3.6	Mockup of the GUI display	20
4.1	Tag distribution within defined zones	23
4.2	Distance distribution per tag	24
4.3	Unfiltered data for tag 'R' on 15 July 2021	25
4.4	Plot of time intervals of raw data from 0-500 (left plot) and from 0-200 (right plot)	26
4.5	Plot of time intervals of raw data and cleaned data from 0-500 (left plot) and from 0-200 (right plot)	27
4.6	Comparison of unfiltered and (Kalman) filtered values for x and y coordinates	31
4.7	Comparison of unfiltered and (Kalman) filtered values for x and y coordinates in the time interval [600:100]	32
4.8	Schematic overview of the different zones	34
4.9	Tag distribution within defined zones of interest	36
4.10	Dashboard centered as merchant A on July 1 between 09.00 - 14.00 (A) Visibility Rate (B) Engagement Rate (C) Interaction Rate	38
4.11	Dashboard centered as merchant B on July 15 between 10.00 - 14.00 (A) Visibility Rate (B) Engagement Rate (C) Interaction Rate	39

4.12 Dashboard centered as merchant A on June 16 between 13.00 - 16.00 (A) Visibility Rate (B) Engagement Rate (C) Interaction Rate	40
--	----

List of Tables

2.1	Comparison of major IPSs ([22]–[25])	8
4.1	Flux query output	22
B.1	Tag EPC-label mapping	55

Chapter 1

Introduction

Unlike in E-commerce, the implementation of customer journey and analytics is not straightforward in physical retail settings. For example, Indoor Positioning Systems (IPS) used for customer journey realization has many open questions. One challenge is the choice of device. Although there are a variety of readily available devices to perform indoor positioning in the current ubiquitous computing field, there are serious implications when location privacy is lost [1]. It is essential to avoid tracking methods that require consumers to actively interact with a product or service because their identity (e.g., documents or face) is profiled, and privacy could therefore not be preserved.

This thesis concerns the analysis and visualization of indoor capacity and people's flow, considering the need for social distancing measures and the need for customer journey analytics in physical retail setting, supported by technical solutions based on Radio-Frequency IDentification (RFID). Livealytics provides a hardware solution that includes Monza R6 based passive tags and an Impinj xArray RFID R680 reader [2]. With the infrastructural support from Livealytics, the study wishes to improve tag localization and turn the data into visualization that serve as visual aids for applications like indoor marketing, indoor social networking, and indoor social distancing monitoring.

The project's relevance can be explained as follows: Firstly, passive RFID tags have many advantages, making them attractive to the market. Previous research also listed many advantages of RFID technology. Among them, one winning characteristic is that the

technology is inexpensive and resistant to hostile environments [3]. RFID technology also wins over other cost-effective location tracking technologies such as Bluetooth and Wi-Fi. For instance, the SwissCovid App adopted Bluetooth technology which requires users to enable devices to commence tracking ([4]–[6]). Secondly, RFID-based real-time tracking has its research and practical potential. The museum and event industry currently have sub-optimal systems for scheduling and locating visitors and on-site staff, resulting in poor resource and room utilization described in [7]. During the ongoing pandemic, additional rules have been imposed also on the Swiss museum industry to reduce health risks [4]. RFID-based real-time analytics could enable such optimization of room utilization, visitor flow and further ensure social distance. Thirdly, although RFID-based tracking techniques have already been used for companies like Munich-based Mandigo for sports timekeeping [8]. Similarly, Axper provides software for people counting and store analytics [9]. Despite these advancements, the development of combining RFID technology and live analytics is still at an early stage warranting further research in this area ([8], [10]–[14]).

1.1 Motivation

To react to the competition that is intensified by internet-based e-commerce, this study of integrating indoor passive RFID technology with visual data analysis can add value to the development of a user-centric and efficient IPS by accomplishing the following goals:

- **Simplify data collection**

Many passive RFID enable people counting studies have rather complex data collection and localization approaches: mobile RFID readers with landmarks [15], multilateration positioning with four static readers [16], a cell-based trilateral positioning [17], to name just a few. These approaches are highly dependent on the environment, require complex hardware and sometimes complex computation algorithms. Thus, they have often proven to have limited scalability.

- **Enhance the efficiency of data visualization**

Being a low level measurement, location coordinates can create visual ambiguities.

Therefore, it is necessary to complement this low level data with suitable analytical tools so that semantic gaps can be closed and meaningful information can be prepared to serve decision makers and other non-technical audiences [18]. Dashboards can provide effective visual information because the large amount of data can be incorporated in the dashboard via aggregated metrics [18], [19].

1.2 Thesis Goals

Driven by the description of the outlined work description, the following master thesis goals are required:

- **Engineering:** Demands the study of processing methods based on data generated by Livealytics passive RFID-embedded trackers that record moving individuals' paths within an indoor area. Further, it involves analyzing the data such as location and associated time from RFID readers/tags. Specifically, this goal can be achieved by (i) writing a python-based program the reader server can host, which ideally should be provided as a web-based prototype; (ii) developing key features that are in a modular manner; (iii) weighing the possibility of integrating the working prototype with other modules and produce results.
- **Research:** Involves building an analytical prototype that provides findings, including visualizations about visitor flow, and a behavioral map for indoor spaces. These can be broken down into the following aspects: (i) relates to the existing research findings to determine appropriate ultra-high frequency band, reasonable proximity between an RFID reader and RFID tags, retrieving relevant variables from Livealytics API, and preparing the well-formed input; (ii) summarises and evaluates methods that are adopted in the existing design process and prototype development for passive UHF RFID hybrid system; (iii) supplementary research support for the sections from motivation and problem description, background information, related work, to design decision and evaluation.

1.3 Description of Work

The purpose of this thesis is to develop a human-motion focused privacy preserving passive UHF RFID indoor tracking application for the use of e.g. live marketing campaigners. In general, two different challenges are tackled in this thesis: focusing on the research aspect and the engineering aspect.

The research aspect involves the issues related to improving the overall accuracy of people tracking through a system powered by RFID data, which requires the research of adequate methods to process and visualize collected data embedded into a prototype. To ensure a balance between research scope and quality, the key search words that are listed in the abstract are identified. Keeping in mind the thesis' objective, a number of research papers from major engineering journals and related books or conference proceedings are collected and reviewed. Setting aside literature that is outdated or less relevant, the final reference list is compiled.

The engineering aspect requires the development of tools to handle the captured information from the RFID tracker through APIs provided by the manufacturer of the RFID devices and the incorporation and analysis of this information in the prototype. The engineering part consists of overall project management as well as software development where software development can be further divided into design and coding. For project management, an iterative approach is followed based on an agile project management methodology to break down the thesis into more manageable tasks completed in weekly deliverables. Trello, a popular web-based project management tool by Atlassian is used to plan & organise. In terms of design Figma, a web-based vector graphics editor and prototyping tool, is used to develop the web app UI and evaluate the design mockup. For coding, the Dash framework is employed. Dash is an open-source software development kit by Plotly that allows a full-stack web app development with interactive data visualization based on Python code.

1.4 Thesis Outline

This thesis is organized as follows: **Chapter 2** provides an overview of papers relevant to this thesis. In **Chapter 3** the overall setup of the system is described both conceptually and with respect to the particular implementation used in this project. **Chapter 4** then describes how the raw data is collected, processed and turned into the prototype for displaying the visualizations. In **Chapter 5** the obtained results are briefly discussed and evaluated, before **Chapter 6** concludes with final considerations for this thesis and an outlook for possible future work.

Chapter 2

Background & Related Work

This Chapter provides background on relevant concepts and presents related works in the field of RFID tracking. Scientific works presented in the sections below tackle the following problem: how can a low-cost indoor positioning technology like RFID effectively locate many items.

2.1 Background

In the first part of this chapter, crucial concepts are introduced, which build a basis for further methods.

The first concept is consumer behavioral analysis. As part of consumer psychology, consumer behavioral analysis is fundamental to unveil positive and negative aspects of products and services ([20]). For example, the study of movement, capacity, and engagement can be essential for the strategic planning of live marketing initiatives in retail spaces, trade shows, and promotional activities. Unlike E-commerce, digital products or services offered via the Web (e.g., e-commerce), such extraction of information is relatively easy to perform, i.e. by tracking each movement (click) of consumers in a website and extracting KPIs based on his/her browsing behavior. However, such analysis is not only necessary for the strategic planning of live marketing initiatives in retail areas, trade shows, and promotional activities, but it is increasingly essential during acute health crises such as

that of COVID-19, where social distancing and safe occupancy rates are a significant concern. The objective of this thesis is to explore the possibility to estimate a customer's interest in a product based on the number of times a given customer was read by a RFID reader and when a customer visited such a merchant.

The second concept is localization services. Localization systems can be grouped into outdoor and indoor positioning systems. The Global Positioning System (GPS) is dominant among outdoor positioning systems. GPS consists of three segments: space, control, and user. The space segment includes 24 satellites that transmit one-way signals; the worldwide monitor and control station make up the control segment and are responsible for satellite maintenance; the GPS receiver that is part of the user segment receives the signal from GPS satellites and converts the transmitted information to three-dimensional position and time. While GPS dominates the outdoor positioning systems, there is no similar dominance in IPS. A variety of systems, including light, sound, sensor, and radiofrequency technologies, exist to provide localization in indoor surroundings. An IPS typically consists of two components: the technology for information collection (hardware) and the software to estimate distance and calculate the position. For example, an RFID indoor positioning system consists of transponders (i.e., tags) and a reader, which enable the information collection, and software where the data is processed [21].

The third concept is the RFID system. We chose the RFID system firstly because of the attractiveness of its lower cost and technical potential [15]. Secondly, as shown in Table 2.1, the RFID system has a clear advantage over Bluetooth in deployment cost (competitive tag cost), security (given one-way communication), and long-term maintenance cost. Within the RFID domain, there are different types of RFID tags, namely, passive and active tags. The activeness refers to the need for a power source. In other words, an active tag can only be functional when a power source is supplied, whereas a passive tag can operate on its own without power [3]. Due to no power source, passive tags are less costly, have longer usage life, and come in smaller sizes.

Features	RFID	Bluetooth	Camera	proximity sensor
Read range & speed	1-5 ft, fast data transfer	20ft, fast data transfer	long	within 1 ft
Reader Cost \$	1500	10-70	1500	10
Tag Cost \$	0.10	20	NA	NA
Large-scale deployment	easy	possible via Bluetooth mesh	can get complex because multiple cameras need to be set up	easy
Communication	one-way	two-way	both	one-way
Security	most secured as checksum method is used	mostly sufficient, sensitive data transmission can be compromised	mostly secured if cameras are encrypted & password-protected	moderate, can be affected by the interaction with other sensors
Battery Life	Really long	2-5 Years	a few hours - 1 Year	NA

Table 2.1: Comparison of major IPSs ([22]–[25])

In the case of a passive tag, the illustration of the process is shown in Figure 2.1: an RFID antenna activates a tag by sending off radio frequency waves to the tag’s IC chip [26]. The RFID tags can further be distinguished into three categories by the frequency level during the information transmission: Ultra-High Frequency (UHF) tag, High-Frequency (HF) tag, and Low-Frequency (LF) tag. This difference comes from the strength of the backscattered signal of a tag’s chip that can reflect and responds to a reader. The reader sends the Unique Identifier (UID) of a tag to its corresponding software. Passive UHF RFID tag is chosen in this study given its long range read capability.

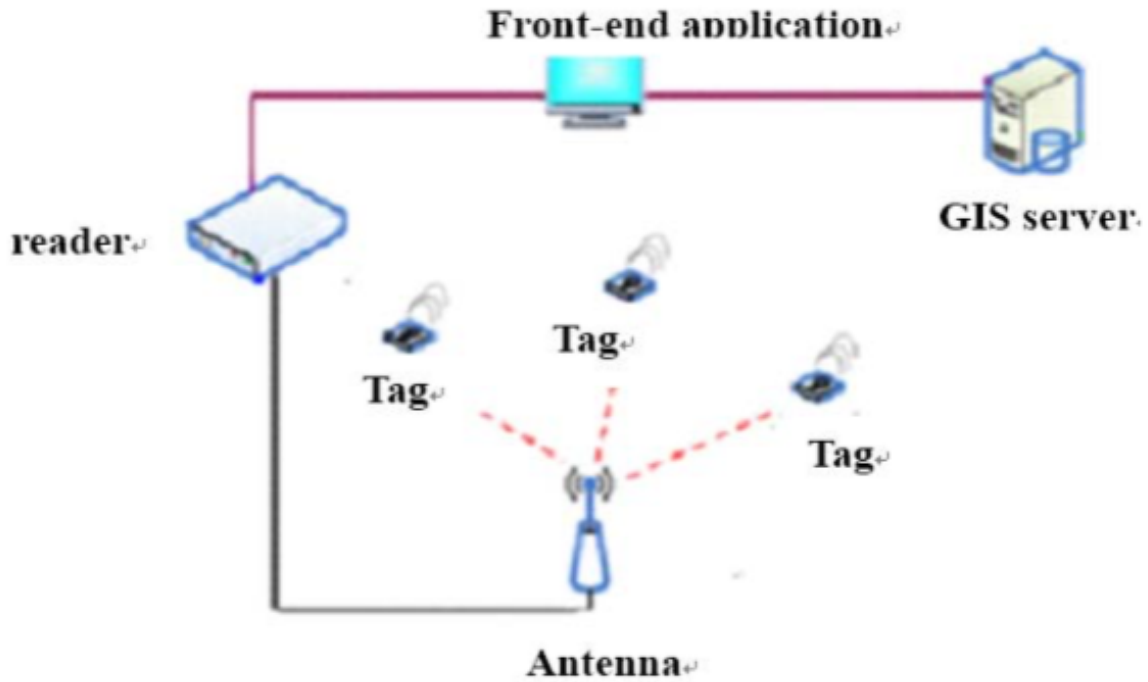


Figure 2.1: RFID mechanism [27]

2.2 Related Work

In this thesis, we investigate passive UHF RFID application with focus on crowd management. For this project the following topics are especially relevant: system design, localization methods and algorithms and data visualisation. This section provides a brief literature review for these topics.

2.2.1 System Design

There have been various papers dealing with practical execution of indoor positioning systems. These works provide valuable insights for the system design of this project. An early and particularly interesting paper [28] for example designed an indoor positioning system for blind people using floor tiles that were equipped with RFID readers. In this case the antenna and the reader were held by the person which needed to be located. By identifying the RFID tag ID and matching the tag ID to the pre-installed map it was then possible to determine the location of the person. The system was also equipped with WLAN, this however was not used for positioning as the accuracy of WLAN was not

deemed to be high enough. Some authors have tried to use WLAN for indoor positioning, but combined it with other technologies to improve the performance of the localisation. [29] for example combined WLAN with RFID technology which showed to improve both speed of the localisation as well as overall accuracy. There are also other hybrid approaches combining different technologies for indoor localization, all of which have the general goal to create synergies by compensating one technology's disadvantages with the advantages of some other technology. One study, for example, combined ultrasound and RFID making use of the ultrasound's high accuracy and the RFID's high communication capacity [30]. Another option is to combine RFID with visual information. Such a configuration was used by [31]. The combination of video images and active RFID tags allowed to authors to improve the accuracy. They achieved this by cross-referencing images with RFID position data so as to avoid time-consuming processing of images for identification purposes, which would have been needed if they had relied solely on image-based positioning. One drawback of this configuration is that it relied on active RFID tags, which are significantly more expensive than passive RFID tags. Therefore, [16] decided to use passive RFID tags in their study which they found to cost only 1/50 of their active counterparts. In their paper the authors demonstrated that passive RFID tags were able to achieve a high accuracy, albeit in a relatively small space. A further paper of interest is [32] as the authors work in a passive RFID setting in a 5m x 5m indoor space and used a reader with multiple antennas. They showed that with the setup all passive RFID tags could be detected.

2.2.2 Localization Methods

Despite the prevalence of RFID systems for localization purposes, the precise localization of objects or people remains challenging. This is partly due to the fact that RFID was initially designed for identification as opposed to localization [26]. As a result the location of an RFID tag cannot directly be observed but needs to be estimated which is challenging, especially for passive RFID tags as they are more susceptible to environmental conditions [33]. For UHF RFID systems the low signal to noise ratio (SNR) and backscatter modulation make obtaining precise locations even more difficult [34]. A variety of dif-

ferent localization methods exist that can be divided, for example, into range-based and range-free methods. Range-based methods are widely used and are based on the propagation characteristics of the RF signal. They can be further split into distance and angle measurements. To obtain the distance between the reader and the tag (distance measure) received signal strength indicator (RSSI), time of arrival/ time difference of arrival (TOA/TDOA) and phase of arrival (POA) can be used. Based on the obtained measures it is then possible to calculate the estimated tag location in a 2 dimensional space using geometric relationships. In a similar manner, angle measurements use the angle of the signal to determine the location of the tag. Due to the nature of range-based methods, i.e., that they use geometric relationships, it is generally necessary to have multiple readers and/or antennas to obtain location estimates. Furthermore the time based measures (e.g., TOA/TDOA) require precise (and expensive) hardware to measure the elapsed time. As electromagnetic waves travel with the speed of light, a small measurement error leads to large errors in distance estimates [34].

Apart from the discussed range-based methods, range-free methods can be used. One of these methods is fingerprinting, which is based on reference tags distributed in the indoor environment. For each reference tag the received signal strength indicator (RSSI) and phase are then saved in a database together with the known location. To localize a target tag with unknown position, the captured data of that tag is compared against the database and the location estimated using an algorithm such as k-nearest neighbor. A commonly and early used version of this approach is LANDMARC (e.g. [35]–[37], which relies on active RFID tags. However, similar concepts exist that use passive tags and are therefore more cost-efficient (e.g. [26]). There exist also range-free methods which do not require reference tags. One of these methods is based on captured phase sequence for moving targets to construct a spatial likelihood map, based on which trajectory and initial position can be inferred [34]. A few authors provide comprehensive reviews of localization methods, e.g., [33], [34], [38]–[41].

2.2.3 Localization Algorithm

Apart from the localization method (i.e. obtaining raw data), the localization algorithm (i.e. obtaining and refining location estimates based on the data) has been another focus of the discussion around indoor localization. This topic is highly relevant in practical applications as the raw data received from the sensors can be rather imprecise and noisy. The commonly used received signal strength (RSS), for example, suffers from attenuation in the presence of indoor obstacles and therefore results in accuracy in a metre unit [26]. Similarly, [21] showed that the distance error using raw RSSI data was not only relatively large but also had a high variance, i.e. the distance error varies considerably over time. Thus a processing of obtained raw data is generally required. A comprehensive review of available algorithms is provided by [33]. The following paragraphs aim to highlight a few important works for this project.

While many earlier works focus on methods in the context of active tags, there are also relevant papers that work with passive RFID tags. [26], for example, used a grid-based setup with known reference tag locations with passive tags. They used the known distances and angle of arrival (AOA) of reference tags to determine the location of an unknown tag. Their approach does not use a predefined number of reference tags for the positioning of the unknown tag, but a number that varies depending on how many reference tags are detected—as opposed to e.g. the LANDMARC system which used 4 nearest neighbours (and also relies on active tags). The results of different reference tags are then weighted in an effort to "smooth out" the position of the unknown tag. The results show that the approach significantly reduced the linear positioning error. It is, however, worth noting that this increased accuracy requires fairly frequent distribution of reference tags (i.e. every 1-3 meters in x and y direction), because when reference tags are less frequent there are fewer reference tags detected leading to inaccuracy.

To overcome the shortcomings of RSSI, [21] used a Kalman filter. In their setup a RFID reader would travel along a path with passive RFID tags along the side in known intervals. The raw data would then be processed by the specified Kalman filter (compensating also for the angle-dependent loss factor). In the experiment it was shown that with the employed approach measurement errors could be significantly reduced. The authors thus

concluded that standalone passive RFID tag systems can provide promising IPS. This is especially true, since the estimates were based on instantaneous RSSI measures making this configuration interesting to near real-time applications. It should be noted, however, that the authors used a Kalman filter that did not rely on the previous state (i.e. the previous location estimates), making it not applicable for navigational purposes (unlike the authors' earlier conceptual study on the topic [42]). Other authors have also employed Kalman filters in their approach. [43] use a Kalman filter for drift removal and note that it outperforms traditional statistic/probabilistic techniques in both speed and accuracy.[15] combined a probabilistic RFID map (based on landmarks) with a Kalman filter. They showed, albeit in simulations, that such a low complexity and low cost algorithm achieves accurate localization.

2.2.4 Data Visualization

There is a semantic gap between low-level measurements captured by IPS hardware and the demand from end-users for processed data with information value to, for example, make decisions based on them. Specifically, location coordinates, one type of low level measurements, can create visual ambiguities when reviewers of measurements have not been to the venue themselves [18] and can be difficult to interpret by itself. Another study also pinpoints that RFID systems have no self-reporting capabilities [44] even though the system employs advanced tracking which outperforms manual tracking by a factor of eight [45].

Besides having awareness of the existing gap in RFID people counting, it is important to understand and evaluate available scientific illustrations that can be used for location coordinates. Heatmaps and dashboards are used in one study, namely, heatmaps display the density of people for a given time and indoor area, whereas dashboard combines people density, people flow, camera and people distribution for a selected visited zone and presents them at once [18]. Another study favors bar chart as the visualization method. The charts include both specific and pivoted values. The difference in bar height allows the author to compare the recorded data across individuals (workers in this case), zone, functions by daily, weekly, or monthly frequency [44].

As dashboards can incorporate a great amount of data with varied type of visual representations via summarized analysis on visual form [19] and dashboards allow to automate the analysis of records for accurate assessment and prompt planning [44], they are viable candidates in the context of near real-time analytics.

2.2.5 Existing Projects

There are already projects, which deal with the topic of live marketing and RFID-enabled people counting. Here a selection of the corresponding papers is presented.

The first project is C-Counter 1.0, a web application developed under the Communication Systems Group at University of Zurich. It is also the basis for this thesis. C-Counter 1.0 uses hybrid hardware, that is, a 3-D camera and a passive UHF RFID system as its hardware. C-Counter 1.0 also has a software part—a customized API for distance calculation including collecting, aggregating, and analyzing. Visually, C-Counter 1.0 uses a heatmap displaying visitor density and line charts for evolution of visitor density. Although it has near real-time calculation features, the underlying data is unfiltered and advanced analytical visualizations are not incorporated [46].

The second project is a commercial solution offered by Mandigo. The Mandigo RFID solution includes both UHF and HF passive tags. According to Mandigo’s website, it claims that the solution provides full-spectrum tracking service from installing the on-premise RFID hardware including own licensed bibchip tags, enabling real-time data collection via the chip-embedded shopping cart or smart ticket to evaluating the collected data [8]. As there are not mentions regarding the integration of other IPS technologies, we can assume the inherent limitation of the RFID technology such as sensitivity to environments, signal attenuation due to metal surfaces, and signal bouncing [47] are still present in this solution.

Chapter 3

System Overview

The main idea of this thesis is to leverage the existing project effort and enhance its usability by adding exploratory visual features. The base of the current system architecture is the RFID system design described in [46]. The whole system is divided into two parts: parts responsible for data collection and parts used for data visualization. This chapter firstly presents the system components and their interactions. Secondly, it explains a set of relevant KPIs that provide contexts for the visualizations. Subsequently, it continues with the realization of the back-end design.

3.1 Design of System Architecture

As shown in Figure 3.1, the whole system is split into two parts: the data collection segment and the data visualization segment. Both segments have their front-end and back-end. Regarding the data collection segment, the back-end components are RFID reader and tags, camera, REST API, data processors. RFID readers, tags, and the camera can be grouped as hardware. Specifically, we deployed a static Impinj xArray R680 reader with linear antennas, passive Monza R6-P tags, and Xovis 3-D Camera, given the described advantages [46]. The remainings are software: namely, AWS and InfluxDB. Data produced by the camera is processed by AWS hosted lambda functions (Process Person Coordinates; Database Connection; Calculate Distances) and stored in AWS S3. Data made by RFID

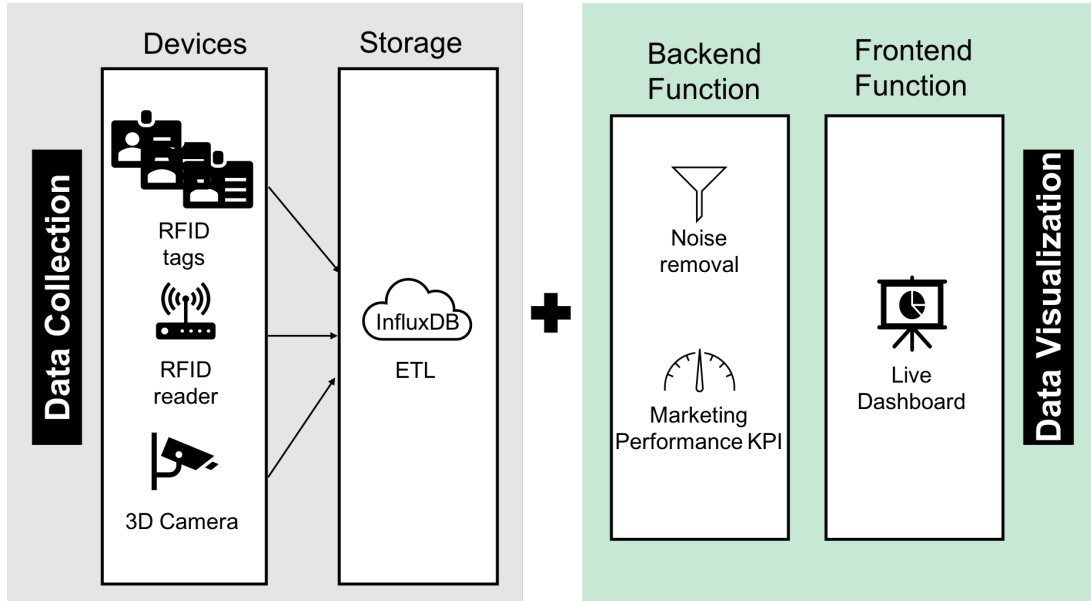


Figure 3.1: System architecture

reader is processed by InfluxDB hosted lambda functions and stored in InfluxDB and AWS S3. The API is an endpoint that connects the back-end and front-end. The API connection is triggered from the rule-based AWS Cloudwatch. The front-end Influx GUI only displays paired data filtered by another InfluxDB hosted lambda function Match Data. The back-end includes two python-based lambda functions (noise removal and KPI converter) for the data visualization part. The front-end is the dashboard displayed in the user's browser. The front-to-back end follows the Dash framework.

The original system cloud counter is coded in React javascript [46]. Considering the amount of integration effort, the familiarity with Python of the author, and the limited timeframe, the Dash framework is selected. Dash displays the following specific advantage: one, Dash is the low-code framework with a strong focus on data application; two, Dash is React for Python. Like React js, the Dash application has a tree-like component layout and is deployable in the web browser [48].

3.2 RFID Reader Configuration

The RFID reader plays a critical role in impacting data accuracy. The influence is driven by the reader configuration parameters, including reader mode, transmit power, session,

and RF Power Sweep [49]. One of the main configuration parameters, reader modes, defines how the reader antenna will transmit over-the-air data to the tag. One can vary reader mode to balance read rate and resistance to interference that best suits the needs of use cases.

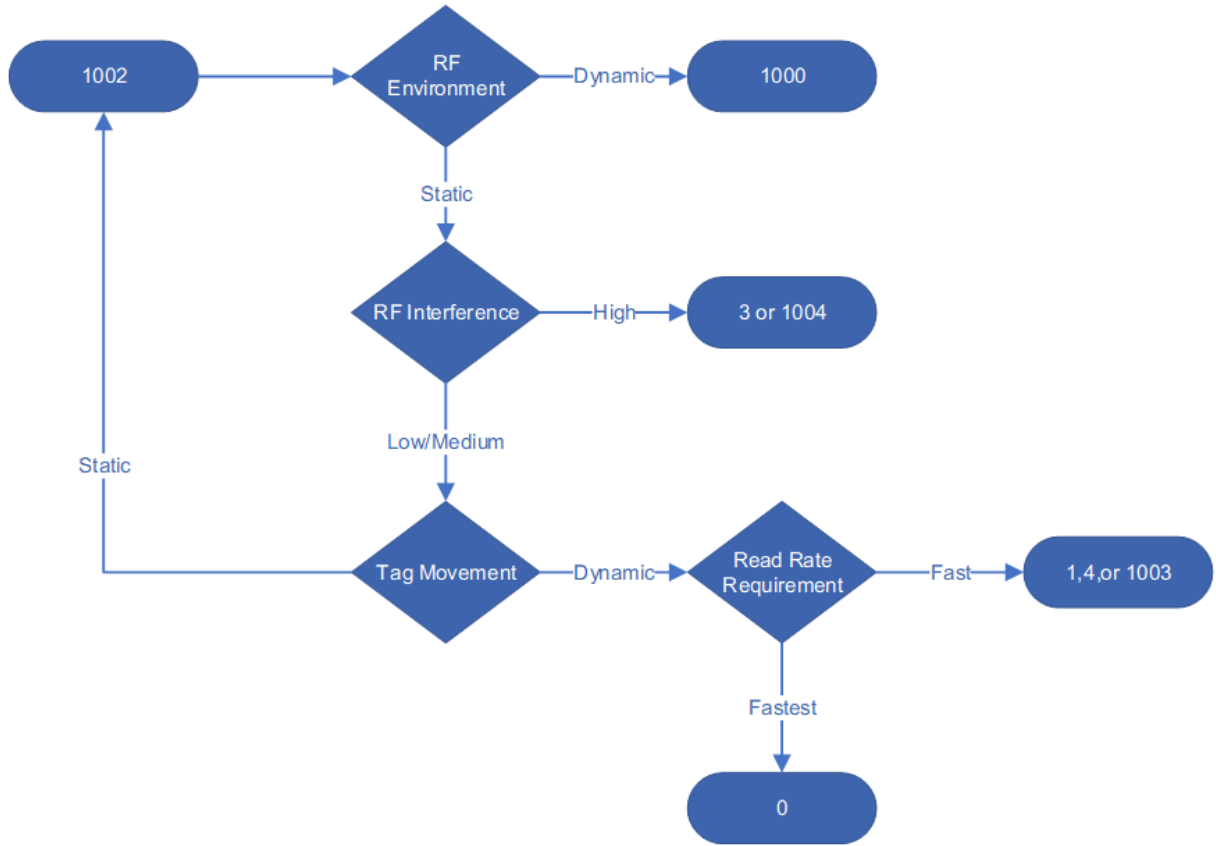


Figure 3.2: Impinj RF modes flow chart [50]

Given the flow diagram 3.2, the determining factors are RF environment, tag movement, and rate requirement. The targeted users are campaign marketers and event managers, so tagged attendees are likely moving around at varied paces. For example, people flow in a museum is probably slower than the crowd pace in an outdoor festival. To capture wide scope of scenarios, different reader modes were selected, such as **max throughput (0)**, **hybrid (1)** and **max miller (4)** according to the description in [50].

Fourteen tags are scattered around the $12m \times 12m$ test space (see illustration 3.3). The result is displayed in Figure 3.4. These heatmaps are populated based on the one-minute cumulative read frequency. Every square represents a tag. As the reader modes have their read rate (tag reads/second), and the experiment was conducted for one minute under

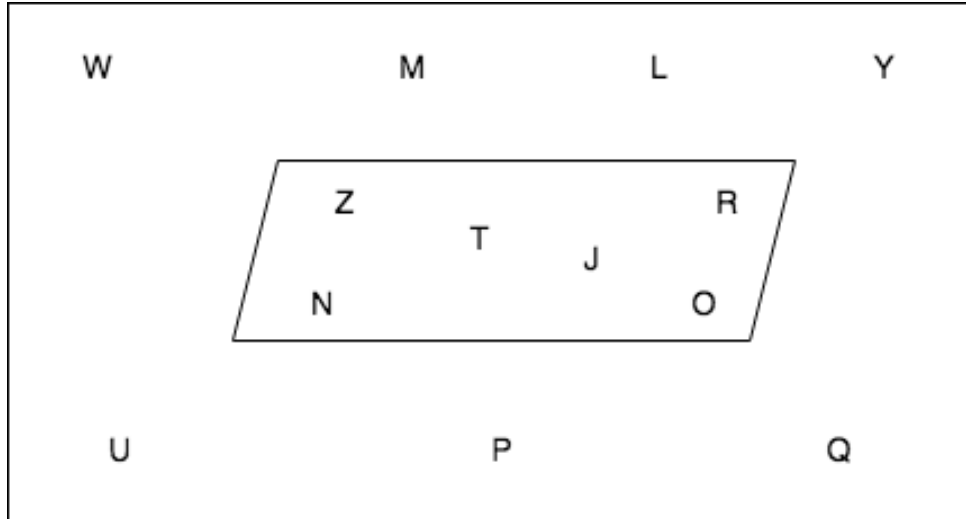


Figure 3.3: Tags layout self-illustration

every reader mode, it is evident that the one-minute cumulative read frequency can be used for comparing the suitability of reader modes. The brightness of the cube in the heatmap indicates how often the reader can detect tag. Both max throughput and max miller have a high read rate. The reader mode max throughput is the most susceptible to interference, whereas max miller has tolerance for multiple readers in the area. However, only one RFID reader is in the area; therefore, the heatmaps for max throughput and max miller display the same pattern. Hybrid has slightly lower unique tag reads, but its read range outperforms others as tags located 4 meters away from the reader antenna are still captured with reliable reads.

As the deployed Impinj reader R680 supports the available reader modes, the same one-minute cumulative read frequency testing applies to the remaining methods. The empirical results shown in the Figure 3.5 are consistent with the recommendation from Impinj - "the reader mode with the highest read rate will deliver the most accurate tag location calculation and the direction accuracy of a tags motion "[50].

3.3 Implementation of System Architecture

The system design described earlier has been implemented using scenario-based experiments. Several experiments were carried out with a set of hybrid hardware that consists

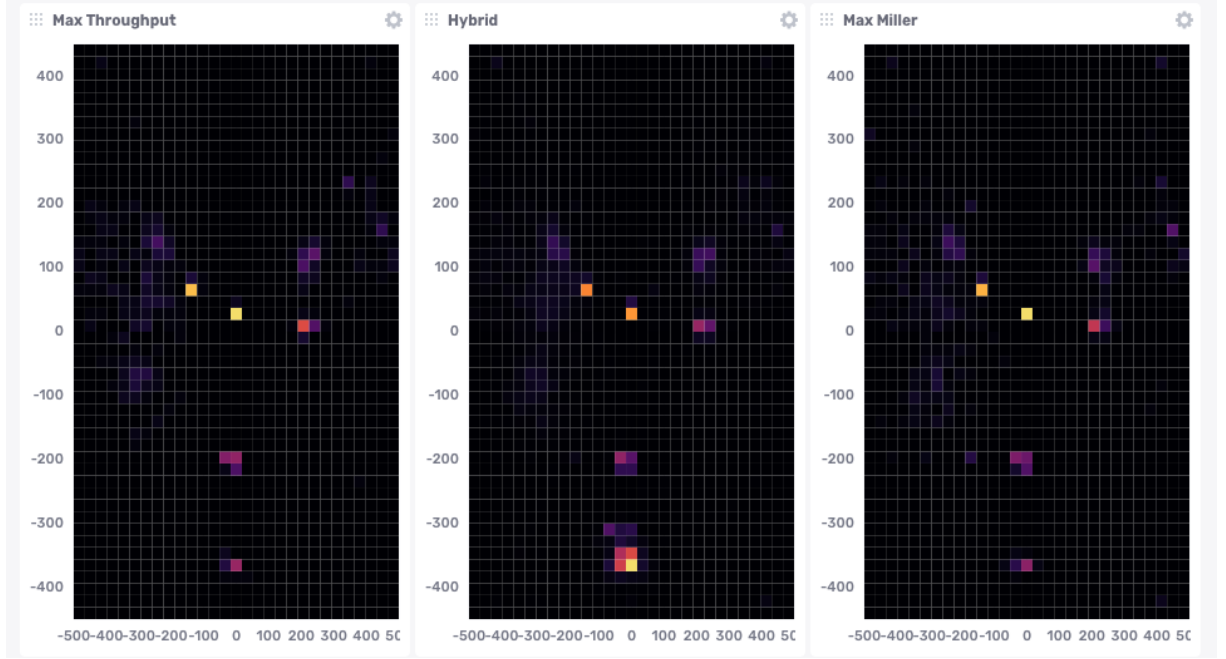


Figure 3.4: Tag detection heatmap under the defined reader modes

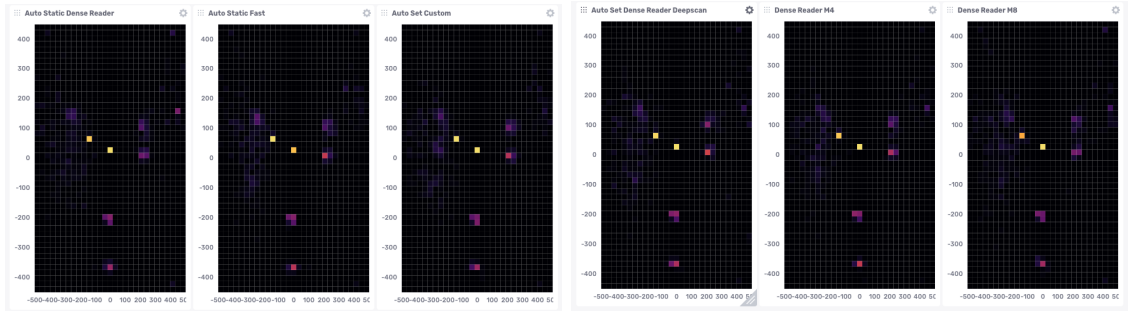


Figure 3.5: Tag detection heatmap under the remaining defined reader modes

of one ceiling-mounted Impinj xArray R680 reader and a 3-D Camera. The section 4 elaborate how this setup is essential to the elevated data collection and the effective data management.

At the initial stage, the hybrid hardware is turned on and connected. Specifically, the configuration of the RFID reader is customized as **hybrid** to minimize the fluctuation factors mentioned in the study [51]: time-varying factor (i.e., the multipath interference, environmental noise) and space-varying factor (i.e., the interaction of neighboring tags); and guarantee reliable reads even when multiple tags are dynamic and closely presented. The choice of reader mode is concluded from the performance comparison of different reader modes explained in the section 3.2. Given desirable tag choice, interruption-free environ-

ment, and detectable placement with appropriate between-tag spacing and configuration of the reader, we are confident that the read accuracy of the reader would remain in 95% – 99% [52]. Upon completing the system setup, the data transmission initiates, and InfluxDB captures this data and stores it for further processing.

The middleware running in the background provides a live feed of data to the upper layers. The primary job of the middleware is two-fold: noises removal and KPI calculation. Implementation of these core algorithms carries out a series of predication and calculations to improve reading accuracy and understand how tags interact in the defined zone of interest. The deep dive of core algorithms are presented in the section 4.2 and 4.3.

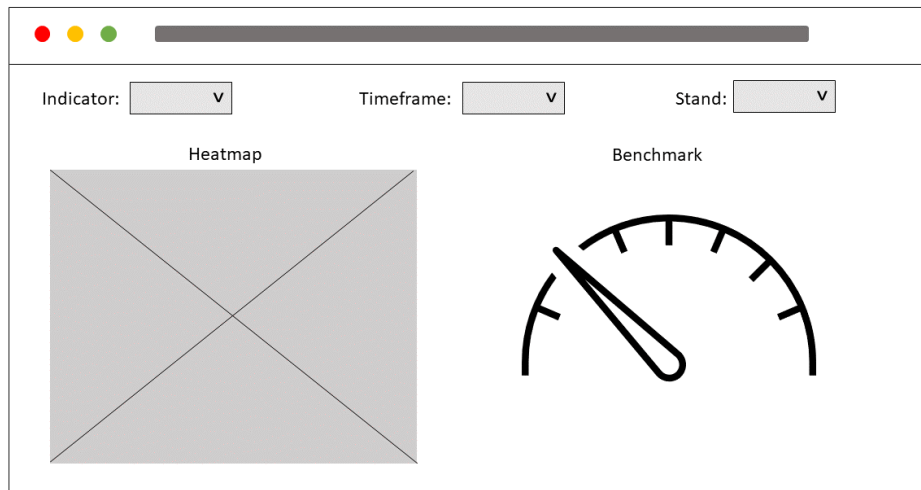


Figure 3.6: Mockup of the GUI display

The output is a dashboard consisting of these calculated KPIs (cf. Figure 3.6). Each field on the top line has a drop-down list. The area **Indicator** includes visibility, engagement rate, and interaction rate. Thus, the user can define the range of time in the field **time-frame**. The last area **Stand** includes all the merchants in the exhibition hall. Once the filters are set, the heatmap and benchmark are accordingly updated. The heatmap is time responsive and can be used for customer movement analysis, whereas the benchmark is Indicator responsive, which gives a quick competitor analysis of the selected merchant.

Chapter 4

Data Management

Data flow is the backbone of the project, so the main idea of this section is to explain the method and corresponding implementation of each step in a data management cycle.

4.1 Data Collection

A similar collection process described in the study [44] occurred in this thesis: the nearly real-time location data and auxiliary tag information such as tag electronic product code (EPC) are recorded and stored in the system. We define an object as an individual with a valid detectable RFID tag. Then we track an object by the EPC of each tag. For data generation, there are three testings conducted: on June 16 2021, July 1 2021, and July 15 2021. For each testing we simulate pre-defined scenarios (e.g. one or multiple testers walking across the room). Descriptions of the scenarios for each testing can be found in the appendix (Testing No.1, Testing No.2, Testing No.3). Once all the testing sessions are done, the records are retrieved using Flux query (see Listing 4.1). It is worth noting that (i) data is of good quality under the current system given Impinj readers have greater than 95% – 99% read accuracy¹ under ideal setting [52]; (ii) data transmitted to InfluxDB is

¹This is not to be confused with location accuracy. Read accuracy refers to the number of tags a system can successfully read versus the deployed tags, while location accuracy describes how well a provided location matches the actual location.

the matching coordinates that appear both in the camera scene and in the reader record, as indicated in the section 3.3.

```

1 from(bucket: "cloudcounter")
2 |> range(start: 2021-06-16T12:20:00Z, stop: 2021-07-15T09:24:25Z)
3 |> filter(fn: (r) => r["_measurement"] == "rfid_location")
4 |> pivot(rowKey: ["fillaleId", "readerName", "_time"], columnKey: ["_field"],
    valueColumn: "_value")

```

Listing 4.1: Flux query to data retrieval

As seen in the Table 4.1, the raw data produced by the Flux query has six columns: time, fillaleId, readerName, EPC as well as x, y coordinates. The first column 'time' is datetime64 type, formatted based on ISO 8601 protocol. ISO 8601 states that this grammar requires the "T" to avoid ambiguity and the "T" may be omitted under some circumstances. The second and third columns are the information about the reader. As we deploy only one reader in the simulation, these two columns would be ignored for the later engineering. The fourth column 'EPC' is string. The last two columns are integers, which are calculated automatically using RSSI values by Impinj xArray R680 reader.

time	fillaleId	reader Name	EPC	xCm	yCm
2021-06-16 12:20:09.858709799+00:00	schlieren	center	E280117000000 21428BFA201	0	-101
2021-06-16 12:20:19.597099141+00:00	schlieren	center	E280117000000 21428BF80B1	188	188
2021-06-16 12:20:30.474379692+00:00	schlieren	center	E280117000000 21428BF80B1	427	427
2021-06-16 12:20:42.609381998+00:00	schlieren	center	E280117000000 21428BF80C1	604	0

Table 4.1: Flux query output

To prepare for analyses, we need to assess the integrity of the full dataset first. To do that, an exploratory data analysis (EDA) was performed. Both codes 'Data Preprocessing' in B.3.1 and 'Data Processing' in B.3.2 are for the EDA purpose. The 'Data Preprocessing' focuses on data cleansing and formatting whereas the 'Data Processing' emphasizes data slicing and visualization. Beside understanding types of columns of full dataset, NA value screening was done but no empty row has found. For easy reference, each tag is assigned a letter (e.g. Z, Y, X) based on the EPC (see Table B.1 for the complete mapping). As

shown in Figure 4.1, the reader was able to read 13 tags. Tag 'V' rank the top, with 18'455 read frequency whereas tag 'W' rank the bottom while captured by the reader 1'607 times. This is however not necessarily an issue as the number of times a specific tag is read depends on the scenarios it was used in.

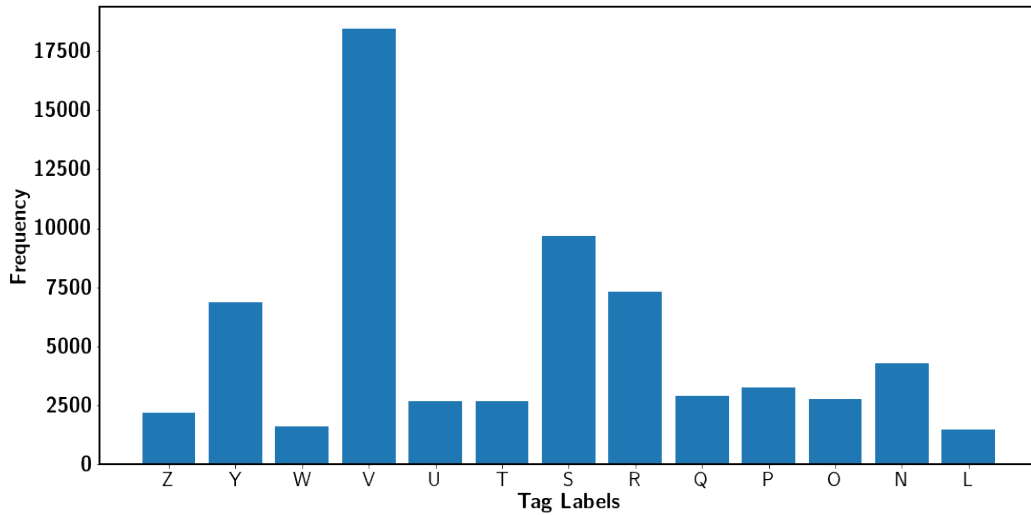


Figure 4.1: Tag distribution within defined zones

By running the code in the Appendix B.3.1, reshaped dataset has now distance column. This distance is euclidean distance between each tag position and center assuming the center coordinate (0,0). Using this calculated column 'distance', Figure 4.2 is made to describe how the tag movement distributed if the reader is seen as the center of testing area. From the Figure 4.2, we can observe that the tag 'V' frequently moves away from the center whereas the movement of tag 's' is also frequent but often close to the center.

The code "Data Preprocessing" finally cleans up the unused data entries and columns, and produces the dataset named "AllData.csv", which becomes input for coding Kalman filter and Dashboard.

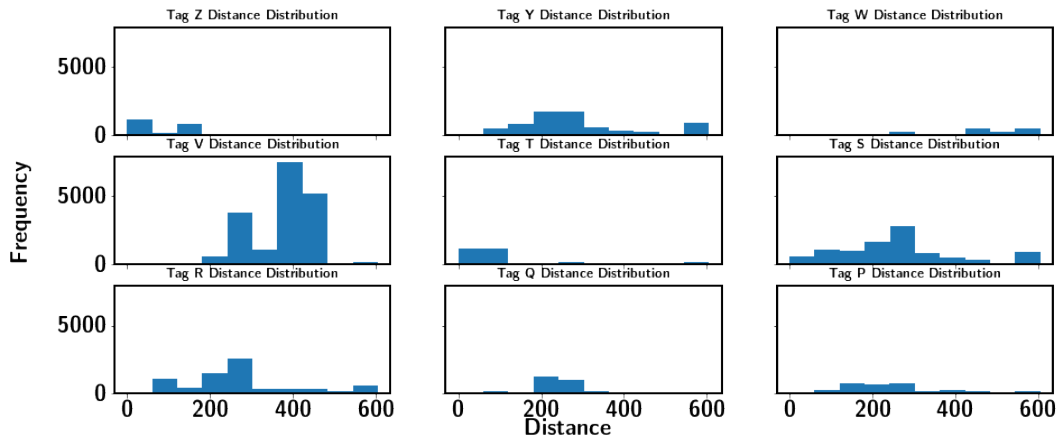


Figure 4.2: Distance distribution per tag

4.2 Localization Algorithm

Despite a sound system design, read anomalies can be contained in RFID measurements because the reader fails to read the tag in range, i.e., false negatives. The reader mistakenly records the tag that was not in range, i.e., false positives [44]. While this can be an issue, it should not be major in the used setup as we have previously outlined the high read accuracy of the Impinj component [52]. A more relevant issue is that even if the reader correctly identifies a tag, the obtained location data suffers from measurement errors, which can make data noisy. This is somewhat exacerbated by the fact that measurement intervals per tag are not fixed. To see this issue, we can look at some of the data. Figure 4.3 shows two plots (in x and y direction) for tag 'R' with time intervals in seconds on the x-axis. Three things are immediately visible: (i) there are, especially in the time interval $[0;200]$, large gaps, meaning that the tag was not read during these times; (ii) there are some rather significant jumps in both directions (contained in the interval $[0;600]$); (iii) even when the tag is perceived to be more or less static (interval $[600;1000]$, there appears to be some noise.

4.2.1 Kalman Filter

In order to smooth out the location data and thereby also improving the accuracy a filter is required. While there are many options (e.g. one could use a simple moving

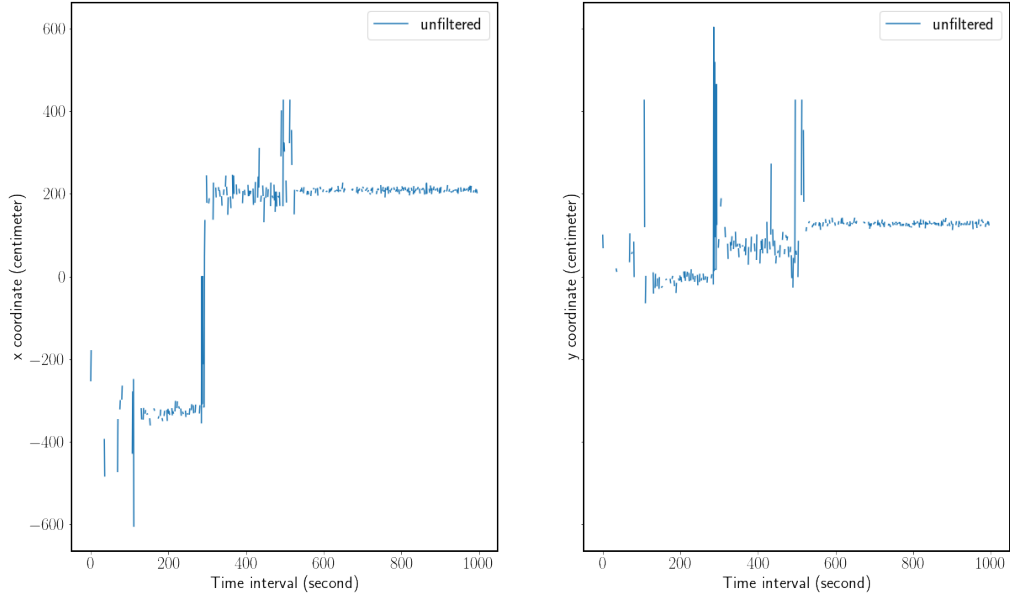


Figure 4.3: Unfiltered data for tag 'R' on 15 July 2021

average), the Kalman filter appears to be one of the most promising methods because it is suitable for near real-time applications: it does not require to have a large memory of past observations, it is recursive and it is quite fast. By contrast, a moving average would require a certain number of rolling past observations to be kept in memory for smoothing. Conceptually, the Kalman filter is also easy to understand. The idea is to predict the future state based on the current state, then use obtained measurements to update the state information. A formal derivation and detailed discussion can be found in the original paper [53] and a more accessible introduction in [54].

4.2.2 Interpolation

Before the Kalman filter can be applied, the measurement data needs to be transformed into the correct format. A key step is to make sure that the time difference (Δt) between measurement is equal across the dataset. We can explore this by plotting the time differences across the observations.

Figure 4.4 displays the magnitude of time intervals in seconds. In the left plot we can

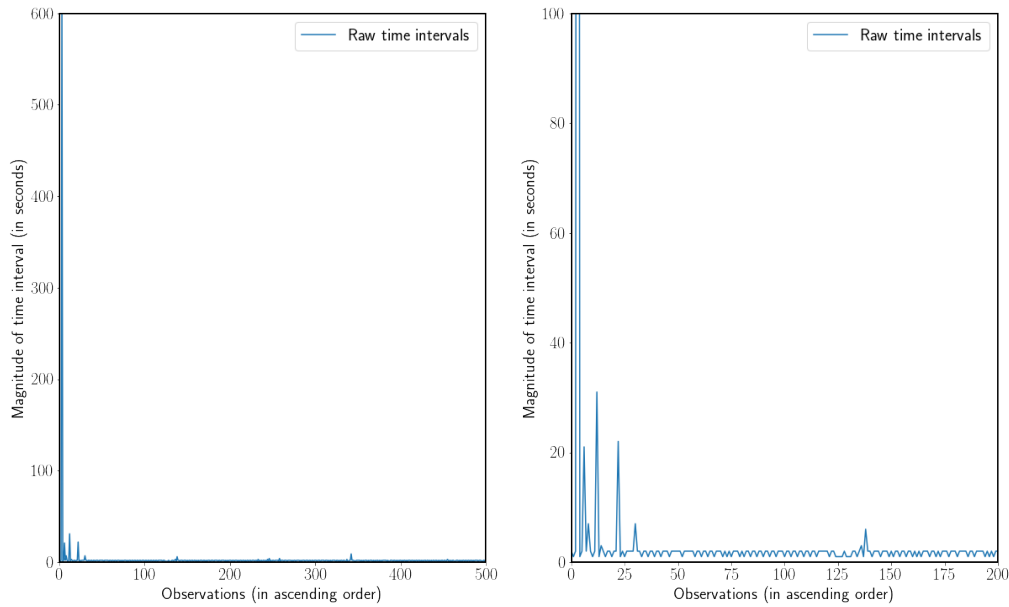


Figure 4.4: Plot of time intervals of raw data from 0-500 (left plot) and from 0-200 (right plot)

immediately see that a large time interval appears in the beginning of the observation period, while the rest of the observations seem to be more evenly spaced across time, although not exactly even.

In order to determine the correct time interval to use we can take the minimum of Δt and interpolate the rest of the dataset with NA values. This ensures that we do not lose any of the actual measurements. While this leaves us potentially with a large number of NA values, it is not an issue for the Kalman filter per se as it can handle NA values and will just assume that the value predicted with the Kalman filter is the measured value. Whether this is a useful assumption in practice is another question, especially if time intervals in the dataset are large. In our dataset the smallest time interval is 1 second and we therefore choose this as our Δt . In the data we might sometimes also come across duplicates and recording with the same time stamp but different locations. We deal with this as follows: The exact duplicates are removed while for multiple observations with the same time stamp we simply take the average x and y coordinates as we have no way of determining which observation is more accurate.

observations with large time intervals if they occur in the beginning of the observation period as this will not result in meaningful (filtered) values for that period. After doing so we can plot the time intervals again.

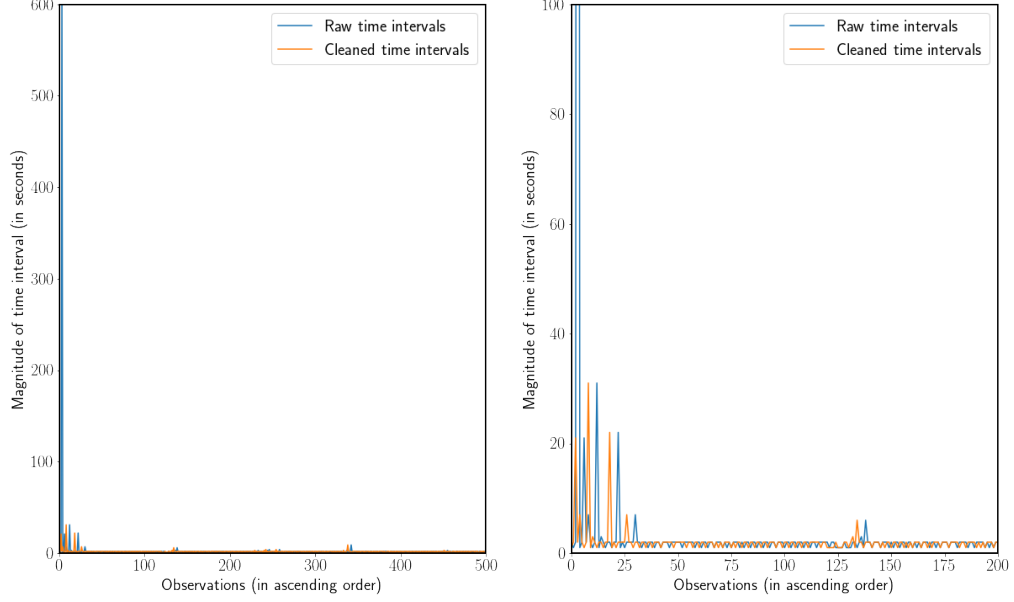


Figure 4.5: Plot of time intervals of raw data and cleaned data from 0-500 (left plot) and from 0-200 (right plot)

As can be seen in figure 4.5, the intervals are now much shorter and more regular. The few remaining outliers and missing observations can be interpolated using the `resample` function with the frequency of 1 second. After doing so the data is ready to apply a Kalman filter to them.

4.2.3 Specification and Parameterization

The Kalman filter uses two central equations that require specification/ parameterization to fit a particular case and context it is used for.² The first equation makes a forward projection about the state \mathbf{s} based on the previous state, an (optional) control input and some error term:³

²The following derivation is based on [54].

³In the literature it is common to denote the state as \mathbf{x} , however, since we are using x for coordinates \mathbf{s} is chosen in this thesis to avoid any confusion.

$$\mathbf{s}_k = A\mathbf{s}_{k-1} + B\mathbf{u}_{k-1} + \mathbf{w}_{k-1}$$

The matrix A relates the state \mathbf{s} at the previous time step ($k - 1$) to the current time step (k). Matrix B relates the optional control input to the state \mathbf{s} . The last part, \mathbf{w} is the process noise, for which the assumption is made that it is Gaussian noise with mean 0 and variance Q :

$$p(w) \sim N(0, Q)$$

The second equation relates the state in time k to the measurement in time k with some measurement error:

$$\mathbf{z}_k = H\mathbf{s}_k + \mathbf{v}_k$$

Therefore the matrix H is called the transition matrix. Much like \mathbf{w} , \mathbf{v} is considered to be Gaussian white noise:

$$p(v) \sim N(0, R)$$

In order to specify the matrices A and B correctly, it is worthwhile to think about the underlying process. The positions x and y in time k depend on the positions of x and y in the previous time $k - 1$ as well as velocities \dot{x} and \dot{y} in time $k - 1$. It is possible to also take into account accelerations \ddot{x} and \ddot{y} . This yields the following equation:

$$\mathbf{s}_k = \begin{bmatrix} x_k \\ y_k \\ \dot{x}_k \\ \dot{y}_k \end{bmatrix} = \begin{bmatrix} x_{k-1} + \dot{x}_{k-1}\Delta t + \frac{1}{2}\ddot{x}_{k-1}\Delta t^2 \\ y_{k-1} + \dot{y}_{k-1}\Delta t + \frac{1}{2}\ddot{y}_{k-1}\Delta t^2 \\ \dot{x}_{k-1} + \ddot{x}_{k-1}\Delta t \\ \dot{y}_{k-1} + \ddot{y}_{k-1}\Delta t \end{bmatrix}$$

This can be rewritten in matrix multiplication form as follows:

$$\mathbf{s}_k = \begin{bmatrix} x_k \\ y_k \\ \dot{x}_k \\ \dot{y}_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \dot{x}_{k-1} \\ \dot{y}_{k-1} \end{bmatrix} + \begin{bmatrix} \frac{1}{2}(\Delta t)^2 & 0 \\ 0 & \frac{1}{2}(\Delta t)^2 \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \begin{bmatrix} \ddot{x}_{k-1} \\ \ddot{y}_{k-1} \end{bmatrix}$$

In our model we choose the constant velocity assumption. That is, the velocity is constant between time steps (Δt) and therefore the accelerations \ddot{x} and \ddot{y} are both 0. While this is a somewhat simplifying assumption it appears to be justifiable in this situation as our time steps are rather small (1 second). Moreover, variations in speed (i.e. deceleration and acceleration) are taken into account in the process noise covariance matrix Q . Therefore, we do not need to specify the control matrix B as with our assumptions it is only zeros. For the transition matrix we obtain:

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

In order for the Kalman filter to work properly, we also need to parameterized R and Q .⁴

One of the parameters that we need is the matrix of measurement noises R .

$$R = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix}$$

This matrix contains the measurement noises for x and y and their covariance. It is assumed that the covariance is zero so we only need to obtain the variance of x and y . For the measurements from the Impinj xarray we do not know the mean and variance. However, the manufacturer states that in 66% of cases the real measurement will be within 1 meter and in 85% of cases the real measurement will be within 1.5 meters [55]. Assuming

⁴While it is possible to optimize the transition or covarinace matrix, instead of providing them, the underlying algorithms for this are computationally expensive and not suitable in the context of near real-time applications. See for example <https://pykalman.github.io/> for more on this.

that the normal distribution holds for the errors, we can fit a normal distribution to the provided values. We do this by adjusting the shape of the distribution (or in other words the parameter σ^2 , since μ is assumed to be 0), so that the area under there curve is equal to 0.66 (with z values of -0.5 and 0.5). We can do the same for the statement that 85% of observations are within 1.5 meters. Doing so yields a standard deviation of around 0.52 meters. We further assume that measurement errors for x and measurements errors for y are independent (i.e. $\sigma_{xy} = 0$).

To determine the matrix H we need to know what we are measuring. In our application we measure only the position in x and y direction, but not the velocity in either direction. Although velocities could be inferred from obtained coordinates, they are not measured and therefore the matrix H looks as follows:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

This essentially states that we obtain measurements of x and y but not of \dot{x} and neither of \dot{y} .

The process noise Q is often derived based on kinematic equations. A widely used model is called random acceleration [56]:

$$Q = \begin{bmatrix} \frac{\Delta t^4}{4} & 0 & \frac{\Delta t^3}{2} & 0 \\ 0 & \frac{\Delta t^4}{4} & 0 & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & 0 & \Delta t^2 & 0 \\ 0 & \frac{\Delta t^3}{2} & 0 & \Delta t^2 \end{bmatrix} \sigma_a^2$$

The choice of σ_a (standard deviation of the acceleration) is obviously critical here. For this we rely on prior research about pedestrian movements [57] and according to this choose 0.12 m/s^2 . It is, however, also acknowledge in the literature that Q is difficult to estimate [54].

With all relevant input defined, the Kalman filter is setup in Python and applied to the measurements (Python code can be found in the appendix).

4.2.4 Results

In order to understand the impact of the Kalman filter it is helpful to look at the same example that was discussed earlier (see figure 4.3) and compared it to the filtered values.

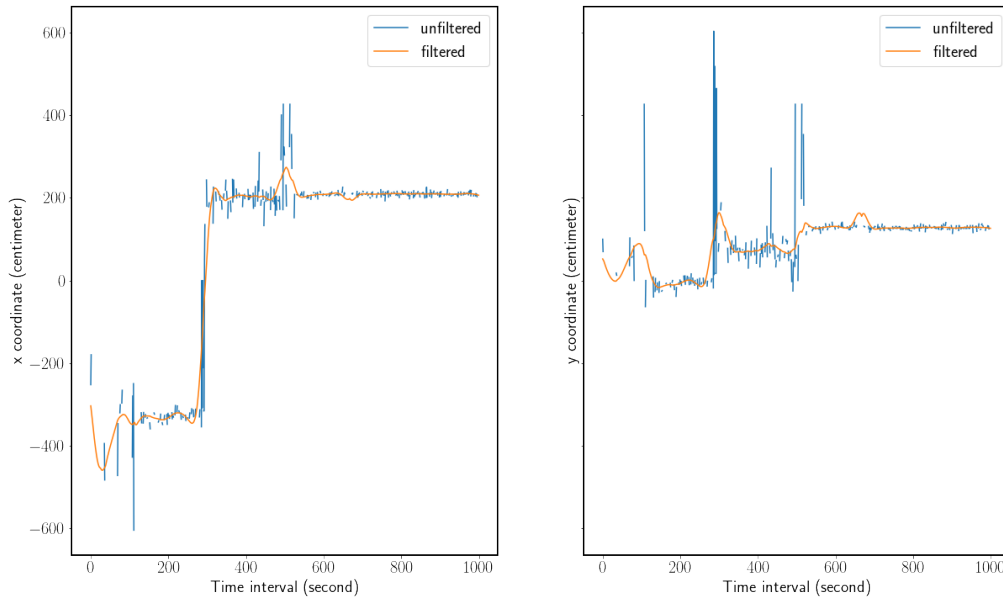


Figure 4.6: Comparison of unfiltered and (Kalman) filtered values for x and y coordinates

Figure 4.6 displays the measured values (blue) versus the filtered values (orange). It is immediately visible, that the large swings are smoothed out (e.g. for x in the interval $[0:200]$, unless there is a sustainable movement in position (e.g. the one that occurs for x in the interval $[200:400]$). This is even more apparent when looking at the y measures as there are a few upward swings, all of which are smoothed out by the Kalman filter.

As stated earlier the scenario used in this example was for a tag which was more or less static in the time interval $[600:1000]$. While figure 4.6 already shows that the filter appears less noisy than the measurements it is worthwhile to examine this in a bit more detail.

Figure 4.7 displays a closeup of the relevant time interval. While the unfiltered values move around quite a bit, the filtered values are almost a straight line (apart from a small variation in the beginning of the interval). This is in line with what we would expect based

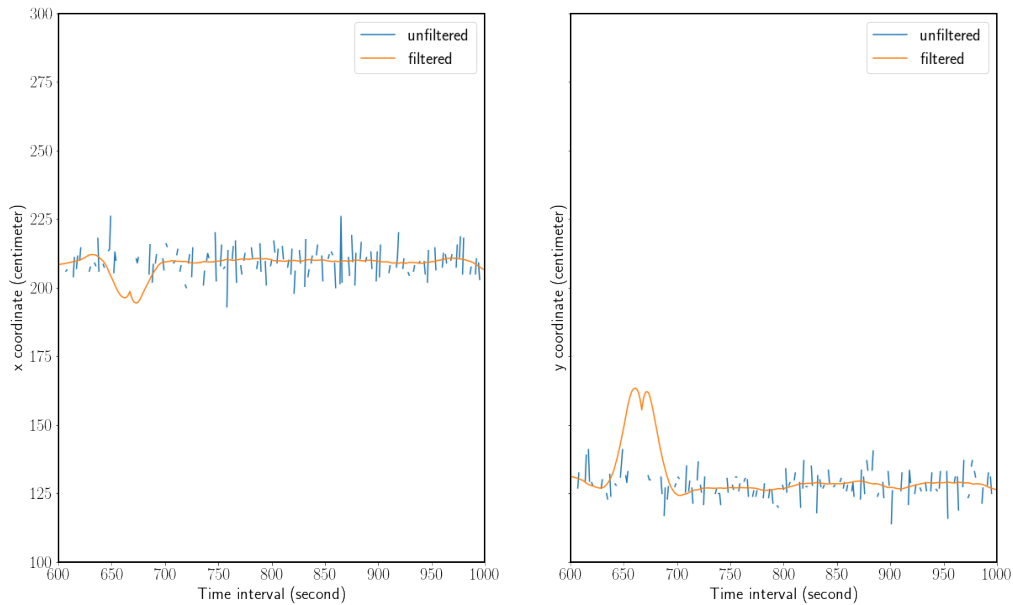


Figure 4.7: Comparison of unfiltered and (Kalman) filtered values for x and y coordinates in the time interval [600:1000]

on the scenario description and thus it appears that the Kalman filter does reasonably well given the initial objective.

4.3 Performance Measures

In the context of real world applications it is often useful to create aggregated views of data to understand, for example, the overall trend and make it accessible for a wider (non-technical) audience. These data can then be used to inform policy decisions or evaluate the effectiveness of marketing measures to name just two examples.

For a tangible scenario, one can think of a trade fair (or any other kind of exhibition) where it is useful for a merchant to understand how visitors behave around its stall. In this context three pieces of information seem of particular interest for practical RFID tracking applications: (i) how many of the visitors spot a particular stall; (ii) how many of the ones who spot the stall move closer to see what it is about; and (iii) how many of the the ones that inform themselves about the stall actually move on to interact with the

merchant. In an ideal setting, KPI are easy to understand and comparable. This is often achieved by using a ratio. We therefore define three key performance indicators (KPI) as follows:

The first KPI relates the views to the opportunities. We call this visibility:

$$\text{Visibility} = \frac{\text{Count of Views}}{\text{Count of Opportunities}}$$

The second KPI relates the visits to the views. We call this engagement rate:

$$\text{Engagement Rate} = \frac{\text{Count of Visits}}{\text{Count of Views}}$$

The third KPI relates the interactions to the visits. We call this interaction rate:

$$\text{Interaction Rate} = \frac{\text{Count of Interactions}}{\text{Count of Visits}}$$

We then need to define zones to determine the count for each variable. In practice this needs to be based on the actual setting and zones may therefore vary (see e.g. [58] for a similar work based on WIFI signals). In our example, we further need to take into account the system design (e.g. possible reading distance of the RFID reader, location accuracy). Therefore we define the zones as circles around each merchant with each zone having a different radius. We need to define at least three zones (one for interaction, one for visit and one for view), with everything else being designated as opportunity. Given that in our example: (i) the reader can read 6m in each direction; (ii) location accuracy is somewhere around 50cm; (iii) we want to simulate multiple merchants; and (iv) the KPI should be sufficiently different for later visualization, we define the zone radii in 100cm steps. That is, the interaction zone has a radius of 100cm, the visit zone has a radius of 200cm and the view zone has a radius of 300cm. These assumptions fit the context of our experiments and also seem realistic. Figure 4.8 shows a schematic overview of these zones.

Further, we establish the following definition:

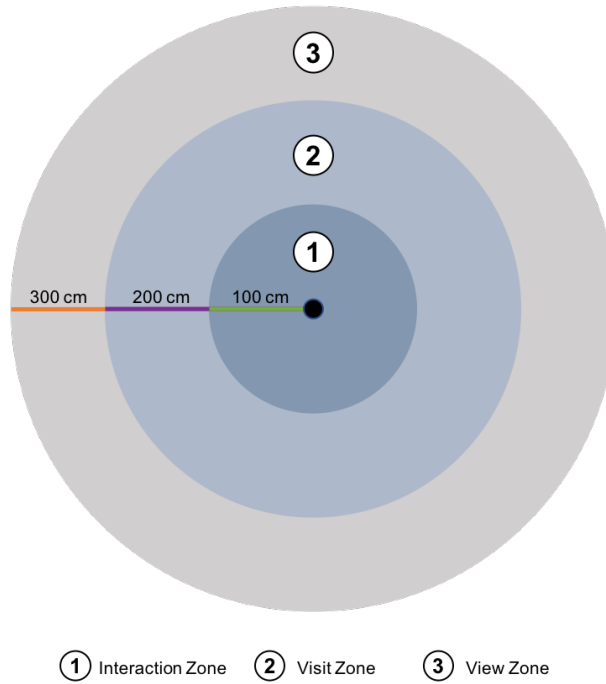


Figure 4.8: Schematic overview of the different zones

$$\text{Interaction} \in \text{Visit} \in \text{View} \in \text{Opportunity}$$

This definition enables easier interpretation (and comparability) in practice as the values are percentages and thus always between 0 and 100. This is because any count in the interaction zone is also a count in the other zones.

Another consideration is comparability. While the above definition enables the reader to quickly understand the number, it does not tell a reader how the number compares. Obviously, one can base the assessment of how good a particular value is on the fact that it is necessarily between 0 and 100. This however is not practical, since we do not know if an interaction rate of 100% is practically achievable. For this we introduce two benchmarks. The first benchmark is an in-group benchmark, comparing the KPI of a particular time period to the minimum and maximum of that KPI during a specified time range. The second benchmark does the same, but instead of using in-group comparison we use out-of-group comparison. Going back to the merchant example of earlier: the first benchmark compares the merchant against itself, while the second benchmark compares

the merchant against all other merchants.⁵

Lastly, we need to define the position of the merchants in our example. In order to have minimal overlap between zones but still enough observation points we chose to position three merchants at the following locations (relative to the origin, i.e. the reader):

- Merchant A (M_A): (150, -250)
- Merchant B (M_B): (250, 250)
- Merchant C (M_C): (0, 300)

If we plot the zones of these merchants against the observations we obtain figure 4.9. We can see from the graph that there is only some overlap in the view zones and that the tag distribution is somewhat different between the zone of the different merchants.

In order to determine whether any tag is within any of the relevant zones, we first calculate the distance between each observation and each merchant. This is done using the Euclidean distance (where p denotes the observation of a tag consisting of x and y coordinates and M denotes the merchant position with x and y coordinates):

$$d(p, M) = \sqrt{(p_x - M_x)^2 + (p_y - M_y)^2}$$

Once this is done a binary variable is created for each merchant's zone. The variables equal 1 if an observation falls within defined zone and 0 otherwise. This ensures that an observation can be simultaneously in multiple zones (as will likely be the case in practice).

4.4 Data Visualization

The KPIs defined in the section above are incorporated into suitable visualization and displayed in the dashboard. There are a range of filters embedded to facilitate the analyt-

⁵Obviously these definitions could be adjusted in practice as needed. For example KPI could be compared across a group of similar merchants and all other.

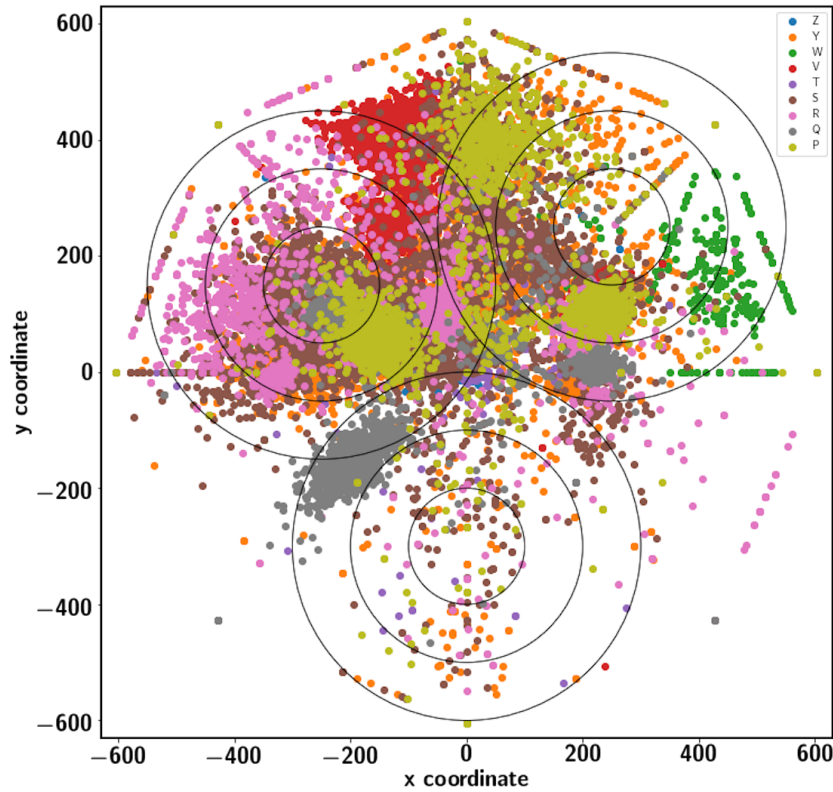


Figure 4.9: Tag distribution within defined zones of interest

ical use of the board. The figures 4.10, 4.11, 4.12 show how the dashboard changes with different filters.

Marketing campaign statistics for merchant A on July 1, 2021 is shown in Figure4.10. It is clear that merchant A has visibility rate at 69%, engagement rate at 45% and interaction rate at 15%. Among the 69% of opportunities that appeared also in the visit zone, less than half of them converts to a visit and less than a fifth of those visitors interacted with merchant representatives. Moreover, one can see that the dials used for benchmarking allows for a quick assessment of the KPI in the broader context. For merchant A, the interaction rate is clearly below its own and peer group averages, while the visibility and engagement rate are both at or slightly above average.

Marketing campaign statistics for merchant B on July 15, 2021 is shown in Figure??. It is clear that merchant B has visibility rate at 45%, engagement rate at 72% and interaction

rate at 7%. Although there is only 45% of opportunities that saw in the visit zone, the conversion rate between people in view zone and in visit zone is as high as nearly three quarters. This heap is quickly dying out and left with just below one in ten visitors who interacted with merchant representatives. Moving right to the dials, the merchant B reaches its own best visibility rate and engagement rate of all time. Its visibility rate is just about the average of the peer group but its engagement rate is only 10% away from the peer group. The interaction rate on the day is very low for both own and peer records.

Marketing campaign statistics for merchant C on June 16, 2021 is shown in Figure???. It is clear that merchant C has visibility rate at 22%, engagement rate at 32% and interaction rate at 44%. Merchant C has least viewers and visitors than other two merchants A and B, which aligns from low density displayed on the scatter plot. However, merchant C has the highest engagement rate when we compare these three snapshots. In other words, merchant C representatives have chance to interact with close to half of people that come by the stand. Looking right at the dials, the merchant C achieves its own best visibility rate and almost best interaction rate of all time. Despite underperforming the peer group in visibility rate and engagement rate, its interaction rate is over the average of the peer group by almost 7%.

4.5 Data Privacy

The RFID-enabled system implemented in the thesis follows strictly the non-invasive and privacy-preserving principle. Firstly, all the hardwares are manufactured by adhering to the privacy requirements. The reader manufacturer Impinj claims "IEEE 802.1x for port-based network access control (MD5, PEAP, MSCHAPv2), encryption of inbound LLRP connections, and SNMP version 3 for remote management for readers and gateways" are in place to prevent outside parties from reading information on RFID tags [52]. The same goes to the Xovis 3D camera, which meets basic privacy requirements and provides customised privacy protection [46]. Secondly, there is no personal information collected in the stage of data capturing and only EPCs of all the tag objects are referenced and used in the stage of data filtering and visual representation.

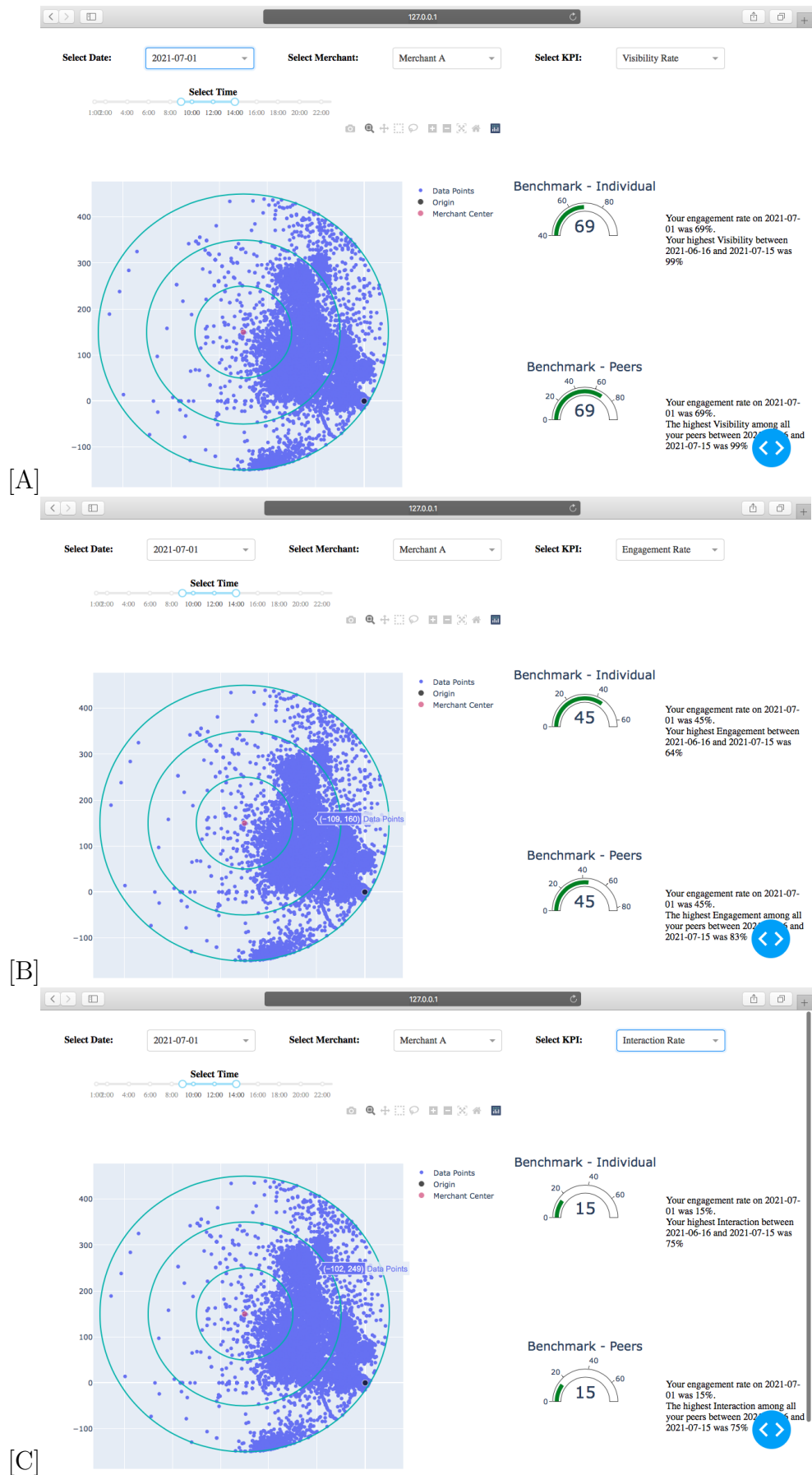


Figure 4.10: Dashboard centered as merchant A on July 1 between 09.00 - 14.00
 (A) Visibility Rate (B) Engagement Rate (C) Interaction Rate

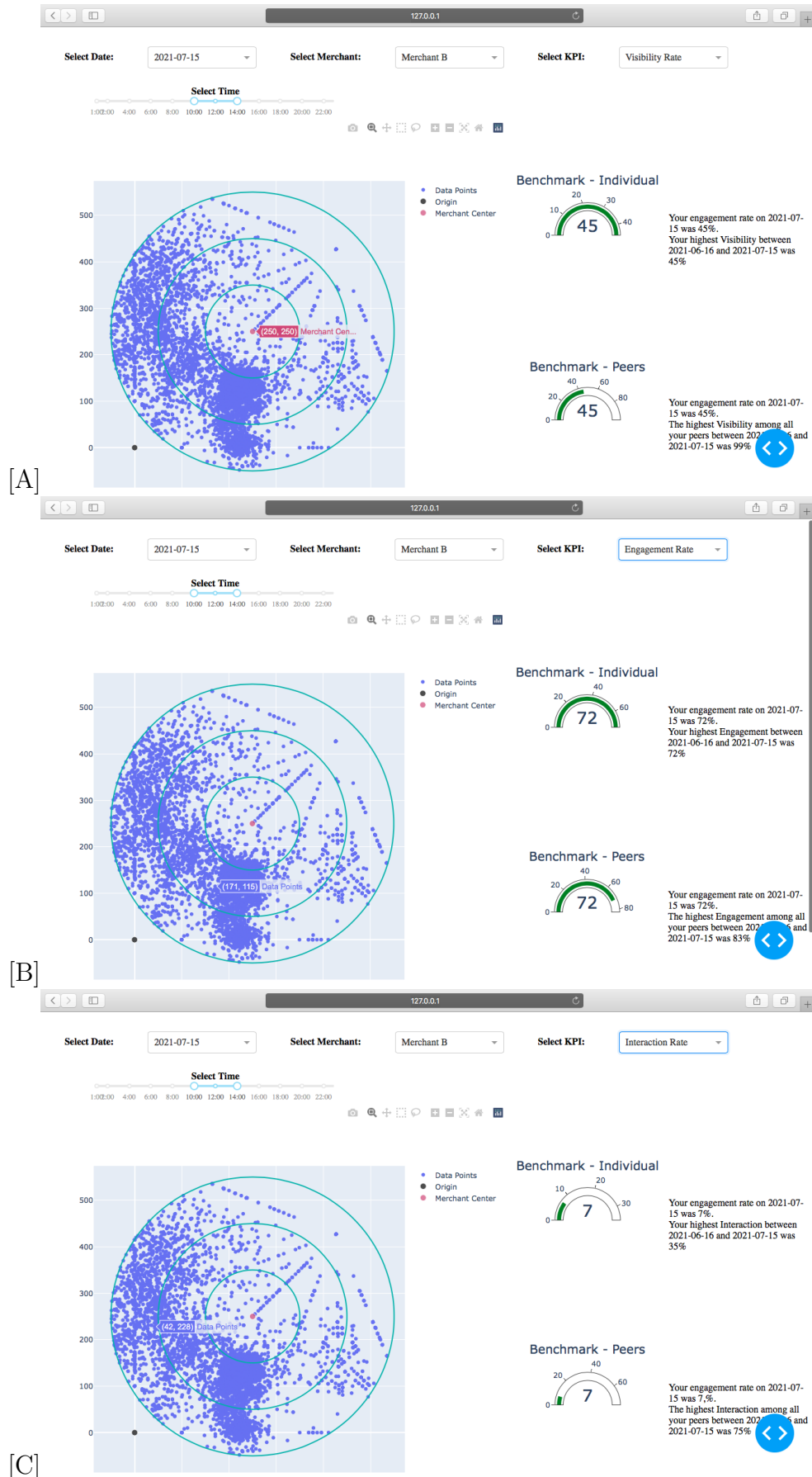


Figure 4.11: Dashboard centered as merchant B on July 15 between 10.00 - 14.00
(A) Visibility Rate (B) Engagement Rate (C) Interaction Rate



Figure 4.12: Dashboard centered as merchant A on June 16 between 13.00 - 16.00
(A) Visibility Rate (B) Engagement Rate (C) Interaction Rate

Chapter 5

Evaluation

This chapter presents the evaluation of the applied Kalman filtering and the crowd management prototype dashboard using UHF passive RFID technology. The focus of the latter lies on both the prototype itself and integration possibility with the existing systems.

5.1 Kalman Filtering

The Kalman filtering used in this thesis has shown to be effective in smoothing out the noisy measurement data from the sensor. The application of such a Kalman filter is also relatively straightforward to implement and due the fact that it only requires the current state as dynamic input while the rest (i.e. matrices A , H , R and Q) are static, it is also reasonably fast. This makes the algorithm a viable candidate for near real-time applications. It has to be noted however, that in the absence of a ground truth it is difficult to evaluate if the localization is really improved or simply smoothed. The presented examples however provide some confidence that the localization is also improved.

5.2 Usability of Prototype

As the web prototype presented in the thesis can be seen as an information-oriented website, the Web Usability Index can be used for evaluating the usability of this web

application [59]. This evaluation will go through the checklist by the Index's five categories because the Web Usability Index consists of a checklist with as many as 150 questions [60].

Starting off from the first category: "Navigation and Orientation: consistency of navigation, color of links, etc". The author believes the web prototype has consistent navigation. It is clear that the dashboard has all the filter options located on the top panel of the website. Three filter options are presented as a dropdown menu and one time frame filter choice is done with a timeline plus two adjustable handles.

In terms of "Interaction and Information-Exchange: availability of a homepage, skip functionality for intros, etc", the web prototype does encourage users to engage on the dashboard by providing responsive display depending on the choice of filters a user has made despite the dashboard having only one page.

For the categories "Being up-to-date and Quality: marking of texts with author and date, absence of spelling mistakes, etc" and "Ease of Access and Accessibility: connection between URL and website, availability of high- and low-tech variants of the website, etc", the prototype has been reviewed by the friend of the author. He has confirmed that the website is available on the localhost and free from context errors.

Coming down to the last category on the list "Information- and Text-Design: size of the font, expressiveness of icons, etc", the three visual charts are intentional choices of the author. The zone of interest scatter plot on the left side aims to present the overview of crowd during the selected time range. The two dial plots are meant to provide prompt comparisons across merchants and/or over time. The text next to the dial is an aid for the users and additionally provides a written explanation and the exact analytical numbers.

5.3 Application Integration

The current dashboard web application is written in python under dash framework. As the framework is built on top of Plotly.js, React, and Flask [48], it is possible to embed the dash app in other web platforms [61]. This integration can be done by embedding dash

apps into the existing cloud counter v1.0 React app via hooks. The multidirectional shared state between React app and Dash app can be realised by importing the components and using it inside JSX.

Chapter 6

Final Considerations

This thesis has processed timestamped location measurements that are produced under the passive RFID indoor tracking system. The scenario-simulated data is then refined using the Kalman filter. These filtered data and a set of KPI for indoor crowd management are combined to build a visual dashboard to display people flow statistics. Overall, an efficient and anonymous RFID-enabled web hosted dashboard is developed for applications of live marketing.

The thesis has two key objectives: one is about data engineering. That is, the author should derive value from data that is collected under the system described in the section 3.1. The second objective is to make a prototype with meaningful visualizations based on information from the first objective.

Both objectives have been fulfilled and documented in the paper. The industrial collaboration with Livealytics and the continuous feedback from my supervisor are the major factors for this completion.

The author had to make the choice of Dash as main framework at the mid of development phase due to complexity of react javascript and limited project duration. Despite the time spent on switching the framework, the beneficial effect of adapting Dash has been shown: firstly, the code for data processing can be leveraged in the implementation as Dash is also based on Python. Secondly, Dash has a strong focus on data analytics, which aligns with the core of this thesis. Thirdly, the initial work spent on react javascript is not wasted

because Dash inherits partially from React. Lastly, Dash combines both front-end and back-end, which makes it ideal for flexible and agile prototype development.

6.1 Future Work

From an academic viewpoint, further studies on the benefit of a RFID-camera hybrid system should be conducted. This study can be seen as early evidence that such hybrid systems have the potential to become robust system to indoor people tracking and analytics. Future effort could also tangibly explain RFID's role in improving accuracy of video-based camera monitoring systems in cases of occlusions as well as camera's role in visually avoiding perspective distortions of RFID reader processing. Researchers could greatly benefit from such studies and spend further time on developing people-oriented indoor tracking applications.

With regards to these tracking applications, the web-based Dash prototype could be deployed on the AWS and be fully integrated with InfluxDB so that the dashboard could run KPI calculation on near real-time data. Lastly, more detailed research into using the Kalman filter in the context of the given system appears promising. For example, further work could establish a ground truth for location data (in experimental settings) and then compare the Kalman filter against that. This would also enable a refinement of the parameters R and Q and thus enhance the value of Kalman filtering in the context of indoor tracking.

Bibliography

- [1] S. Holcer, J. Torres-Sospedra, M. Gould, and I. Remolar, “Privacy in indoor positioning systems: A systematic review”, in *2020 International Conference on Localization and GNSS (ICL-GNSS)*, Tampere, Finland: IEEE, Jun. 2020, pp. 1–6, ISBN: 978-1-72816-455-7. DOI: 10.1109/ICL-GNSS49876.2020.9115496. [Online]. Available: <https://ieeexplore.ieee.org/document/9115496/> (visited on 08/05/2021).
- [2] Livealytics. (2021). “Livealytics – analytics and IoT services for optimizing customer engagement.”, [Online]. Available: <http://livealytics.com/>.
- [3] S. Ahuja and P. Potti, “An introduction to RFID technology”, *Communications and Network*, vol. 02, no. 3, pp. 183–186, 2010. DOI: 10.4236/cn.2010.23026. [Online]. Available: <http://www.scirp.org/journal/doi.aspx?DOI=10.4236/cn.2010.23026> (visited on 07/08/2021).
- [4] BAG. (2021). “How does the SwissCovid app work”, Coronavirus: SwissCovid app and contact tracing, [Online]. Available: <https://www.bag.admin.ch/bag/en/home/krankheiten/ausbrueche-epidemien-pandemien/aktuelle-ausbrueche-epidemien/novel-cov/swisscovid-app-und-contact-tracing.html#-690395909>.
- [5] A. Sagahyroon, M. Eqbal, and F. Khamisi, “Drawing on the benefits of RFID and bluetooth technologies”, in *2010 IEEE Asia Pacific Conference on Circuits and Systems*, Kuala Lumpur, Malaysia: IEEE, Dec. 2010, pp. 180–183, ISBN: 978-1-4244-7454-7. DOI: 10.1109/APCCAS.2010.5774850. [Online]. Available: <http://ieeexplore.ieee.org/document/5774850/> (visited on 08/24/2021).
- [6] C. Matte and M. Cunche, “Spread of MAC address randomization studied using locally administered MAC addresses use historic”, p. 11, 2018.
- [7] R. Richardson. (2018). “Benefits of RFID Technology in Museums”, Benefits of RFID Technology in Museums, [Online]. Available: <https://viscolour.com/using-rfid-technology-in-museums/>.
- [8] Mandigo. (2021). “mandigo:RFID-Systemintegration”, Live-Besucherstrom-Tracking für Veranstaltungen und Messen, [Online]. Available: <https://www.mandigo.de/>.

- [9] Axper. (2021). “Customer behaviour analysis”, People counting & tracking solutions, [Online]. Available: https://axper.com/wp-content/themes/axper_responsive/pdf/CUSTOMER_BEHAVIOR_ANALYSIS.pdf (visited on 03/25/2021).
- [10] Ipsos. (2021). “Footfall counters & analysis”, Ipsos Retail Performance, [Online]. Available: <https://www.ipsos-retailperformance.com/en/services/people-counters/footfall-counters-and-analysis/>.
- [11] Eurotech. (2021). “How Automatic Passenger Counters (APC) can improve Public Transport”, [Online]. Available: <https://www.eurotech.com/en/page/automatic-passenger-counting-system>.
- [12] V-Count. (2021). “V-Count: solution”, People counting, [Online]. Available: <https://v-count.com/>.
- [13] Technis. (2021). “Counting: Manage your visitor’s flow”, Technis Flow: Counting and Air solutions, [Online]. Available: https://technis.com/wp-content/cache/wp-rocket/mytechnis.com/index-https.html_gzip.
- [14] Localise. (2021). “AI-Powered Advanced People Counter Solution”, An advanced people counting solution based on highly-accurate 3D sensors, [Online]. Available: <https://localise.com/solutions/localise-traffic/>.
- [15] A. Bekkali, H. Sanson, and M. Matsumoto, “RFID indoor positioning based on probabilistic RFID map and kalman filtering”, in *Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2007)*, White Plains, NY, USA: IEEE, Oct. 2007, pp. 21–21, ISBN: 978-0-7695-2889-2. DOI: 10.1109/WIMOB.2007.4390815. [Online]. Available: <http://ieeexplore.ieee.org/document/4390815/> (visited on 08/05/2021).
- [16] S. Ting, S. Kwok, A. H. Tsang, and G. T. Ho, “The study on using passive RFID tags for indoor positioning”, *International Journal of Engineering Business Management*, vol. 3, p. 8, Jan. 1, 2011, ISSN: 1847-9790, 1847-9790. DOI: 10.5772/45678. [Online]. Available: <http://journals.sagepub.com/doi/10.5772/45678> (visited on 09/25/2021).
- [17] A. Fu and G. Retscher, “Active rfid trilateration and location fingerprinting based on rssi for pedestrian navigation”, *Journal of Navigation*, vol. 62, pp. 323–340, 2009, ISSN: 2. DOI: 10.1017/S0373463308005195. [Online]. Available: <https://doi.org/10.1017/S0373463308005195> (visited on 2009).
- [18] C. Ruz, C. Pieringer, B. Peralta, I. Lillo, P. Espinace, R. Gonzalez, B. Wendt, D. Mery, and A. Soto, “Visual recognition to access and analyze people density and flow patterns in indoor environments”, in *2015 IEEE Winter Conference on Applications of Computer Vision*, Waikoloa, HI, USA: IEEE, Jan. 2015, pp. 1–8, ISBN: 978-1-4799-6683-7. DOI: 10.1109/WACV.2015.8. [Online]. Available: <http://ieeexplore.ieee.org/document/7045862/> (visited on 08/26/2021).

- [19] B. Zoghi, R. Singhal, R. Fink, and Y.-K. Jung, “RFID solutions: Parking optimization and customer satisfaction”, presented at the Proceedings of The 2006 IJME - INTERTECH Conference (Session ENT-103-007), Texas A&M University, 2006. [Online]. Available: https://www.ijme.us/cd%5C_06/PDF/ENT%5C%20103-007.pdf (visited on 07/08/2021).
- [20] G. R. Foxall, “Invitation to consumer behavior analysis”, *Journal of Organizational Behavior Management*, vol. 30, no. 2, pp. 92–109, May 28, 2010, ISSN: 0160-8061, 1540-8604. DOI: 10.1080/01608061003756307. [Online]. Available: <http://www.tandfonline.com/doi/abs/10.1080/01608061003756307> (visited on 07/08/2021).
- [21] S. S. Saab and Z. S. Nakad, “A standalone RFID indoor positioning system using passive tags”, *IEEE Transactions on Industrial Electronics*, vol. 58, no. 5, pp. 1961–1970, May 2011, ISSN: 0278-0046, 1557-9948. DOI: 10.1109/TIE.2010.2055774. [Online]. Available: <https://ieeexplore.ieee.org/document/5504205/> (visited on 07/08/2021).
- [22] J. Arellano. (Jul. 9, 2019). “Bluetooth vs. Wi-Fi for IoT: Which is Better?”, Bluetooth vs. Wi-Fi for IoT: Which is Better?, [Online]. Available: <https://www.verypossible.com/insights/bluetooth-vs.-wi-fi-for-iot-which-is-better>.
- [23] Assetinfinity. (Feb. 2021). “RFID vs BLE: How Are They Different in Terms of Asset Tracking?”, [Online]. Available: <https://www.assetinfinity.com/blog/rfid-vs-ble-how-are-they-different-in-terms-of-asset-tracking>.
- [24] Filmora. (Jun. 25, 2021). “360 Degree Cameras Vs 3D Cameras: What are the Differences”, 360 Degree Cameras Vs 3D Cameras: What are the Differences, [Online]. Available: <https://filmora.wondershare.com/virtual-reality/360-degree-cameras-vs-3d-cameras.html>.
- [25] Omron. (Aug. 2021). “Proximity Sensors”, Proximity Sensors, [Online]. Available: <https://www.ia.omron.com/support/guide/41/introduction.html>.
- [26] F. Manzoor, Y. Huang, and K. Menzel, “Passive RFID-based indoor positioning system, an algorithmic approach”, in *2010 IEEE International Conference on RFID-Technology and Applications*, Guangzhou, China: IEEE, Jun. 2010, pp. 112–117, ISBN: 978-1-4244-6697-9 978-1-4244-6698-6. DOI: 10.1109/RFID-TA.2010.5529871. [Online]. Available: <http://ieeexplore.ieee.org/document/5529871/> (visited on 07/08/2021).
- [27] W. Huang, C. Ding, S. Wang, X. Jing, Y. Luo, Y. Zhang, and Z. Zhang, “An efficient visualization method of RFID indoor positioning data”, in *The 2014 2nd International Conference on Systems and Informatics (ICSAI 2014)*, Shanghai, China: IEEE, Nov. 2014, pp. 497–504. DOI: 10.1109/ICSAI.2014.7009338. [Online]. Available: <http://ieeexplore.ieee.org/document/7009338/> (visited on 08/24/2021).

- [28] J. Na, “The blind interactive guide system using RFID-based indoor positioning system”, in *Computers Helping People with Special Needs*, K. Miesenberger, J. Klaus, W. L. Zagler, and A. I. Karshmer, Eds., vol. 4061, Series Title: Lecture Notes in Computer Science, Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 1298–1305, ISBN: 978-3-540-36020-9 978-3-540-36021-6. DOI: 10.1007/11788713_187. [Online]. Available: http://link.springer.com/10.1007/11788713_187 (visited on 09/20/2021).
- [29] S. C. Spinella, A. Iera, and A. Molinaro, “On potentials and limitations of a hybrid WLAN-RFID indoor positioning technique”, *International Journal of Navigation and Observation*, vol. 2010, pp. 1–11, Jun. 7, 2010, ISSN: 1687-5990, 1687-6008. DOI: 10.1155/2010/397467. [Online]. Available: <https://www.hindawi.com/journals/ijno/2010/397467/> (visited on 09/25/2021).
- [30] S. Holm, “Hybrid ultrasound-RFID indoor positioning: Combining the best of both worlds”, in *2009 IEEE International Conference on RFID*, Orlando, FL: IEEE, Apr. 2009, pp. 155–162, ISBN: 978-1-4244-3337-7. DOI: 10.1109/RFID.2009.4911169. [Online]. Available: <http://ieeexplore.ieee.org/document/4911169/> (visited on 09/25/2021).
- [31] C. Wang and L. Cheng, “RFID & vision based indoor positioning and identification system”, in *2011 IEEE 3rd International Conference on Communication Software and Networks*, Xi’an, China: IEEE, May 2011, pp. 506–510. DOI: 10.1109/ICCSN.2011.6014945. [Online]. Available: <http://ieeexplore.ieee.org/document/6014945/> (visited on 09/25/2021).
- [32] A. Savochkin, Y. Mickhayluck, V. Iskiv, A. Schekaturin, A. Lukyanchikov, D. Savochkin, and E. Levin, “Passive RFID system for 2d indoor positioning”, in *2014 20th International Conference on Microwaves, Radar and Wireless Communications (MIKON)*, Gdansk, Poland: IEEE, Jun. 2014, pp. 1–3, ISBN: 978-83-931525-2-0 978-617-607-553-0 978-617-607-555-4. DOI: 10.1109/MIKON.2014.6899990. [Online]. Available: <http://ieeexplore.ieee.org/document/6899990/> (visited on 07/08/2021).
- [33] J. Zhou and J. Shi, “RFID localization algorithms and applications—a review”, *Journal of Intelligent Manufacturing*, vol. 20, no. 6, pp. 695–707, Dec. 2009, ISSN: 0956-5515, 1572-8145. DOI: 10.1007/s10845-008-0158-5. [Online]. Available: <http://link.springer.com/10.1007/s10845-008-0158-5> (visited on 09/25/2021).
- [34] C. Li, L. Mo, and D. Zhang, “Review on UHF RFID localization methods”, *IEEE Journal of Radio Frequency Identification*, vol. 3, no. 4, pp. 205–215, Dec. 2019, ISSN: 2469-7281, 2469-729X. DOI: 10.1109/JRFID.2019.2924346. [Online]. Available: <https://ieeexplore.ieee.org/document/8744656/> (visited on 09/25/2021).
- [35] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, “LANDMARC: Indoor location sensing using active RFID”, *Wireless Networks*, vol. 10, pp. 701–710, 2004.

- [36] M. A. Khan and V. K. Antiwal, "Location estimation technique using extended 3-d LANDMARC algorithm for passive RFID tag", in *2009 IEEE International Advance Computing Conference*, Patiala, India: IEEE, Mar. 2009, pp. 249–253, ISBN: 978-1-4244-2927-1. DOI: 10.1109/IADCC.2009.4809016. [Online]. Available: <http://ieeexplore.ieee.org/document/4809016/> (visited on 09/27/2021).
- [37] K. Han and S. H. Cho, "Advanced LANDMARC with adaptive k-nearest algorithm for RFID location system", in *2010 2nd IEEE International Conference on Network Infrastructure and Digital Content*, Beijing, China: IEEE, Sep. 2010, pp. 595–598, ISBN: 978-1-4244-6851-5. DOI: 10.1109/ICNIDC.2010.5657852. [Online]. Available: <http://ieeexplore.ieee.org/document/5657852/> (visited on 09/27/2021).
- [38] A. Bhatia, B. Mehta, and R. Gupta, "Different localization techniques for real time location sensing using passive RFID", Ph.D. dissertation, Indian Institute of Technology, Aug. 12, 2007, p. 32. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.452.5773>.
- [39] M. Bouet and A. L. dos Santos, "RFID tags: Positioning principles and localization techniques", in *2008 1st IFIP Wireless Days*, Dubai, United Arab Emirates: IEEE, Nov. 2008, pp. 1–5, ISBN: 978-1-4244-2828-1. DOI: 10.1109/WD.2008.4812905. [Online]. Available: <http://ieeexplore.ieee.org/document/4812905/> (visited on 09/27/2021).
- [40] T. Sanpechuda and L. Kovavisaruch, "A review of RFID localization: Applications and techniques", in *2008 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, Krabi, Thailand: IEEE, May 2008, pp. 769–772, ISBN: 978-1-4244-2101-5. DOI: 10.1109/ECTICON.2008.4600544. [Online]. Available: <http://ieeexplore.ieee.org/document/4600544/> (visited on 09/27/2021).
- [41] International Conference on Advanced Communication Technology, *ICACT: 2010 The 12th International Conference on Advanced Communication Technology : 7-10 February 2010*. 2010, OCLC: 1035788794, ISBN: 978-1-4244-5428-0 978-89-5519-146-2 978-1-4244-5427-3. [Online]. Available: <https://ieeexplore.ieee.org/servlet/opac?punumber=5433809> (visited on 09/27/2021).
- [42] S. S. Saab, M. Wassim, and S. Saliba, "A conceptualisation study for using rfid as a stand-alone vehicle positioning system", *International Journal of Radio Frequency Identification Technology and Applications*, vol. 2, no. 1-2, pp. 27–45, 2009. (visited on 2009).
- [43] Chung-Hao Huang, Lun-Hui Lee, C. C. Ho, Lang-Long Wu, and Zu-Hao Lai, "Real-time RFID indoor positioning system based on kalman-filter drift removal and heron-bilateration location estimation", *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 3, pp. 728–739, Mar. 2015, ISSN: 0018-9456, 1557-9662. DOI: 10.1109/TIM.2014.2347691. [Online]. Available: <http://ieeexplore.ieee.org/document/6891303/> (visited on 07/08/2021).

- [44] A. M. Costin, J. Teizer, B. Schoner, and T. ThingMagic, “Rfid and bim-enabled worker location tracking to support real-time building protocol control and data visualization”, *Journal of Information Technology in Construction*, p. 23, Jan. 2015, ISSN: 1874-4753. [Online]. Available: <http://www.itcon.org/2015/29>.
- [45] D. Grau, C. Caldas, C. Haas, P. Goodrum, and J. Gong, “Assessing the impact of materials tracking technologies on construction craft productivity”, *Automation in Construction*, vol. 18, pp. 903–911, 2009.
- [46] I. Cepilov, “Cloud counter (c-count) 2.0”, masterthesis, University of Zurich, Aug. 10, 2021, 99 pp. [Online]. Available: <https://files.ifi.uzh.ch/CSG/staff/rodrigues/extern/theses/ma-ile-cepilov.pdf>.
- [47] B. Vogt and J. Teizer, “Sparse laser scanning to improve off-shore pipe-fitting productivity”, presented at the Proceedings of the Construction Research Congress, Bahamas: Freeport, May 6, 2007.
- [48] Plotly. (2021). “Introduction to Dash”, Introduction to Dash, [Online]. Available: <https://dash.plotly.com/introduction>.
- [49] I. Developers. (2021). “Reference Guide”, Reference Guide, [Online]. Available: https://platform.impinj.com/site/docs/itemsense/reference_guide/index.gsp%20readerconfigurationnamedefaultconfiguration-36.
- [50] J. Skinner. (Jul. 8, 2021). “Reader Modes Made Easy”, Reader Modes Made Easy, [Online]. Available: <https://support.impinj.com/hc/en-us/articles/360000046899-Reader-Modes-RF-Modes-Made-Easy>.
- [51] Z. Zhang, Z. Lu, V. Saakian, X. Qin, Q. Chen, and L.-R. Zheng, “Item-level indoor localization with passive UHF RFID based on tag interaction analysis”, *IEEE Transactions on Industrial Electronics*, vol. 61, no. 4, pp. 2122–2135, Apr. 2014, ISSN: 0278-0046, 1557-9948. DOI: 10.1109/TIE.2013.2264785. [Online]. Available: <http://ieeexplore.ieee.org/document/6519282/> (visited on 10/04/2021).
- [52] J. Skinner. (Jan. 21, 2021). “Speedway Reader and Gateway FAQ”, Impinj support, [Online]. Available: <https://support.impinj.com/hc/en-us/articles/360000130180-Speedway-Reader-and-Gateway-FAQ>.
- [53] R. Kalman, “A new approach to linear filtering and prediction problems”, *Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.
- [54] G. Welch and G. Bishop, “An introduction to the kalman filter”, p. 16, 2006.
- [55] Impinj. (2021). “xArray Gateway Always-On, Wide-Area Monitoring, Locationing”, xArray Gateway Always-On, Wide-Area Monitoring, Locationing, [Online]. Available: https://rfid4ustore.com/content/data_sheet/Impinj_xArray_productbrief_110415-2.pdf.

- [56] K. Saho, “Kalman filter for moving object tracking: Performance analysis and filter design”, in *Kalman Filters - Theory for Advanced Applications*, G. L. d. O. Serra, Ed., InTech, Feb. 21, 2018, ISBN: 978-953-51-3827-3 978-953-51-3828-0. DOI: 10.5772/intechopen.71731. [Online]. Available: <http://www.intechopen.com/books/kalman-filters-theory-for-advanced-applications/kalman-filter-for-moving-object-tracking-performance-analysis-and-filter-design> (visited on 10/10/2021).
- [57] K. Teknomo, “Microscopic pedestrian flow characteristics: Development of an image processing data collection and simulation model”, p. 141, 2002.
- [58] L. Pelloni, “Analytics with passive wi-fi signals”, Ph.D. dissertation, University of Zurich, Dec. 6, 2019, 117 pp. [Online]. Available: <https://files.ifi.uzh.ch/CSG/staff/rodrigues/extern/theses/ma-lucaspelloni.pdf>.
- [59] W. S. Harms and J. Strobel, “Usability evaluation von webangeboten mit dem web usability index”, presented at the In Proceedings der 24. DGI Online-Tagung 2002, Frankfurt am Main, Germany: DGI, 2002, pp. 283–292.
- [60] T. Creutzenberg, “Prototyping: An overview of current trends, developments, and research in prototyping”, *Evaluating prototypes for web applications*, pp. 40–48, Feb. 2010, ISSN: 1862-5207. DOI: 10.1.1.227.7242. [Online]. Available: <http://cites.eerx.ist.psu.edu/viewdoc/download?doi=10.1.1.227.7242&rep=rep1&type=pdf#page=47> (visited on 09/25/2021).
- [61] Plotly. (2021). “Embedding Dash Apps in other Web Platform”, [Online]. Available: <https://dash.plotly.com/integrating-dash>.

Glossary

- API** Application programming interfaces simplify software development and innovation by enabling applications to exchange data and functionality easily and securely
- EDA** Exploratory data analysis is a process of performing initial investigations on data so as to discover patterns, to spot anomalies.
- EPC** Electronic product code is a universal identifier that follows the EPCglobal Tag Data Standard and provides a unique identity for every physical object anywhere in the world, for all time.
- GPS** Global positioning system is a satellite-based radionavigation system that provides geolocation and time information to a GPS receiver anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites.
- IPS** Indoor positioning system is a technology that used to locate people of objects for innodr environment
- KPI** Key Performance Indicators are the critical (key) indicators of progress toward an intended result
- RFID** Radio-Frequency IDentification is a system consists of a tiny radio transponder, a radio receiver and transmitter. It uses eletromagnetic fields to automatically identify and track tags attached to objects.
- RSSI** Received signal strength indicator is an estimated measurement of how good a particular radio can hear other connected client's radios.
- UHF** Ultra-High Frequency describes frequency that a passive RFID tag which operates up to 3 GHz and produces 3,000,000,000 electromagnetic waves per second with read range up to 12 meters.
- UID** Unique identifiers define as label that are encoded onto individual tags and carried by radio waves at extremely high rates and at distances well in excess of 10 metres, without line-of-sight contact. One example of UID is the Electronic Product Code (EPC).
- WLAN** Wireless local area network is a network that allows devices to connect and communicate wirelessly.

Appendix A

Installation Guidelines for Dash Dashboard

This guide assumes that the user has the latest python installed under the virtual environment directory below

1. create a virtual environment, call it rfid_dashboard

```
$python3 -m venv rfid_dashboard
```

2. activate the virtual environment

```
$source rfid_dashboard/bin/activate
```

3. install dependencies

```
$Pip3 install dash  
$Pip3 install daytime
```

4. redirect the directory to where the 'dash_dashboard' code stored

```
$cd dash_dashboad
```

5. run dash app

```
$Python dash-app.py
```

6. If the browser does not redirect to <http://127.0.0.1:8050/> automatically, start the web application by opening a web browser and manuell entering <http://127.0.0.1:8050/>

Appendix B

Contributory Facts

B.1 Contents of The Hand-in

- i. Source Codes: Contains all the source codes used for the thesis.
- ii. Thesis: Contains documents of the written report.
 - Zusfsg.txt: Plain text version of the German abstract.
 - Abstract.txt: Plain text version of the English abstract.
 - Thesis.pdf : PDF version of the written report.
 - Sources: Latex source files used to generate the content of the thesis.

B.2 Additional Tables

EPC	Label
E28011700000021501FDEAC1	K
E28011700000021428BFA221	L
E28011700000021428BFA201	N
E28011700000021428BF80F1	O
E28011700000021428BF80E1	P
E28011700000021428BF80D1	Q
E28011700000021428BF80C1	R
E28011700000021428BF80B1	S
E28011700000021428BF80A1	T
E28011700000021428BF8091	U
E28011700000021428BF8081	V
E28011700000021428BF8071	W
E28011700000021428BF8061	Y
E28011700000021428BF8051	Z

Table B.1: Tag EPC-label mapping

B.3 Codes

B.3.1 Data Preprocessing

Type: python jupyter notebook **Input:** full_data.csv **Output:** AllData.csv

```

1 # import libraries
2 import os
3 import pandas as pd
4 import numpy as np
5 import matplotlib.pyplot as plt
6 import matplotlib.dates as mdates
7 import datetime
8 import time # method mktime is used
9
10 # import dataset, using 'parse_dates' to convert the dates from strings to datetime
11 file = "full_data.csv" # contains all the three simulations
12 full_data = pd.read_csv(file,
13                          parse_dates=['time'])
14
15 # understand data
16 full_data.info() # view dataframe info
17 full_data.dtypes # view column data types
18 type(full_data['time'][0])
19
20 full_data.head()
21
22 #####reshaping#####
23 #####add new col "New Time" by adding 2hrs
24 #####add euclidean distance from each coordinates #to the center (0,0): new col "Distance"
25 #####add new col "timestamp" for time calculation
26
27 ## create a new column with right time
28 AdjustH = np.timedelta64(2, 'h')
29 newT=full_data.time + AdjustH
30 full_data['NewTime']= list(newT)
31
32 ## create a new column for distance to the center
33 tag_coordi=full_data.loc[:, 'xCm': 'yCm']
34 # center array
35 p0=np.array((0,0)) # center
36 # tag coordinate array
37 pi=tag_coordi.values
38
39 p0_arr = np.empty((pi.shape))
40 for i in range(len(p0_arr)):
41     p0_arr[i]=p0
42 # tag coordinate array
43 pi=tag_coordi.values
44
45 # calculating Euclidean distance (L2 norm)
46 d=np.sum(np.abs(pi - p0)**2,axis=-1)**(1./2)
47
48 # append to the dataframe
49 full_data['Distance'] = list(d)
50
51 # convert datetime64 to datetime to timestamp
52 time_input = full_data['NewTime'].values
53 tes=[datetime.datetime.utcfromtimestamp((i - np.datetime64('1970-01-01T00:00:00')) /
54      np.timedelta64(1, 's')) for i in time_input]
54 Tes = [time.mktime(i.timetuple()) for i in tes]
55 full_data['timestamp']= Tes

```

```

56
57 # review dataset
58 full_data.head()
59 full_data.shape
60
61 #####cleaning up#####
62 # Delete time col
63 del full_data['time']
64
65
66 # keep only the data related to the deployed tags
67 full_data['epc'].nunique() # number of unique value in a col
68 full_data['epc'].unique()
69
70 # tag dictionary
71 tag_dict = {'K': 'E28011700000021501FDEAC1', 'L': 'E28011700000021428BFA221', 'N': '\n
           E28011700000021428BFA201',
72            'O': 'E28011700000021428BF80F1', 'P': 'E28011700000021428BF80E1', 'Q': '\n
           E28011700000021428BF80D1',
73            'R': 'E28011700000021428BF80C1', 'S': 'E28011700000021428BF80B1', 'T': '\n
           E28011700000021428BF80A1',
74            'U': 'E28011700000021428BF8091', 'V': 'E28011700000021428BF8081', 'W': '\n
           E28011700000021428BF8071',
75            'Y': 'E28011700000021428BF8061', 'Z': 'E28011700000021428BF8051'}
76
77 # filter only used tags
78 filter_list=list(tag_dict.values())
79 active_tags = full_data[full_data['epc'].isin(filter_list)]
80
81 # current data shape
82 active_tags.shape
83
84 active_tags.head()
85
86 #####export#####
87 active_tags.to_csv(r'AllData.csv', index = False, header=True)

```

B.3.2 Data Processing

Type: python jupyter notebook **Input:** AllData.csv

Output: barOverview.png; activity_of_tag.png; ZoneOfInterst.png; DistanceDistribution-PerTag.png

```

1 # import libraries
2 import os
3 import pandas as pd
4 import numpy as np
5 import matplotlib.pyplot as plt
6 import matplotlib.dates as mdates
7 import datetime
8 import time # method mktime is used
9 from matplotlib import rc,rcParams #plot formatting
10 from pylab import *
11
12 # import dataset, using 'parse_dates' to convert the dates from strings to datetime
13 file = "AllData.csv"
14 full_data = pd.read_csv(file)
15
16 # understand data
17 full_data.info() # view dataframe info
18 full_data.dtypes # view column data types
19 full_data.head()
20
21 # EDA
22 full_data.info(memory_usage='deep')

```

```

23 full_data.describe(percentiles=[0.1, 0.25, 0.5, 0.75, 0.9]).T #Statistical \
    description of the train dataset
24 full_data.isna().sum().sum() # no missing data
25
26 full_data_byUID = pd.DataFrame(full_data.groupby(by='epc', as_index=False).count())
27
28 # add labels for tags
29 full_data_byUID["tag_label"]=np.select([full_data_byUID.epc=='\
    E28011700000021501FDEAC1', full_data_byUID.epc=='E28011700000021428BFA221',
30                                     full_data_byUID.epc=='E28011700000021428BFA201', \
    full_data_byUID.epc=='E28011700000021428BF80F1', \
31                                     full_data_byUID.epc=='E28011700000021428BF80E1', \
    full_data_byUID.epc=='E28011700000021428BF80D1', \
32                                     full_data_byUID.epc=='E28011700000021428BF80C1', \
    full_data_byUID.epc=='E28011700000021428BF80B1', \
33                                     full_data_byUID.epc=='E28011700000021428BF80A1', \
    full_data_byUID.epc=='E28011700000021428BF8091', \
34                                     full_data_byUID.epc=='E28011700000021428BF8081', \
    full_data_byUID.epc=='E28011700000021428BF8071', \
35                                     full_data_byUID.epc=='E28011700000021428BF8061', \
    full_data_byUID.epc=='E28011700000021428BF8051'\
    ],
36                                     list(tag_dict.keys()), full_data_byUID.epc)
37 full_data_byUID
38
39 # plot overview of detection frequency
40 import matplotlib.ticker as ticker
41
42 fig, ax = plt.subplots(figsize=(16, 8))
43 ax.bar(x=full_data_byUID.tag_label, height= full_data_byUID.readerName)
44
45 ax.set_xlabel('Tag Labels', fontsize=15)
46 ax.set_ylabel('Frequency', fontsize=15)
47 ax.set_title("Overview of detection frequency", fontsize=18)
48
49 # Tag mapping
50 ## tag dictionary
51 tag_dict = {'K': 'E28011700000021501FDEAC1', 'L': 'E28011700000021428BFA221', 'N': '\
    E28011700000021428BFA201',
52            'O': 'E28011700000021428BF80F1', 'P': 'E28011700000021428BF80E1', 'Q': '\
    E28011700000021428BF80D1',
53            'R': 'E28011700000021428BF80C1', 'S': 'E28011700000021428BF80B1', 'T': '\
    E28011700000021428BF80A1',
54            'U': 'E28011700000021428BF8091', 'V': 'E28011700000021428BF8081', 'W': '\
    E28011700000021428BF8071',
55            'Y': 'E28011700000021428BF8061', 'Z': 'E28011700000021428BF8051'}
56
57 # Filtering
58 ## Date: July 15, 2021, Scenario #1
59 Round one: 10 24 '40 - 10'29 29
60
61 #####
62 #####
63 # Dynamic User input
64 #####
65 start_time = '2021-06-16T14:20:02'
66 end_time = '2021-07-15T14:51:44'
67 labels = ['Z', 'Y', 'W', 'V', 'T', 'S', 'R', 'Q', 'P']
68 num_tags = len(labels)
69
70 # time filtering
71 Time_LL=datetime.datetime.utcfromtimestamp((np.datetime64(start_time) - np.\
    datetime64('1970-01-01T00:00:00')) / np.timedelta64(1, 's'))
72 Time_LL=time.mktime(Time_LL.timetuple())
73
74 Time_UL=datetime.datetime.utcfromtimestamp((np.datetime64(end_time) - np.datetime64(\
    '1970-01-01T00:00:00')) / np.timedelta64(1, 's'))
75 Time_UL=time.mktime(Time_UL.timetuple())
76
77 # tag filtering
78 tags = [tag_dict[i] for i in labels]

```

```

79
80 ## result
81 filtered_data = full_data.loc[(full_data['epc'].isin(tags)) & (full_data['timestamp']
    >=Time_LL) & (full_data['timestamp']<=Time_UL), ['epc', 'xCm', 'yCm', 'Distance',
    'NewTime', 'timestamp']]
82
83 # overview of dataset
84 filtered_data.head()
85
86 # Visualization
87 ## Visual #1: Line Chart – Tag capture over time
88 # looping per tag
89 hist_ls=[]
90 bins_ls=[]
91 newHist_ls=[]
92 datetimeBins_ls=[]
93
94 for i in tags:
95     sgTag = filtered_data.loc[filtered_data['epc']==i]
96     hist, bins = np.histogram(sgTag['timestamp'])
97     newHist = np.append(np.array((0)), hist)
98     datetimeBins = [datetime.datetime.fromtimestamp(i) for i in bins]
99     hist_ls.append(hist)
100    bins_ls.append(bins)
101    newHist_ls.append(newHist)
102    datetimeBins_ls.append(datetimeBins)
103
104 # plot tags
105 fig, axes = plt.subplots(1,1, figsize=(20,8))
106 formatter = mdates.DateFormatter('%Y-%m-%d %H:%M:%S')
107 axes.xaxis.set_major_formatter(formatter)
108
109 for j in range(num_tags):
110     axes.plot(datetimeBins_ls[j], newHist_ls[j], label=labels[j])
111
112 # Set title and labels for axes
113 axes.set_xlabel('Date', fontsize = 15)
114 axes.set_ylabel('number of detections', fontsize = 15)
115 axes.set_title('a timeline showing the number of detections per tag over time',
    fontsize = 18)
116
117
118 # Rotate tick marks on x-axis
119 plt.setp(axes.get_xticklabels(), rotation=45)
120 # add legend
121 plt.legend(loc='upper right')
122
123 ## Visual #2: Scatter Chart – Tag distribution in zones of interests
124 # visualisation prep
125 # dataset per tag
126 sgTag_ls = []
127 for i in tags:
128     sgTag = filtered_data.loc[(filtered_data['epc']==i)]
129     #sgTag_ls.append(sgTag.iloc[0:4,:]) # slice rows
130     sgTag_ls.append(sgTag)
131     print(sgTag.iloc[0:4,:])
132
133
134 fig, axes = plt.subplots(1,1, figsize=(20,8))
135
136 for i in range(num_tags):
137     x=pd.DataFrame(sgTag_ls[i], columns=['xCm'])
138     y=pd.DataFrame(sgTag_ls[i], columns=['yCm'])
139     axes.scatter(x,y, label=labels[i])
140     axes.set_xlim(-630, 630)
141     axes.set_ylim(-630, 630)
142
143 plt.legend(loc='upper right')
144 # add circles
145 dc100=plt.Circle((-250,150),300, fill=False) #merchant A
146 dc200=plt.Circle((-250,150),100, fill=False) # interactive zone of A

```

```

147 dc300=plt.Circle((-250,150),200, fill=False) # visit zone of A
148 #dc200=plt.Circle((0,0),200, fill=False)
149 #dc300=plt.Circle((0,0),300, fill=False)
150 #dc500=plt.Circle((0,0),600, fill=False)
151 axes.set_aspect(1)
152 axes.add_artist(dc100)
153 axes.add_artist(dc200)
154 axes.add_artist(dc300)
155 #axes.add_artist(dc500)
156
157 dc100_b=plt.Circle((250,250),300, fill=False) #merchant B
158 dc200_b=plt.Circle((250,250),100, fill=False) # interactive zone of B
159 dc300_b=plt.Circle((250,250),200, fill=False) # visit zone of B
160 #dc200=plt.Circle((0,0),200, fill=False)
161 #dc300=plt.Circle((0,0),300, fill=False)
162 #dc500=plt.Circle((0,0),600, fill=False)
163 axes.set_aspect(1)
164 axes.add_artist(dc100_b)
165 axes.add_artist(dc200_b)
166 axes.add_artist(dc300_b)
167
168 dc100_c=plt.Circle((0,-300),300, fill=False) #merchant C
169 dc200_c=plt.Circle((0,-300),100, fill=False) # interactive zone of C
170 dc300_c=plt.Circle((0,-300),200, fill=False) # visit zone of C
171 #dc200=plt.Circle((0,0),200, fill=False)
172 #dc300=plt.Circle((0,0),300, fill=False)
173 #dc500=plt.Circle((0,0),600, fill=False)
174 axes.set_aspect(1)
175 axes.add_artist(dc100_c)
176 axes.add_artist(dc200_c)
177 axes.add_artist(dc300_c)
178
179 # Set title and labels for axes
180 axes.set_xlabel('x coordinate', fontsize = 15)
181 axes.set_ylabel('y coordinate', fontsize = 15)
182 axes.set_title('tag distribution within defined zones', fontsize = 18)
183
184 #plt.savefig('ZoneOfInterst_v1.png', dpi=80)
185
186 # Visual #3: Histogram – Distance distribution per tag
187 cols = 3
188 rows = 3
189 tag_distance = filtered_data['Distance'].values
190 bins = np.linspace(tag_distance.min(), tag_distance.max()) # the bin edges
191
192 fig, axs = plt.subplots(cols, rows, figsize=(15, 6), sharex=True, sharey=True)
193 fig.suptitle('Distance distribution per tag', fontsize=18)
194 fig.text(0.5, 0.04, 'Distance', ha='center', fontsize=15)
195 fig.text(0.04, 0.5, 'frequency', va='center', rotation='vertical', fontsize=15)
196
197 k = 0
198 for i in range(rows):
199     for j in range(cols):
200
201         Dis_S=sgTag_Is[k]['Distance'].values
202
203         # plot histogram
204         counts0, bins0 = np.histogram(Dis_S)
205         axs[i,j].hist(bins0[:-1], bins0, weights=counts0)
206         #axs[i,j].set_xlabel('Distance', fontsize=10.5)
207         #axs[i,j].set_ylabel('frequency', fontsize=10.5)
208         axs[i,j].title.set_text('Tag %s Distance Distribution'%(labels[k]))
209         k += 1
210
211
212 plt.show()

```

B.3.3 Kalman Filtering

Type: python jupyter notebook **Input:** AllData.csv

Output: Time_Int.png; 'Kalman_Res_Zoom.png'; 'Kalman_Res_Full.png'

```

1  # importing all the necessary libraires
2  import numpy as np
3  import pandas as pd
4  from numpy.linalg import inv
5  from datetime import datetime
6  from pytz import timezone
7  import matplotlib.pyplot as plt
8  import seaborn as sns
9
10 # Kalman Library
11 from pykalman import KalmanFilter
12 from pykalman import UnscentedKalmanFilter
13
14 import warnings
15 warnings.filterwarnings("ignore")
16
17 # Reading the dataset and dropping not required columns
18 data = pd.read_csv("AllData.csv")
19 data = data.drop(columns = ['filialeId', 'readerName'])
20
21 def to_time(time): #Function to get time from "NewTime"
22     if isinstance(time, str):
23         return time.split()[1].split('.')[0]
24 def to_date(time): #Function to get date from "NewTime"
25     if isinstance(time, str):
26         return time.split()[0]
27
28 data['Time'] = data['NewTime'].apply(to_time)
29 data['Date'] = data['NewTime'].apply(to_date)
30
31 ###Sorting the data based on the days.
32 # Sorting per dates
33 day1 = data[data['Date'] == "2021-06-16"]
34 day2 = data[data['Date'] == "2021-07-01"]
35 day3 = data[data['Date'] == "2021-07-15"]
36
37 Tes = data.loc[(data['epc']=='E28011700000021428BF80C1') & (data['Date']=='2021-07-15')]
38
39 #Sort the data based on the timestamp
40 Tes = Tes.sort_values(by=['timestamp'])
41
42 #Drop column that are not needed
43 Tes = Tes.drop(columns=['NewTime', 'Time', 'Date'])
44
45 #Drop all duplicates
46 Tes = Tes.drop_duplicates()
47
48 #Supress scientific notation
49 pd.set_option('display.float_format', lambda x: '%.0f' % x)
50
51 #There are sometimes multiple measurements at the same time, to use the Kalman
52   filter average them
53 Tes_adjusted = Tes.groupby(['timestamp', 'epc'])['xCm', 'yCm', 'Distance'].mean().reset_index()
54
55 #Now check the time interavalls for the measurement as Kalman needs constant time
56   intervalls
57 raw_time_diff = np.diff(Tes_adjusted.timestamp)
58
59 #The data shows that there is a big jump in the beginning, let's find out where it
60   is
61 max(Tes_diff)
62 res=[idx for idx in range(len(Tes_diff)) if Tes_diff[idx] == max(Tes_diff)]
63 res

```



```

61
62 #The jump occurs at diff[3], i.e. from 3rd to 4th observation
63 Tes_adjusted = Tes_adjusted.drop([0, 1, 2, 3])
64 Tes_adjusted = Tes_adjusted.reset_index(drop=True)
65
66 #Now the intervalls look much more balanced but there is still a need to interpolate\
    data, let's look at the smallest time intervall
67 #Smallest time intervall
68 min(np.diff(Tes_adjusted.timestamp))
69 #We need to have 1s intervalls
70
71 # Check time intervalls in plots
72 fig, axes = plt.subplots(1,2, figsize=(20,12),sharex=False, sharey=False)
73
74 #Plot x
75 axes[0].plot(raw_time_diff, label='Raw time intervals')
76 axes[0].plot(np.diff(Tes_adjusted.timestamp), label='Cleaned time intervals')
77 axes[0].legend(loc="upper right", fontsize=20)
78
79 axes[0].set_xlabel('Observations (in ascending order)',fontsize=20)
80 axes[0].set_ylabel('Magnitude of time interval (in seconds)',fontsize=20)
81
82 axes[0].xaxis.set_tick_params(labelsize=20)
83 axes[0].yaxis.set_tick_params(labelsize=20)
84
85 #Set axis ranges
86 axes[0].set_xlim([0, 500])
87 axes[0].set_ylim([0, 600])
88
89 #Plot y
90 axes[1].plot(raw_time_diff, label='Raw time intervals')
91 axes[1].plot(np.diff(Tes_adjusted.timestamp), label='Cleaned time intervals')
92 axes[1].legend(loc="upper right", fontsize=20)
93
94 axes[1].set_xlabel('Observations (in ascending order)',fontsize=20)
95 axes[1].set_ylabel('Magnitude of time interval (in seconds)',fontsize=20)
96
97 axes[1].xaxis.set_tick_params(labelsize=20)
98 axes[1].yaxis.set_tick_params(labelsize=20)
99
100 #Set axis ranges
101 axes[1].set_xlim([0, 200])
102 axes[1].set_ylim([0, 100])
103
104 #Export the Plot
105 plt.savefig('Time_Int.png')
106
107 #library
108 import datetime
109 import time # method mktime is used
110
111 t_copy = Tes_adjusted['timestamp'].values
112 #date_col = [time.strftime("%a %d %b %Y %H:%M:%S", time.gmtime(i)) for i in t_copy]
113 date_col = [datetime.datetime.fromtimestamp(i) for i in t_copy]
114 Tes_adjusted['datetime'] = date_col
115 Tes_adjusted.dtypes
116
117 # Now let's resample for consistent time intervals
118 Tes_adjusted
119 Tes_resampled = Tes_adjusted.set_index('datetime')
120 Tes_adjusted.dtypes
121 Tes_resampled = Tes_resampled.resample('1S').asfreq()
122
123 #Test with 1000 observations
124 Kalman_Test = Tes_resampled.iloc[:1000,: ]
125
126 #Create a masked num.py array with the measurements
127 #measurements = np.ma.masked_invalid(Tes_resampled[['xCm', 'yCm']].values)
128 measurements = np.ma.masked_invalid(Kalman_Test[['xCm', 'yCm']].values)
129
130 #Specify the transition matrix (position and velocity)

```

```

131 A = np.array([[1, 0, 1, 0],
132               [0, 1, 0, 1],
133               [0, 0, 1, 0],
134               [0, 0, 0, 1]])
135
136 #Specify which elements are measured by the sensors, i.e. x and y coordinates
137 H = np.array([[1, 0, 0, 0],
138               [0, 1, 0, 0]])
139
140
141 #Specify the measurement noise according to derived sigma (i.e. 0.52m)
142 R = np.array([[2704, 0],
143               [0, 2704]])
144
145 #Take the first measurement as starting point
146 initial_state_mean = np.hstack([measurements[0, :], 2*[0.]])
147
148 #Initiate the initial state covariance
149 P = np.array([[2704, 0, 0, 0],
150               [0, 2704, 0, 0],
151               [0, 0, 1, 0],
152               [0, 0, 0, 1]])
153
154 #Defined the process noise covarinace based on kinematic equaions with acc. var. 144\
155   cm m/s^2
156 Q = np.array([[36, 0, 72, 0],
157               [0, 36, 0, 72],
158               [72, 0, 1, 0],
159               [0, 72, 0, 1]])
160
161 #Setup the Kalman filter
162 kf = KalmanFilter(transition_matrices=A,
163                  observation_matrices=H,
164                  transition_covariance = Q,
165                  observation_covariance=R,
166                  initial_state_mean=initial_state_mean,
167                  initial_state_covariance=P,
168                  em_vars=['transition_covariance'])
169
170 filtered_state_means, filtered_state_covariances=kf.smooth(measurements)
171
172 from matplotlib import rc,rcParams
173 from pylab import *
174
175 # activate latex text rendering
176 rc('text', usetex=True)
177 rc('axes', linewidth=2)
178 rc('font', weight='bold')
179
180
181 # plot beautify
182 fig, axes = plt.subplots(1,2, figsize=(20,12),sharex=True, sharey=True)
183
184
185 #Plot x
186 axes[0].plot(measurements[:,0], label='unfiltered')
187 axes[0].plot(filtered_state_means[:,0], label='filtered')
188 axes[0].legend(loc="upper right", fontsize=20)
189
190 #axes[0].title.set_text(r'\textbf{Tag R x coordinate over second time interval}')
191 axes[0].set_xlabel('Time interval (second)', fontsize=20)
192 axes[0].set_ylabel('x coordinate (centimeter)', fontsize=20)
193
194 axes[0].xaxis.set_tick_params(labelsize=20)
195 axes[0].yaxis.set_tick_params(labelsize=20)
196
197 #Plot y
198 axes[1].plot(measurements[:,1], label='unfiltered')
199 axes[1].plot(filtered_state_means[:,1], label='filtered')
200 axes[1].legend(loc="upper right", fontsize=20)

```

```

201
202 #axes[1].title.set_text(r'\textbf{Tag R y coordinate over second time interval}')
203 axes[1].set_xlabel('Time interval (second)', fontsize=20)
204 axes[1].set_ylabel('y coordinate (centimeter)', fontsize=20)
205
206 axes[1].xaxis.set_tick_params(labelsize=20)
207 axes[1].yaxis.set_tick_params(labelsize=20)
208
209 plt.savefig('Kalman_Res_Full.png')
210
211 #Set axis ranges
212 plt.xlim([600, 1000])
213 plt.ylim([100, 300])
214 plt.savefig('Kalman_Res_Zoom.png')

```

B.3.4 Dash Dashboard

Type: python3 **Input:** AllData.csv **Output:** dash dashboard app

```

1  import dash
2  from dash import html
3  from dash import dcc
4  from dash.dependencies import Input, Output
5  import numpy as np
6  import pandas as pd
7  from numpy.linalg import inv
8  from datetime import datetime
9  from pytz import timezone
10 import plotly.express as px
11 import plotly.graph_objects as go
12
13 # Reading the dataset and dropping not required columns
14 data = pd.read_csv("AllData.csv")
15 data = data.drop(columns = ['filialeId', 'readerName'])
16 def to_time(time): #Function to get time from "NewTime"
17     if isinstance(time, str):
18         return time.split()[1].split('.')[0]
19
20 def to_date(time): #Function to get date from "NewTime"
21     if isinstance(time, str):
22         return time.split()[0]
23
24 def to_hour(time): #Function to get time from "NewTime"
25     if isinstance(time, str):
26         return time.split()[1].split(':')[0]
27
28 def Crtitcal_Zone(Distance_M): #Function to get the zone based on the distance
29     if (Distance_M <= 100):
30         return "Interaction"
31     elif (Distance_M <= 200):
32         return "Visit"
33     elif (Distance_M <= 300):
34         return "View"
35     elif (Distance_M > 300):
36         return "Opportunity"
37
38 def zoneValues(centerX, centerY): # To get zone values for plotly, This returns x0\
, y0, x1 and y1 that required to draw the zones the scatter plot
39     cx = centerX
40     cy = centerY
41     radius = [100, 200, 300]
42     c_1 = []
43     c_2 = []
44     c_3 = []
45     c_4 = []
46     for i in radius:
47         c_1.append(cx - i)

```

```

48         c_2.append(cy - i)
49         c_3.append(cx + i)
50         c_4.append(cy + i)
51     df = pd.DataFrame(data = {
52         "x0": c_1,
53         "y0": c_2,
54         "x1": c_3,
55         "y1": c_4
56     })
57     return df
58
59 def kpi_matrix(j): #This creates the KPI matrix based on the zone passed in as \
series.
60     v = []
61     e = []
62     inter = []
63     zone = []
64     unique_dates = data['Date'].unique()
65     for i in unique_dates:
66         testdf = data[data['Date'] == i]
67         #some hard coded values.
68         I = testdf[testdf[j] == 'Interaction'].count()[0]
69         V = testdf[(testdf[j] == 'Visit') | (testdf[j] == 'Interaction')].count()[0]
70         WW = testdf[(testdf[j] == 'Visit') | (testdf[j] == 'Interaction') | (testdf[j] \
j == 'View')].count()[0]
71         O = testdf[(testdf[j] == 'Visit') | (testdf[j] == 'Interaction') | (testdf[j] \
j == 'View') | (testdf[j] == 'Opportunity')].count()[0]
72         #finding the KPI values
73         Visibility = WW/O
74         Engagement = V/WW
75         Interaction = I/V
76         # The below if statements are required for error management
77         if (np.isnan(Visibility)):
78             v.append(0)
79         else:
80             v.append(Visibility * 100)
81         if (np.isnan(Engagement)):
82             e.append(0)
83         else:
84             e.append(Engagement * 100)
85         if (np.isnan(Interaction)):
86             inter.append(0)
87         else:
88             inter.append(Interaction * 100)
89         zone.append(j)
90         # Creating the dataframe that will be returned later
91     KPI_df = pd.DataFrame(data = {
92         'Visibility': v,
93         'Engagement': e,
94         'Interaction': inter,
95         'Date': unique_dates,
96         'Zone': zone
97     })
98     return KPI_df
99
100
101 #Applying the date and time functions
102 data['Time'] = data['NewTime'].apply(to_time)
103 data['Date'] = data['NewTime'].apply(to_date)
104 data['Hour'] = data['NewTime'].apply(to_hour).astype(int)
105
106 # Finding the distance between the merchants and the coordinates
107 data['Distance_M1'] = np.sqrt(np.power((data['xCm'] - (-250)),2) + np.power((data['\
yCm'] - (150)),2))
108 data['Distance_M2'] = np.sqrt(np.power((data['xCm'] - (250)),2) + np.power((data['\
yCm'] - (250)),2))
109 data['Distance_M3'] = np.sqrt(np.power((data['xCm'] - (0)),2) + np.power((data['yCm'] \
- (-300)),2))
110
111 data['Zone_M1'] = data['Distance_M1'].apply(Crtitcal_Zone)
112 data['Zone_M2'] = data['Distance_M2'].apply(Crtitcal_Zone)

```

```

113 data['Zone_M3'] = data['Distance_M3'].apply(Critical_Zone)
114
115 # Getting all unique dates and sorting them in ascending order
116 date_for_dropdown = data['Date'].unique()
117 date_for_dropdown.sort()
118 minDate = date_for_dropdown.min()
119 maxDate = date_for_dropdown.max()
120
121 # Creating the kpi matrix for each zone.
122 m1_kpi = kpi_matrix('Zone_M1')
123 m2_kpi = kpi_matrix('Zone_M2')
124 m3_kpi = kpi_matrix('Zone_M3')
125 kpi_all = m1_kpi.append(m2_kpi).append(m3_kpi)
126
127 app = dash.Dash(__name__)
128
129 app.layout = html.Div([
130     html.Div(
131         className="row", children=[
132
133             # DATE SELECTION
134             html.Div(
135                 children="Select Date:",
136                 style={"font-weight": "bold", 'margin': '2.0em'},
137             ),
138
139             html.Div(className='six columns', children=[
140                 dcc.Dropdown(
141                     id='dropdown_date',
142                     options= [{ 'label': i, 'value': i } for i in date_for_dropdown ],
143                     value=date_for_dropdown[0],
144                     clearable=False,
145                     searchable=False,
146                 ),
147                 style={"width": "180px", 'margin': '1.5em'})
148
149             # MERCHANT SELECTION
150             , html.Div(
151                 children="Select Merchant:",
152                 style={"font-weight": "bold", 'margin': '2.0em'},
153             )
154
155             , html.Div(className='six columns', children=[
156                 dcc.Dropdown(
157                     id='dropdown_merchant',
158                     options=[
159                         { 'label': 'Merchant A', 'value': 'Zone_M1' },
160                         { 'label': 'Merchant B', 'value': 'Zone_M2' },
161                         { 'label': 'Merchant C', 'value': 'Zone_M3' },
162                     ],
163                     value = 'Zone_M1',
164                     clearable=False,
165                     searchable=False,
166                 ), style={"width": "180px", 'margin': '1.5em'})
167
168             # KPI SELECTION
169             , html.Div(
170                 children="Select KPI:",
171                 style={"font-weight": "bold", 'margin': '2.0em'},
172             )
173
174             , html.Div(className='six columns', children=[
175                 dcc.Dropdown(
176                     id='dropdown_kpi',
177                     options=[
178                         { 'label': 'Visibility Rate', 'value': 'Visibility' },
179                         { 'label': 'Engagement Rate', 'value': 'Engagement' },
180                         { 'label': 'Interaction Rate', 'value': 'Interaction' },
181                     ],
182                 ),
183                 value = 'Visibility',

```

```

184         clearable=False,
185         searchable=False,
186     ),
187     ], style={"width": "180px", 'margin': '1.5em'}),
188 ], style=dict(display='flex')),
189
190 # Time selector
191 html.Div( className="row", children=[
192     html.Div(children="Select Time", style={"font-weight": "bold", '\
marginTop': '5px', "textAlign": 'center'}),
193     html.Div(className='six columns', children=[ dcc.RangeSlider(
194         id='my-range-slider',
195         min=1,
196         max=int(data['Time'].max().split(':')[0]),
197         marks={
198             1: '1:00',
199             2: '2:00',
200             4: '4:00',
201             6: '6:00',
202             8: '8:00',
203             10: '10:00',
204             12: '12:00',
205             14: '14:00',
206             16: '16:00',
207             18: '18:00',
208             20: '20:00',
209             22: '22:00',
210             24: '24:00',
211         },
212         value=[int(data['Time'].min().split(':')[0]), int(data['Time\
': ''].min().split(':')[0]) + 5] ) ]),
213 ], style={"width": "35%", "marginLeft": '60px'}),
214
215 html.Div(
216     className="row", children=[
217         # Scatter plot rendering
218         html.Div(className='six columns', children=[
219             dcc.Graph(id='dd-output-container', style={"width": "100vh", "\
height": "90vh"}),
220         ]),
221
222         # Kpi guage rendering
223         html.Div(
224             className="row", children=[
225                 html.Div( children=[
226                     dcc.Graph(id='dd-output-kpi-container-1', style={"width": "\
: "45vh", "height": "40vh"}),
227                     html.P( id='kpi-1', style={"textAlign": 'left', "\
marginTop": "20vh"}),
228                 ], style=dict(display='flex')),
229
230                 html.Div(className='six columns', children=[
231                     dcc.Graph(id='dd-output-kpi-container-2', style={"width": "\
": "45vh", "height": "40vh"}),
232                     html.P( id='kpi-2', style={"textAlign": 'left', "\
marginTop": "20vh"}),
233                 ], style=dict(display='flex')),
234             ])
235
236     ], style=dict(display='flex') )
237
238
239 ])

```

```

1 # Call back for graph
2 @app.callback(
3     Output('dd-output-container', 'figure'),
4     Input('dropdown_date', 'value'),
5     Input('dropdown_merchant', 'value'),
6     Input('my-range-slider', 'value')
7 )

```

```

8
9 # Upading the scatter plot
10 def update_plot(date_selected, merchant_selected, time_selected):
11     updatedDate = data.copy()
12     updatedDate = updatedDate[updatedDate['Date'] == date_selected]
13     updatedDate = updatedDate.drop(updatedDate[updatedDate[merchant_selected] == '
        Opportunity'].index)
14     updatedDate = updatedDate[(updatedDate['Hour'] >= time_selected[0]) & (
        updatedDate['Hour'] <= time_selected[1])]
15
16     fig = go.Figure()
17     if (merchant_selected == 'Zone_M1'):
18         centerX = -250
19         centerY = 150
20     elif (merchant_selected == 'Zone_M2'):
21         centerX = 250
22         centerY = 250
23     else:
24         centerX = 0
25         centerY = -300
26
27     te = zoneValues(centerX, centerY)
28     for i in te.index:
29         fig.add_shape(type="circle", xref="x", yref="y", x0=te['x0'][i], y0=te['y0'][i],
            x1=te['x1'][i], y1=te['y1'][i], line_color="LightSeaGreen")
30     fig.add_trace(go.Scatter(x=updatedDate['xCm'], y=updatedDate['yCm'], mode='
        markers', name='Data Points'))
31     fig.add_trace(go.Scatter(x=[0], y=[0], mode='markers', name='Origin',
        marker=dict(size=[10], color=['#000000'])))
32     fig.add_trace(go.Scatter(x=[centerX], y=[centerY], mode='markers', name='
        Merchant Center', marker=dict(size=[10], color=[2])))
33     return fig
34
35 # Call back for KPI benchmark 1
36 @app.callback(
37     Output('dd-output-kpi-container-1', 'figure'),
38     Output('kpi-1', 'children'),
39     Input('dropdown_date', 'value'),
40     Input('dropdown_kpi', 'value'),
41     Input('dropdown_merchant', 'value'),
42 )
43
44 # Updating the KPI 1 guage
45 def update_kpi_1(date_selected, kpi_selected, merchant_selected):
46     kpiMatrix = kpi_matrix(merchant_selected)
47     valueAftering1 = int(kpiMatrix[kpiMatrix['Date'] == date_selected][kpi_selected\
        ])
48     fig = go.Figure(go.Indicator(
49         domain={'x': [0, 1], 'y': [0, 1]},
50         value = int(kpiMatrix[kpiMatrix['Date'] == date_selected][kpi_selected]),
51         mode = "gauge+number",
52         title = {'text': "Benchmark - Individual"},
53         gauge = {'axis': {'range': [int(kpiMatrix[kpi_selected].min()), int(
            kpiMatrix[kpi_selected].max()) ]}},
54     )))
55     stringForKPI1 = "Your engagement rate on ", date_selected, " was ",
        valueAftering1, "%." ,html.Br(), "Your highest ", kpi_selected, " between ",
        minDate, " and ", maxDate, " was ", int(kpiMatrix[kpi_selected].max()), "%"
56     return fig, stringForKPI1
57
58 # Call back for KPI benchmark 2
59 @app.callback(
60     Output('dd-output-kpi-container-2', 'figure'),
61     Output('kpi-2', 'children'),
62     Input('dropdown_date', 'value'),
63     Input('dropdown_kpi', 'value'),
64     Input('dropdown_merchant', 'value'),
65 )
66
67 # Updating the KPI 2 guage
68 def update_kpi_2(date_selected, kpi_selected, merchant_selected):

```

```

69     valueAftering2 = int(kpi_all[(kpi_all['Date'] == date_selected) & (kpi_all['Zone'
70     ''] == merchant_selected)][kpi_selected]),
71     print(valueAftering2)
72     fig = go.Figure(go.Indicator(
73         domain = {'x': [0, 1], 'y': [0, 1]},
74         value = int(kpi_all[(kpi_all['Date'] == date_selected) & (kpi_all['Zone'] ==
75         merchant_selected)][kpi_selected]),
76         mode = "gauge+number",
77         title = {'text': "Benchmark — Peers"},
78         gauge = {'axis': {'range': [int(kpi_all[kpi_selected].min()), int(kpi_all[
79         kpi_selected].max())]},
80         stringForKPI2 = "Your engagement rate on " , date_selected , " was " , str(
81         valueAftering2)[1:3] , "%" ,html.Br() , "The highest " ,kpi_selected , " among
82         all your peers between " , minDate , " and " , maxDate , " was " ,int(kpi_all[
83         kpi_selected].max()) , "%"
84     return fig , stringForKPI2
85
86 if __name__ == '__main__':
87     app.run_server(debug=True)

```

B.4 Simulation Description

- i. First round data simulation on June 16th
- ii. Second round data simulation on July 1st
- iii. Third round data simulation on July 15th

Testing 16/06/2021

Track a single tag

Scenario #1 Time: 14:20-14:22

Base case

Start to track the tester when he/ she enters the entrance with the reader. At the same time, the tester uses stopwatch to record his/ her time at the entrance, and at the exit. Compare the data

Extended case

Continue the base case, ask the tester to walk fast/ slow to check if the tracking result changes

Track multiple tags

Scenario #2 Time: 14:23-14:26

Base case

Ask three testers to move around the reader with different radiuses check which tags are recorded. Distances between each pair of testers should be 1.5 meter apart.

Extended case #1

Continue the base case but allow one pair of testers to have less than 1.5 meter distance, check the record to see if both tags are recorded with the same value

Extended case #2

Continue with the base case, however, there is a barrier between Tester C and the reader. Allow the Tester C to move behind the barrier. Then check if these activities are correctly recorded.

Scenario #3 Time: 14:29-14:33

Base case

Let testers carry tags in different places, measure the number of tags that are captured. Setting: Tag P, R are carried with hands; Tag S, Q, M, N are in the testers' pocket; Tag T is in different places, lost and found again. Here, Testers move around random directions

Scenario #4 Time: 14:36

Base case

Ask all the testers to move together from one side of room to another with similar speed

Scenario #5 Time: 14:41

Base case

One person carries tags of the rest. All the testers move around within the scope of camera. The distance between the tag-carrier and the rest is not controlled.

Scenario #6 Time: 14:46 – 14:48

Base case

Split the group of testers into test group and control group. The test group test whether the phone has impact on reading RFID tag. The control group has no phone around the tags.

Setting: Tag P, S, M are carried next to the phone; Tag N, Q are also next to the phone that is constantly flipped; Tag R, Y are in the control group.

Testing 1/07/2021

One tag only

Scenario #1

Base case

Two people walk in parallel, one holds a tag, then he/she passes the tag to the partner on the halfway. Check if the camera captures the tag passing

Two tags only

Scenario #2

Base case

Two people both take one tag each, start from two opposite directions and walk towards and pass each other. No tag swaps.

Extended case #2-0-1

Extend from the base case, but two testers walk with slow speed (~13s per one way)

Extended case #2-0-2

Extend from the base case, but two testers walk with fast speed (<12s per one way)

Extended case #2-1

Extend from the base case, but two testers walk with slow speed (~13s per one way) and swap their tags with each other on the halfway.

Hypothesis: algorithm cannot capture neither the tags nor the swap action

Many tags but only one moving tag

Scenario #3

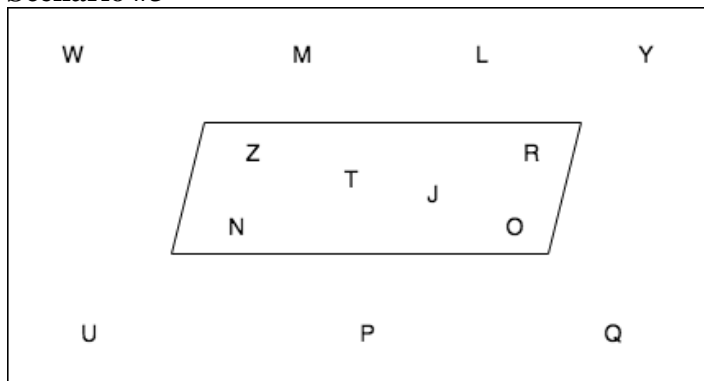


Figure 1: Tags Distribution (self-illustrated)

Base case #3-1

Thirteen tags in total. They are distributed around where the reader/ camera locates (Figure 1). One tester takes a tag S and walk in a linear path. Check if the path of moving tag is correctly captured despite the noise of other tags

Extended case #3-2

Thirteen tags in total. They are distributed around where the reader/ camera locates. One tester takes a tag S and walk randomly including turns. Check if the path of moving tag is correctly captured despite the noise of other tags

Many tags but two moving tags

Scenario #4

Base case

Thirteen tags in total. They are distributed around where the reader/ camera locates (Figure 1). Two tags (N, S) are carries by two testers. Testers then walk pass each other.

Reader performance

Scenario #5

Base case

Thirteen tags in total. They are distributed around where the reader/ camera locates (Figure 1). Keep one type configuration at a time and check the number of tags that are captured.

Different Configuration

Scenario #6

Base case

A test holds a tag and walk across the reader and camera for one minute

Extended case #6-1

Keep base case setting, switch location mode to “Auto Dense Reader”

Extended case #6-2

Keep base case setting, switch location mode to “Auto Static Fast”

Extended case #6-3

Keep base case setting, switch location mode to “Hybrid”

Extended case #6-4

Keep base case setting, switch location mode to “Standard Miller”

Scenario Configuration Table

Scenario No.	Location Mode	Test Time
1	0-max throughput 3 session	11'03'07
2-0-1		11'08'56 - 11'09'59
2-0-2		11'10'00 - 11'11'20
2-1		11'13'00 - 11'14'20
3-1		11'37'00 - 11'37'55
3-2		11'38'37 - 11'39'10
4		11'47'30 - 11'48'45
5		11'54'00 - 11'55'00
2-0-1	Autoset 2 session	11'19'30 - 11'20'57
2-0-2		11'21'30 - 11'22'20
2-1		11'23'00 - 11'24'10
3-1		11'43'10 - 11'44'00
3-2		11'44'20 - 11'45'18
4		11'50'00 - 11'51'18
5		11'52'00 - 11'53'59
6-1	Auto Dense Reader 2 session	11'57'00 - 11'58'20
6-2	Auto Static Fast 2 session	11'59'35 - 12'01'10
6-3	Hybrid 2 session	12'01'15 - 12'02'10
6-4	Standard Miller 2 session	12'02'10 - 12'03'59

Testing 15/07/2021

Note:

the most inner circle - around 2-3m from centre

a few steps further from centre – around 3-6 m from centre

Zones and dwell time, 2 tags only

Scenario #1

Base case

Both Tag Y (E28011700000021428BF80B1) and Tag S (E28011700000021428BF8061) move within the most inner circle, stay for 2 mins, go a few steps further from centre, stay for 1 min, then leave so the reader cannot identify any more.

1-round #1 #2 means the repetition of the same scenario.

Scenario #2

Base case

Tag Y moves in the most inner circle, Tag S moves in the area in the second most inner circle and apart from Tag Y. Both stay for 2 mins. Then Tag S steps in the most inner circle to meet Tag Y. They both stay for 1 min and leave.

Scenario #3

Base case

Both Tag Y and Tag S walk parallelly for two repetitions

Scenario #4

Base case

Tag Y starts from one end, Tag S starts from the other end. Tags Y and S meet and pass each other around centre and finish walking once reach the opposite end. Do that process for two repetitions

1 tag only

Scenario #5

Base case

Tag S walks alone

Many static tags but different configurations

Scenario #6

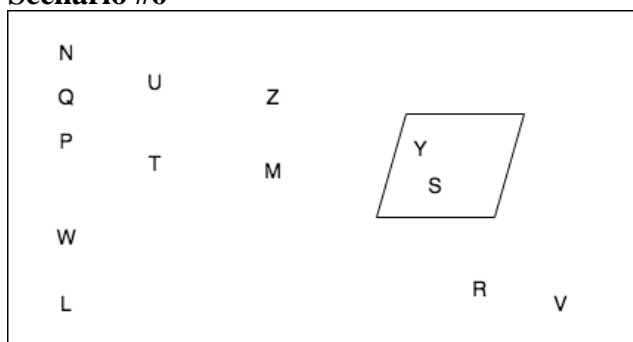


Figure 1: Tags Distribution (self-illustrated)

Thirteen tags in total. They are distributed around where the reader/ camera locates (Figure 1). Keep one type configuration at a time and check the number of tags that are captured.

Extended case #6-1

Keep base case setting, switch location mode to “Auto Static Dense Reader”

Extended case #6-2

Keep base case setting, switch location mode to “Auto Static Fast”

Extended case #6-3

Keep base case setting, switch location mode to “Auto Set Custom”

Extended case #6-4

Keep base case setting, switch location mode to “Auto set dense reader deep scan”

Extended case #6-5

Keep base case setting, switch location mode to “Dense Reader M4”

Extended case #6-6

Keep base case setting, switch location mode to “Dense Reader M8”

Extended case #6-7

Keep base case setting, switch location mode to “Hybrid M2”

Extended case #6-8

Keep base case setting, switch location mode to “Max Miller”

Extended case #6-9

Keep base case setting, switch location mode to “Max Throughput”

Scenario Configuration Table

Scenario No.	Test Time
1-round #1	10'24'40 - 10'29'29
1-round #2	10'31'20 – 10'34'29
2	10'38'33 - 10'41'33
3-round #1	10'51'00 - 10'53'40
3-round #2	10'55'00 - 10'57'15
4-round #1	10'58'30 - 11'00'04
4-round #2	11'02'00 - 11'04'30
5	11'06'30 - 11'08'30
6-1	11'13'50 - 11'15'00
6-2	11'15'05 - 11'16'05
6-3	11'16'15 - 11'17'15
6-4	11'17'30 - 11'18'30
6-5	11'18'40 - 11'19'40
6-6	11'19'50 - 11'20'50
6-7	11'21'00 - 11'22'00
6-8	11'22'15 - 11'23'15
6-9	11'23'25 - 11'24'25