SENSOR EXPLORATION FOR LOCALIZATION IN CLEANROOMS

BY

ISHAN PANKAJ SHAH

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2020

Urbana, Illinois

Adviser:

Professor Klara Nahrstedt

ABSTRACT

Indoor localization is used to locate people or objects within indoor environments such as buildings and rooms. The major applications of indoor localization involve the expansion of location-aware computing such as monitoring and surveillance in scientific labs, accessibility aids, targeted advertisements, inventory tracking etc. This thesis explores localization within the context of the scientific labs called cleanrooms. The philosophy behind indoor localization is first broken down into three parts: presence detection, movement tracking and proximity sensing. The unique challenges for each of these parts facing the cleanroom environments are evaluated and then solved using specific technologies: 1) RFID (Radio Frequency Identification) is used for presence detection to automatically detect the entry / exit of the person into the cleanroom; 2) BLE (Bluetooth Low Energy) is used to continuously track the movement of the person across multiple cleanrooms; 3) Computer vision through a single monocular camera is used to measure the relative proximity of a person to different areas of interest within a room. Our results with RFID provide 82.5% accuracy for presence detection, BLE yields 2.1m accuracy for movement tracking and 62% accuracy for presence detection while computer vision gives an accuracy of 71% for proximity sensing. A comparative study between the three types of sensors highlights the specific benefits and limitations of each technology. It shows that different sensors will have to be combined for full-scale indoor localization which will act as a building block towards the next generation of computing environments.

ACKNOWLEDGEMENTS

I am grateful to my adviser, Professor Klara Nahrstedt, for her vision and guidance. She was always open to new ideas and encouraged me to pursue them. Without her support, resources and valuable insights, this would have never been possible.

I would like to thank my colleagues Anjali Menon and Aabhas Chauhan for their various contributions throughout the project. They were instrumental in successful completion of this work. I am also grateful to Tuo Yu, Patrick Su and Robert Kaufman for their assistance. Building this has been a team effort.

I am thankful to Priyanka Sharma and Vaibhav Vora for their love and affection. Despite being thousands of miles away, they always felt close to me and without them, this journey would have stayed incomplete.

Lastly, I am all because of my parents, Pankaj and Vibha Shah. Their love, sacrifice and motivation has given me the confidence to overcome difficulties and to be brave when lost. They always taught me to accept the present and find happiness and joy in whatever I did. This is dedicated to them.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

Ubiquitous computing envisions the use of smart computing devices to augment physical spaces around us. These devices will be able to perceive its environment and react accordingly through sensors with location and contextual awareness. Location awareness refers to the state where devices can determine their position and help construct the world model around them. The devices can then adapt their behavior based on this contextual information.

Indoor localization is a methodology to locate people or objects within indoor environments such as buildings, rooms etc. As existing satellite positioning systems such as GPS lack precision for such environments, a network of sensor devices is used for this task. This thesis explores such sensor devices for performing localization within cleanrooms.

A cleanroom is a facility utilized as part of special industrial production or scientific research, including the manufacturing of pharmaceutical items, integrated circuits, semiconductors etc. They are designed to maintain extremely low levels of particulates, such as dust, airborne organisms, or vaporized particles. In order to maintain this controlled level of contamination, access to these rooms is restricted to those wearing a cleanroom suit. This suit fully covers the wearer to prevent skin and hair being shed in the cleanroom environment.

There are many useful applications of performing indoor localization in cleanrooms. The data pertaining to the number of people in a cleanroom can be used to evaluate the exposure to the risk levels associated with the type of equipment in there. Tracking the movement of people across the cleanrooms can help in surveillance. The monitoring of which person is working on which equipment can help determine its real-time availability and the usage pattern. These applications fall within the domain of localization and can be modeled into its three subsets - presence detection, movement tracking and proximity sensing.

Presence detection can be defined as detecting the existence of an object in an area. In a cleanroom setting, this can be used for tracking the entry / exit of a person to a cleanroom. The challenges for automatic presence detection involve multiple people passing by at the same time, nondeterministic movements and possible impediments in detection. RFID is a widely used technology used for object detection and inventory tracking [1]. Most of the existing work on

RFID focuses on building an EAS (Electronic Article Surveillance) for detecting objects passing through the exit. However, presence detection in cleanrooms using RFID comes with its set of challenges - vulnerability to human body, reliability on the passive tags to withstand different conditions and false classification of tags (tags that pass near to the exit but not directly through the exit). We propose a specific system for deploying Radio Frequency Identification (RFID) for resolving these problems. A custom algorithm is introduced that tracks RSSI (Received Signal Strength) across time and utilizes peak detection along with predictors for classification. Our solution is able to achieve an accuracy of 82.5% for presence detection.

The essence behind localization is getting the position in real-time of a person or an object as it is moving. This is defined as movement tracking that can help evaluate the coverage area of supervisors moving around the cleanrooms. Accurate indoor positioning is difficult due to physical barriers and obstacles, change in movement speed, large number of people etc. There are many wireless technologies such as Wifi, BLE, Zigbee etc. available for solving this problem, however there are special cases within cleanrooms that have to be taken into account. Cleanrooms have metal equipment that will block or attenuate the signal. There can be interference in the channel due to other electromagnetic waves operating at the same bandwidth. Considering the above issues, BLE is chosen for movement tracking due to its ease of deployment, frequency hopping and compatibility with users' smartphone devices. We explore various solutions to estimate users' position and determine the one that can work in an environment with noisy signals. The average accuracy between the estimated position and the actual position of our implementation with BLE for movement tracking is 2.1m.

The idea behind proximity sensing is measuring the relative proximity of a person to different areas of interest within a room. This can be used to identify the equipment a user is working on. Proximity sensing requires fine-grained localization information. The exact coordinates of a user or an equipment need not be known however its relative proximity needs to be ascertained. This can be particularly difficult if the equipment are located close to each other as is the case within cleanrooms. The traditional sensors generally fail to capture such fine grained details or be too cumbersome to deploy. Hence, we propose a computer vision based approach to solve this problem. The major hurdle in performing proximity sensing with vision

involves translating image to world coordinates which is commonly solved using a stereo-camera. However, most of the cleanrooms already have a monocular camera installed and replacing all of these with stereo cameras will be cost prohibitive with additional deployment challenges. Therefore, a single monocular camera is chosen for this approach with calibration involving intrinsic and extrinsic parameters to compute the homography to go from 2D image to 3D world coordinates. In cleanrooms, as both the person wearing the white suit and the background is white, some of the vision algorithms do not work as they rely on feature descriptors. We have introduced a vision algorithm for proximity sensing that can work taking such characteristics into account yielding an accuracy of 71%.

The comparative study of these three technologies demonstrates that each one can only be used to solve a specific function of localization effectively. It is not possible for a single type of sensor system to perform all of the different functions of localization due to technical limitations and practical constraints. Therefore, full scale indoor localization will require a hybrid sensor fusion approach to leverage the complementarity of several of these technologies.

The major contributions of the thesis can be summarized as follows: (1) analyzing challenges in performing localization in industrial / scientific labs such as cleanrooms; (2) introducing sensor systems that can overcome these challenges - RFID for presence detection, BLE for movement tracking and computer vision for proximity sensing; (3) comparing the advantages, limitations and applicability of these sensor systems.

RFID for presence detection is explained in chapter 2 with the custom classification algorithm described in section 2.4.3 and the experiment along with its results in section 2.5. Chapter 3 enumerates on BLE for movement tracking with challenges in section 3.2, position estimation pipeline in section 3.4 and experimental results in section 3.5.3. The computer vision system for proximity sensing is described in chapter 4 with the sensing algorithm in section 4.4 and the resulting image with evaluation in section 4.5.3. We conclude the thesis in chapter 5 with summary in section 5.1 and discussions and future work in section 5.2.

CHAPTER 2: PRESENCE DETECTION WITH RFID

2.1 Introduction

Radio Frequency Identification (RFID) uses electromagnetic fields to automatically capture digital data encoded in tags. An RFID system consists of three components: reader, antenna and a tag. The tag which consists of an integrated circuit and a radio transponder transmits the digital data when triggered by radio waves emitted by a nearby reader through an antenna. The reader receives the data transmitted by the tag which can be used for identification and tracking. There are two types of RFID tags, active and passive. Active tags are powered by a battery while the passive tags are energized by the radio waves emitted by the reader. While active tags have a higher read range, passive tags are cheaper and smaller. The passive tags however require a stronger power level by the reader to get illuminated and respond back. The signaling between the reader and the tag is done in a lot of frequency bands ranging from low frequency (120 - 150 kHz) to ultra wide band (3.1 - 10 Ghz) [1].

RFID is widely used as an identification technology in supply-chain and inventory management. The low cost of RFID passive tags make them an economically feasible option for tagging objects. Presence detection involves detecting the existence of an object in an area without the need to obtain its exact coordinates. RFID passive tags can be effectively used for presence detection due to its portability and economic feasibility. The objects can be easily tagged and the reader can detect these tags the moment they enter the range even without having a clear line of sight. The coverage area of detection can be contained through the transmission power from the reader to the antenna. This is in contrast better than Near Field Communication (NFC) where the objects need to be manually tapped near the reader. In the case of cleanrooms, it is possible to tag the person with an RFID tag which can be detected by the reader placed near the exit. This is similar to an EAS (Electronic Article Surveillance) that detects objects passing through an exit.

2.2 Challenges

Owing to its nature as a fully automatic data collection technology, RFID suffers from different phenomena surrounding the capturing of RF signals. One such problem is 'false-positive' reads, which is the undesired detection of RFID-equipped objects within the range of the electromagnetic field. This poses a challenge in using RFID based systems for applications such as electronic article surveillance (EAS) as they rely on precise detection. RFID detection is also vulnerable to occlusion of tags by aluminium, human body and liquids which is a limitation of the technology itself.

The major challenges in using RFID for presence detection inside cleanrooms are vulnerability to human body, reliability of the passive tags and false classification of tags (tags that pass near to the exit but not directly through the exit are classified as true). The correct placement of the antenna and the tag will have to be determined to minimize occlusion of the human body in this case. The passive tags on the suit need to be able to withstand harsh conditions and continuous washing of the suits. The false classification problem will have to be addressed to get usable accuracy.

2.3 Related Work

Bottani et al [2] explored the potential of implementing UHF RFID technology to support EAS applications. They used two near field and two far field circular polarized antennas deployed at the exit at a width of 1.8m. They demonstrated that RFID performed well in most cases except when the tag is occluded by hand, booster bag or aluminium foil. One of the issues highlighted was the false alarm which could be triggered due to stationary tags in the surrounding region or if the tag is passing near to the exit but not directly through the exit. The proposed counter measure was to reduce the transmission power of the antenna.

Mathias et al [3] demonstrated that reducing the RFID EAS antenna power leads to diminished detection rate and thus potential undetected thefts. They tried to solve this false classification problem through data analytics technique. They deployed RFID readers at the exit and used different machine learning models with over 40 predictors based out of reader values to

classify if the tag has moved out of the gate. Their work shows promising results with an accuracy of 95% but has deployment challenges in the real world - if multiple people are exiting at the same time or if a person quickly walks through the gate, the detection may fail altogether.

Goller et al [4] combined computer vision with RFID to mitigate false positive observations and improve tag localization. They deployed three RFID readers in a line on the ceiling along with the camera towards the exit and used a probabilistic data association technique combining both measurements. They tested their deployment with multiple tags along with multiple people and demonstrated a robust system with respect to detection and suppression of false positives. However, they pointed out that in order for their system to work, it required a tremendous calibration procedure for the camera which may not be feasible in practice.

2.4 Our Approach

The RFID reader detects the tag value and its RSSI (Received Signal Strength Indicator) through the antenna the moment the tag enters the coverage area. The problem of possible occlusion of the human body between the tag and the antenna is solved through the proper placement of the antenna near the exit and the tag on the arm of the suit. The tag readings are then processed on the edge server for classification. [2] tried to solve the false classification of tags by reducing the transmission power while [3] used machine learning models. Our approach combines both of these by reducing the transmission power and using simple data analytics techniques. This works particularly well for the presence detection of entry / exit of the person in the cleanroom as the tag placement is fixed in contrast to EAS in [2] and [3]. The overall system architecture is presented in figure 2.1.

The UHF (Ultra High Frequency) RFID is used for our deployment which operates at the frequency band of 865 - 928 MHz. There are two types of antennas available for UHF RFID: linear polarization antennas and circular polarization antennas. Linear polarization occurs when electromagnetic waves broadcast on a single plane which requires the tag orientation to be fixed upon the same plane as the antenna. Circular polarized antennas emit electromagnetic waves in a circular fashion thus not requiring the same orientation of the tag. While the tag placement is fixed with respect to the antenna for presence detection of persons in a cleanroom, it is difficult

to get a perfect orientation of the tag due to variance in height of the person, the attachment of the tag to the white suit and the movement of the arm itself. Thus, circular polarized antennas are used to account for these inconsistency in tag orientation.

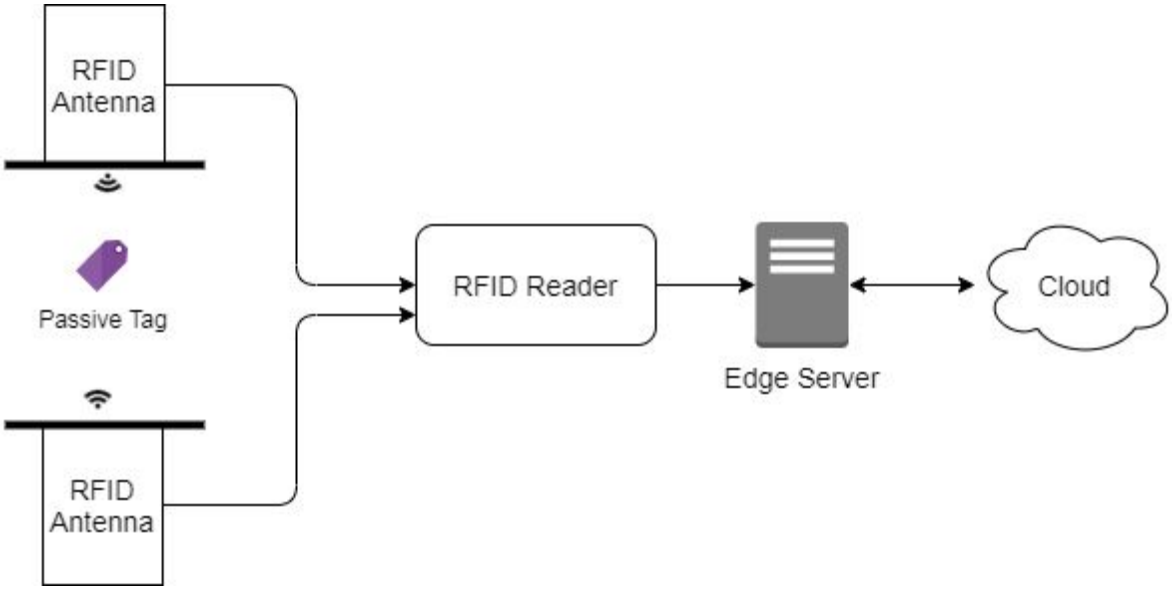


Figure 2.1: RFID System Architecture

2.4.1 Placement and Tagging

The RFID reader needs to detect the tagged person while entering / exiting the clean room minimizing occlusion between the tag and the reader along with reducing false positives. RFID does not require a clear line of sight however the human body attenuates the signal. As passive tags require a stronger signal strength to get energized and respond back, the attenuation of the signal will hinder detection. Considering this limitation, two antennas should be placed on both sides of the exit on a stand at a height of approximately 5.5 feet. This placement will be closest to the tag being nearest to the exit. This will also allow flexibility in controlling the coverage area of the antenna. The passive tag should be placed inside the placeholder of the right or left upper arm of the person. The antennas will be connected to the reader that will read the tags and detect the person passing by. The placement of the antenna at the exit and the possible movement paths are shown in figure 2.2.

There are RFID passive tags coated with different materials that can be used to withstand harsh usage and laundry. It is also possible to imprint RFID passive tags on wearables through screen printing. Screen printing is a versatile technique, which is already used in the electronics industry to print thick film structures like conductors, dielectrics and passive components on the fabric [5]. We have not experimented with these tags however they are commercially available and there is alot of scientific literature available on this topic.

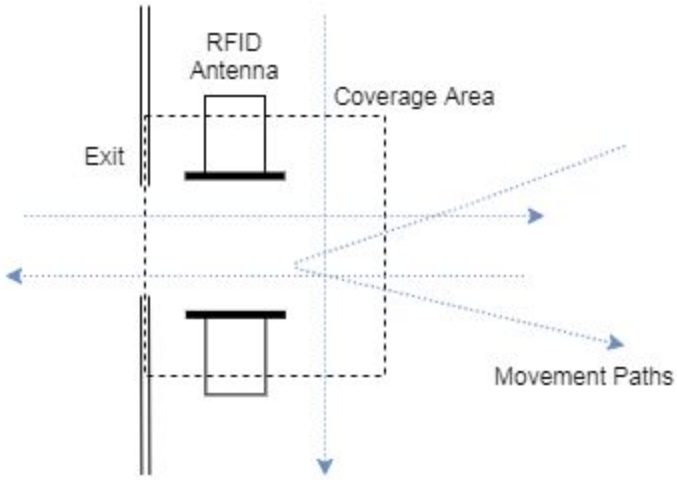


Figure 2.2: Placement of antennas with possible movement paths

2.4.2 Coverage Area

In order to reduce false positives, an optimal transmission power from the reader to the antenna has to be determined. The transmission power should be above a minimum threshold to reduce false positives without running into the issues of diminished detection rate [3]. Higher transmission power will result in a larger coverage area within which the tag will be detectable. Therefore, a balance between the both will have to be determined as done experimentally in section 2.5.2.

2.4.3 Classification

The problem of false classification of tagged people that pass near to the exit but not directly through the exit is mitigated through the classification algorithm. When the tagged

person enters the coverage area and leaves that zone, RSSI values received on the reader for the tag follow a bell curve distribution across time as seen in figure 2.3. This happens even if the person walks near the antenna and again returns back without passing through. The distribution also varies significantly based on the movement of the person. If the person stands inside the coverage zone for some time and then moves out, the RSSI distribution across the initial seconds will roughly stay the same and then peak in the last few seconds as the person walks out. The RSSI values also contain noise due to the relative orientation, elevation and the placement of the tag with respect to the antenna.

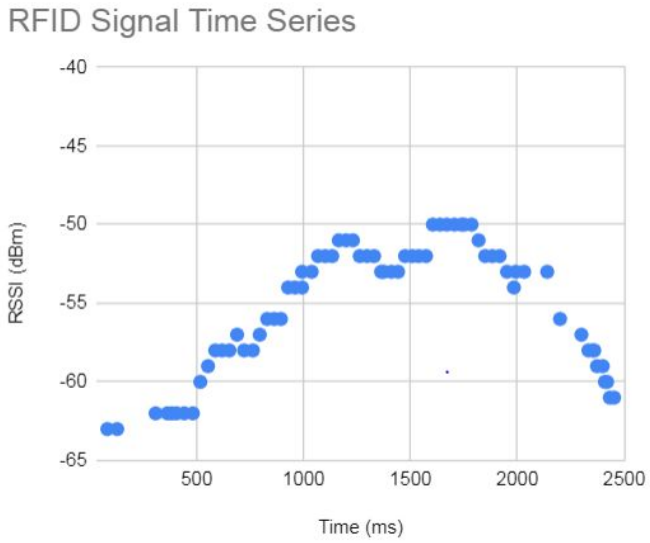


Figure 2.3: RFID Signal Time Series

When the person enters the coverage area, the reader continuously receives the tag values along with its RSSI and tracks them with respect to time. If there are no more readings for the tag since the last two seconds, the data for this tag is sent for classification to the edge server. The classification algorithm on the edge server first finds the peak with the maximum width. The python scipy signal peak detection algorithm is used for finding the peak [6]. This algorithm searches for peaks (local maxima) based on simple value comparison of neighboring samples and returns the peaks whose properties match the specified conditions for height, prominence, width, distance to each other etc. If no such peak is found, the tag is classified to be not passing

through the exit. If found, the one with the maximum width is selected and predictors such as the mean and standard deviation of the RSSI, read count, total read time etc. are calculated from its distribution and compared with experimentally determined thresholds. If these values pass the threshold, the tag is considered to have gone through the exit. The timestamp corresponding to the peak of the distribution is used for tracking. The hyperparameters and the predictors are determined and tuned through experiments. Algorithm 2.1 enumerates the classification algorithm that executes on the edge server once it receives the data from the RFID reader.

Algorithm 2.1 Classification

```
1:  constants: τHeight, τWidth, τProminence        : Peak detection hyperparameters
2:  tags ← rfidReader.getTags()
3:  for each t ∈ tags do
4:      rssiTimeSeries ← getRSSITimeSeries(t)
5:      peaks ← scipy.signal.find_peaks(rssiTimeSeries) with:
6:          height=τHeight, width = τWidth and prominence = τProminence
7:      if no peaks then:
8:          continue
9:      maxWidthPeak ← getMaxWidthPeak(peaks)
10:     tagPredictors ← getPredictors(maxWidthPeak)
11:     for each p ∈ tagPredictors do
12:         τP ← getPredictorThreshold(p.type)
13:         if p < τP then:
14:             continue
15:     classifyTagAsExited(t)
```

2.5 Experiment

2.5.1 Setup

The experiment is carried out in University of Illinois-Urbana Champaign, Siebel Center for Computer Science, lab 3113 SC. While the experiment is not performed in the cleanroom, the simulation is very similar and the results should not vary much. A Sparkfun Simultaneous RFID reader stacked on top of Elegoo Uno arduino compatible microcontroller is used. The reader is connected to the Laird circular polarized antenna (S8658WPL-T-01). The arduino

microcontroller is connected via USB to the laptop which acts as an edge server. A Python program that executes on the laptop interacts with the microcontroller / RFID reader through serial communication.

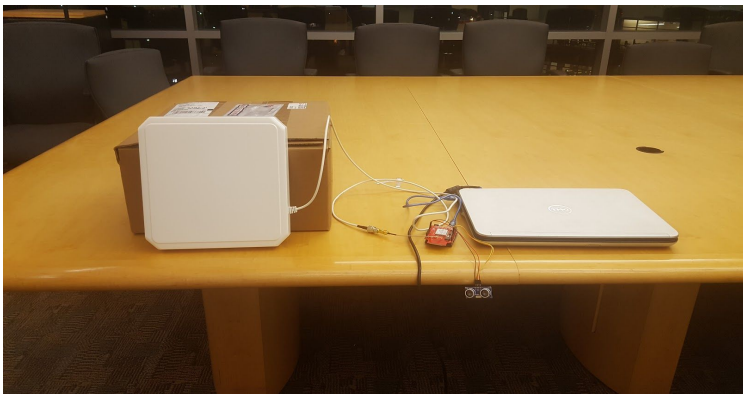


Figure 2.4: From left to right: Laird Circular Polarized Antenna (S8658WPL-T-01), Sparkfun RFID reader in red, Dell XPS Laptop.

2.5.2 Transmission Power - Coverage Area

The appropriate transmission power from the RFID reader to the antenna with respect to the coverage area has to be determined. With varying transmission power, a single RFID tag was placed around the antenna at different distances to determine the detectable range and maximum readability.

Tx Power	HRR	VRR
13 dBm	0.39m	0.91m
15 dBm	0.60m	1.37m
17 dBm	0.76m	2.1m
21 dBm	1m	3.5m
23 dBm	1.3m	4.9m

Table 2.1: Transmission power with read ranges

The horizontal read range (HRR) of the antenna corresponds to reading the tag at least 1 feet away from the reader in any direction. The vertical read range (VRR) is the ability of the

antenna to read the tag in front of it. Table 2.1 demonstrates the change in read range as the transmission power is increased. The transmission power of 17 dBm is appropriate to balance readability with coverage area. At this power level, the reader was able to detect the tag in almost all of the positions within the coverage area as the person is passing by. The classification algorithm mitigates the false positive reads within this coverage area of 2.1 x 0.76m (VRR of 2.1m and HRR of 0.76m).

2.5.3 Classification Hyperparameters and Predictors

The classification algorithm requires experimentally determined peak detection hyperparameters and predictors. These vary as per the equipment and so it is important to carry out few test runs to determine these values. We carried out close to 5 test runs walking through the exit zone and based on observations and the mean values from these runs, the classification parameters and predictors were determined as expressed in table 2.2 and 2.3.

Peak Detection Hyperparameter	Value
Peak Detection Minimal Width	1000ms
Peak Detection Minimal Height	-53 dBm
Peak Detection Minimal Prominence	10 dBm

Table 2.2: Hyperparameters used for peak detection

Predictors	Minimum Value
Mean RSSI	-60 dBm
Standard Deviation RSSI	1.3 dBm
Read Count	30
Total Read Time	1.5s

Table 2.3: Predictors used for classification

2.5.4 Test Setting

There are 40 test runs with a person manually carrying the tag inside a placeholder and walking at different speeds. The person always enters the coverage area however only exits in half of the test runs. Some of the tests also involve two people passing by right one after the other. However, as tags are dealt individually, it does not impact the results. Generally, only one person can exit at a time due to the size of the exit door. The placeholder is on the right hand of the person as the antenna is placed to the right side. While the experiment is carried out with a single antenna covering one side, the results also hold true for the other side. A sample reading for a single person walking through the zone has a read count of 100, mean RSSI of -60dBm, standard deviation RSSI of 3.3dBm for a total read time of 2.7 seconds.

2.5.5 Results

	Predicted Exited	**Predicted Not Exited**	
Actual Exited	17	3	20
Actual Not Exited	4	16	20
	21	19	

Table 2.4: Confusion matrix for classification of whether a person has exited or not

The classification results for the 40 test runs are shown in table 2.4. The TP (True Positives) are 17, where the person actually exited and our algorithm classified this correctly. The FN (False Negatives) are 3, where our algorithm classified the reading as not exited while the person actually exited the room. This happened when the person either almost ran through the exit, or managed to occlude the tag and the reader by moving the arm. The TN (True

Negatives) are 16, where the person did not exit and this got classified correctly. This is important as the cases where the person walked parallel within the coverage area along with the ones where he barely entered and returned all got correctly classified as negative. With the TN being 16, the FP (False Positives) are only 4, where the person did not exit and yet got classified as exited. This only happened when the person came very close to exiting and then again returned in a similar motion. The total number of false positives of 4 is drastically low with our classification technique. The metrics such as precision, recall, accuracy and f1-score for the confusion matrix are given by:

$$Precision = \frac{TP}{TP + FP} = 0.80 \tag{2.1}$$

$$Recall = \frac{TP}{TP + FN} = 0.85 \tag{2.2}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = 0.825 \tag{2.3}$$

$$F1\ Score = 2\ \frac{Precision * Recall}{Precision + Recall} = 0.82 \tag{2.4}$$

2.6 Future Work

The system hasn't yet been tested in an actual cleanroom with the tag being placed on the white suit. This will also entail the use of the wearable or washable RFID passive tags and evaluating the life of the tag when used within the cleanrooms. The deployment of the RFID presence detection system in cleanrooms will really help evaluate the feasibility of this implementation.

CHAPTER 3: MOVEMENT TRACKING AND PRESENCE DETECTION WITH BLE

3.1 Introduction

Bluetooth Low Energy (BLE) is a form of wireless communication which allows devices to communicate with each other. BLE is designed for short data exchanges considerably reducing power consumption in comparison to classic Bluetooth. There are beacons that broadcast data with contextual information at regular intervals that get detected by BLE devices in the range triggering specific actions [7]. BLE is integrated in most of the mobile smartphone devices thus making it usable for various applications. Due to low battery consumption, periodic data transfer and long range of upto 100m, BLE can be used for continuously tracking the movement of a person. The mobile device of a person can listen for advertisements from beacons that continuously transmit unique identifiers. The location of the device can then be estimated through the received signal strength indicator (RSSI) and the position of these beacons. This can be used for movement tracking across multiple cleanrooms.

It is also possible to use RFID passive tags for movement tracking although it will require deployment of multiple antennas covering the area. Even within a single room, multilateration can be done through three or more RFID antennas for positioning a passive tag. However, the occlusions blocking the signal will hinder detection and eventually positioning. Moreover, the range of RFID passive tag is only upto 7m while BLE goes upto 100m. While passive RFID has limitations for movement tracking, BLE can be used for presence detection as it is able to determine the location of the person.

3.2 Challenges

The BLE localization accuracy is significantly affected due to slow and fast fading which is the fading of signal amplitude over distance. Slow fading occurs as structural obstructions give rise to a deterministic shift in the signal strengths. Fast fading or multipath effect occurs when

signals reach a receiver via many paths and their relative strengths and phases differ thus causing fluctuations in the RSSI.

The major hindrances in using BLE for cleanrooms are physical obstacles that will block or attenuate the signal and interference in the channel due to other electromagnetic waves operating at the same bandwidth. The environment of cleanrooms is different as it contains equipment made from different materials some of which are metals. As metals are conductors, it absorbs electromagnetic waves thus attenuating the BLE signal or even creating a dead zone. Dead zones are locations which do not receive any signals as it is in the shadow of the metal object which in the case of cleanrooms can be equipment. BLE operates on 2.4 Ghz bandwidth which is the same as the one used by other technologies such as wifi, zigbee, microwaves etc. In cleanrooms, there are other devices and equipment operating at this frequency that can hamper with BLE. This can cause radio frequency interference thus causing latency and data loss. While BLE utilizes channel switching to avoid interference, the propagation of the signals will still be impacted if there is too much congestion. The delay in receiving the BLE advertisements will impact the localization accuracy.

3.3 Related Work

De Blas et all [8] proposed two post-processing filters to improve indoor positioning through trilateration. They were able to achieve an accuracy of 3.98m with three BLE beacons placed at a distance of about 7m and a phone acting as a reader. While their enhancements brought some improvements, they measured their accuracy only with a stationary device.

Thaljaoui, Adel et all [9] proposed a model for indoor BLE positioning which implemented the inter ring localization algorithm (iRingLA). iRingLA is an alternative solution to trilateration which draws rings instead of circles around the beacons. The radius of the rings are computed using RSSI measurements found through calibration. They used three BLE beacons placed at a distance of 4m and a phone as a BLE reader and tested in a two-dimensional grid with a clear line of sight obtaining an accuracy of 0.4m. While the iRingLA seems like a good alternative for trilateration, it still needs testing in a practical deployment.

Röbesaat, Jenny, et al [10] carried out extensive experiments for BLE localization and used multilateration with least squares regression to estimate user's position. They used eight beacons in a corridor of 15 x 3 metres and were able to achieve a localization accuracy of 1m. This was achieved with a combination of corrective functions and additional sensor values from the users device such as the accelerometer, orientation sensor etc.

3.4 Our Approach

There are multiple BLE beacons placed throughout the cleanrooms that will be constantly advertising. These beacons are placed in positions avoiding a near-complete overlap of the regions within their ranges. The coordinate system across the region is defined and the positions of the beacons are recorded. The person's mobile device will be listening for advertisements from the beacons and the RSSI from the advertisements along with its respective beacon coordinates will be used to estimate the person's location. The estimated location on the mobile device can be sent to the cloud as required. Figure 3.1 shows a sample placement of beacons across cleanrooms and the circles demonstrate the idea behind multilateration for position estimation.

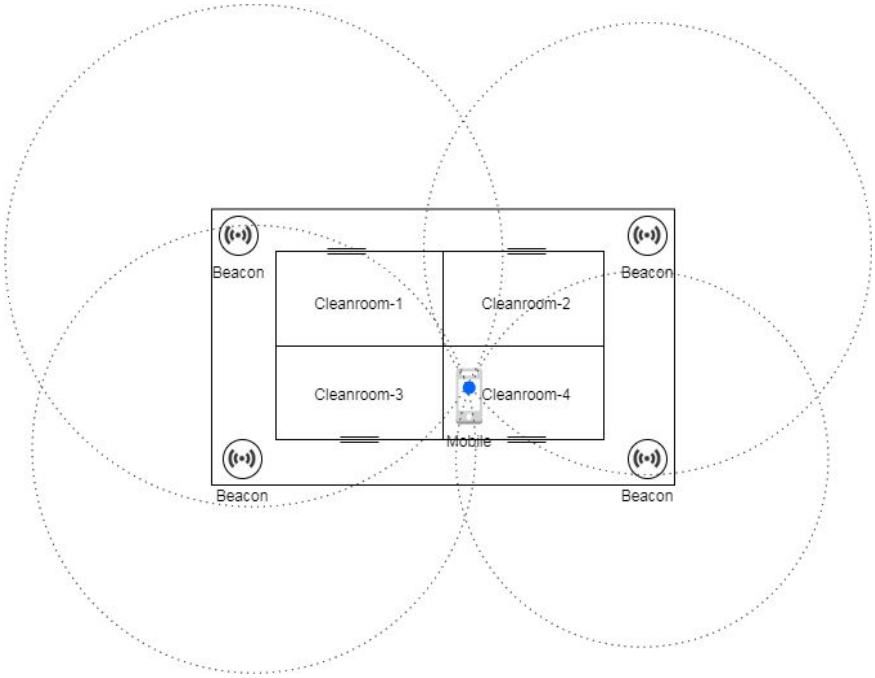


Figure 3.1: BLE localization in cleanrooms

The received signal strength (RSSI) values and the coordinates of the beacons received through advertisements on the mobile device will go through the following pipeline to estimate the location. The pipeline executes on the persons mobile device the moment an advertisement is received by any of the beacons.

1. The RSSI value for each beacon is first passed through the Kalman filter to reduce noise.
2. The normalized RSSI values are then used to determine the distance of the user from each reader using the log-distance path loss model.
3. These distances along with its corresponding beacon coordinates are then fed into a Non-Linear Least Squares Solver to perform multilateration and get the estimated user position.

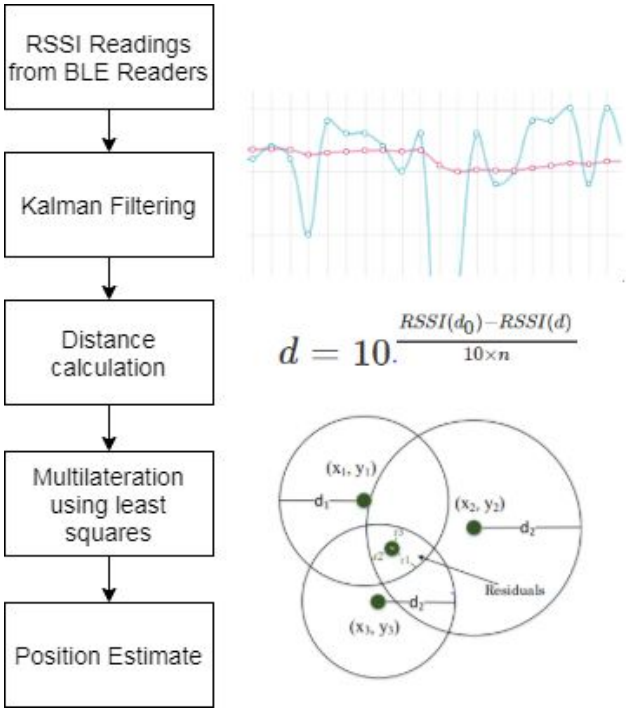


Figure 3.2: BLE Processing Pipeline

3.4.1 Kalman Filtering

The RSSI values are heavily influenced by the environment and have consequently high levels of noise caused by multipath reflections, interference and fading. Kalman filter is a state

estimator that makes an estimate of some unobserved variable based on noisy measurements. It is a recursive algorithm which takes the history of measurements into account. A one-dimensional linear Kalman filter similar to [10] is used for RSSI filtering to estimate the true RSSI value based on the measurements minimizing the noise in it.

For Kalman filtering, we assume that there is a random change in RSSI with no transformation and no control unit. Thus, the measurement z_t (e.g. RSSI received from a signal) at a given point in time *t* is given by its actual state u_t with some noise δ_t:

$$z_t = u_t + \delta_t \tag{3.1}$$

The goal of Kalman filtering will be to estimate x_t which is as close to u_t given the process noise Q, measurement error *R* and the current measurement z_t. The process noise is a constant describing the noise caused by the system itself. The measurement error is also a constant which corresponds to the noise in the actual measurement (e.g. variance in RSSI). Kalman filtering has two phases: the prediction phase and the update phase. With a simple transition model and a static system, the prediction phase is given by:

$$\bar{x}_t = x_{t-1} \tag{3.2}$$

$$\overline{P}_t = P_{t-1} + Q \tag{3.3}$$

where

$\bar{x}_t$ is the prediction without incorporating measurement

x_{t-1} is prior predicted value

$\overline{P}_t$ is the predicted error covariance

P_{t-1} is prior error covariance

Q is the process noise

The error covariance defines the certainty of measurement error. The higher value of P_t implies that the measurement cannot be trusted due to noise. Using this, a Kalman gain function can be derived which is the amount of correction applied by the filter to the incoming measurements to make them less noisy. It is the relative weight given to the measurements and

current state estimate. Based on the Kalman gain, the final prediction of the system and the actual error covariance can be computed. This defined as part of the update phase as:

$$K_t = \frac{\overline{P}_t}{(\overline{P}_t + R)} \tag{3.4}$$

$$x_t = \bar{x}_t + K_t (z_t - \bar{x}_t) \tag{3.5}$$

$$P_t = (1 - K_t) \overline{P}_t \tag{3.6}$$

where

K_t is the Kalman Gain

$\overline{P}_t$ is the predicted error covariance

R is the measurement error

x_t is the predicted value

$\bar{x}_t$ is the prediction without incorporating measurement

z_t is the measurement

P_t is the error covariance

As the number of iterations of the Kalman filter increases, Kalman gain increases as well and consequently the error covariance decreases. After some iterations, both the Kalman gain and the error covariance converge to a value being dependent only on constants. At this point, the Kalman gain reaches the maximum amount of correction that it can provide to the incoming noisy measurements. This implies that the trust on the measurement is high as the noise from it is being minimized. A measurement error of 3 and a process noise of 0.125 is used as constants in our implementation determined experimentally.

3.4.2 Distance Calculation

The standard log-distance path loss formula to compute the estimated distance of the user from the RSSI value:

$$d = 10^{\log_{10} \frac{RSSI(d_0) - RSSI(d)}{10 \, x \, n}} \tag{3.7}$$

where RSSI (d_0) is the received signal strength measured at 1 meter, RSSI (*d*) is the received signal strength at distance *d* and n is the path loss index which depends on the transmission medium, the emitter and the receiver. RSSI (*d*) before being used in the log distance path loss formula goes through the Kalman filter. The initial value corresponds to measurement value z_t of Kalman Filter in equation (3.1) which then gets replaced by the predicted value x_t. The value for n is calculated during the calibration.

3.4.3 Position Estimation through Multilateration

The position estimation problem can be formulated as follows: A node N has determined the distances to three (or more) other nodes A, B, and C. The distances from node N to each of the other nodes are a_r , b_r , c_r . The coordinates of other nodes are : A = (a_x , a_y), B = (b_x , b_y) and C = (c_x , c_y). A circle can be drawn around the nodes A, B and C with the radius equal to the estimated distances to N. The trilateration problem is to find the coordinates of node N = (n_x , n_y) which will lie at the intersection of these circles from the given information. This is demonstrated in figure 3.3. The same problem is extended for multilateration where the number of nodes is greater than three.

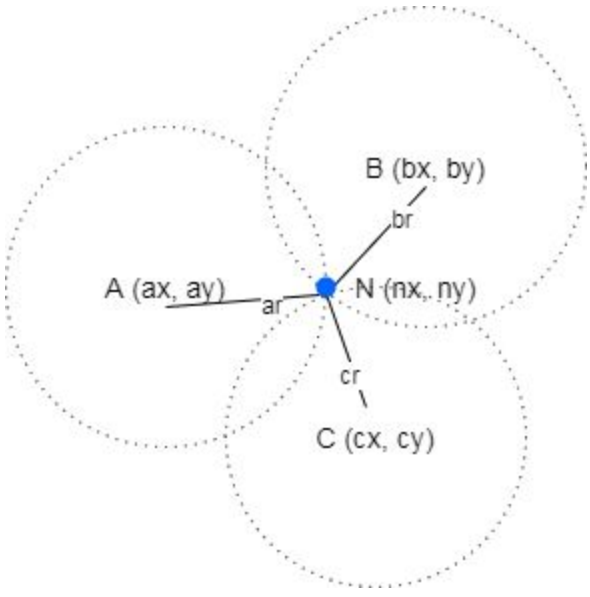


Figure 3.3: Trilateration with perfect intersection

The complicating factor is that the known nodes coordinates and distances typically include measurement errors. This can result in the circles not intersecting at a single point or not intersecting at all as shown in figure 3.4. The three commonly used methods of solving multilateration are circle intersections with clustering (closed-form geometry), iRingLA and nonlinear least squares.

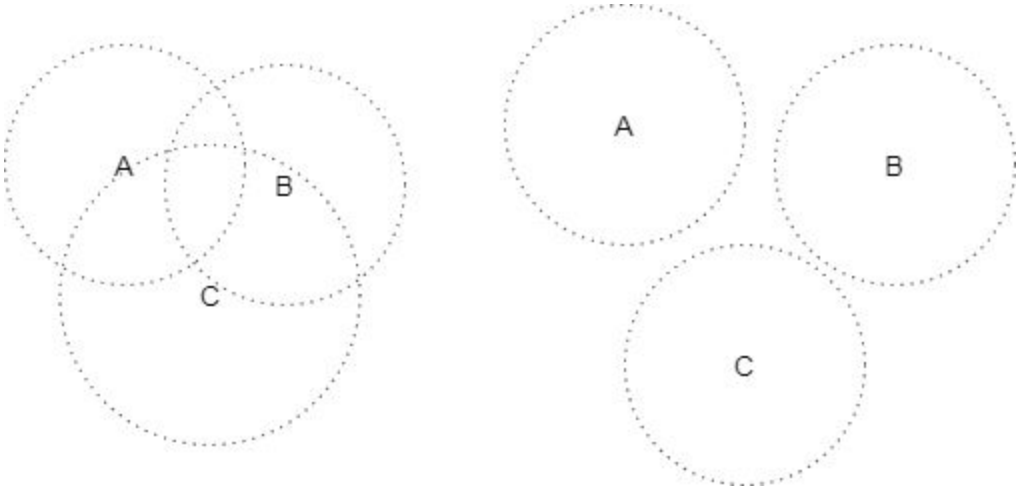


Figure 3.4: Trilateration with Noise

Circle Intersection with Clustering (Closed Form Geometry)

A circle is drawn with radius equal to the estimated distances between the nodes A, B and C and unknown point N. There is a common intersection point between each of the three circles as observed in figure 3.3. From this given information, the system of equations can be written as:

$$(n_x - a_x)^2 + (n_y - a_y)^2 = a_r^2 \quad (3.8)$$

$$(n_x - b_x)^2 + (n_y - b_y)^2 = b_r^2 \quad (3.9)$$

$$(n_x - c_x)^2 + (n_y - c_y)^2 = c_r^2 \quad (3.10)$$

Subtracting (3.9) from (3.8) and expanding, rearranging the equations:

$$(-2a_x + 2b_x)\, n_x + (-2a_y + 2b_y)\, n_y = a_r^2 - b_r^2 - a_x^2 + b_x^2 - a_y^2 + b_y^2 \quad (3.11)$$

Subtracting (3.10) from (3.9) and expanding, rearranging the equations:

$$(-2b_x + 2c_x)\, n_x \; + \; (-2b_y \; + \; 2c_y)\, n_y \; = \; b_r^2 \; - \; c_r^2 \; - \; b_x^2 + \; c_x^2 \; - b_y^2 + \; c_y^2 \qquad (3.12)$$

(3.11) and (3.12) represent a system of two equations with two unknowns and can be written as:

$$A \; n_x + \; B \; n_y \; = \; C \qquad (3.13)$$

$$D \; n_x + \; E \; n_y \; = \; F \qquad (3.14)$$

The solution for n_x and n_y becomes:

$$n_x = \; \frac{CE - FB}{EA - BD} \qquad (3.15)$$

$$n_y = \; \frac{CD - AF}{BD - AE} \qquad (3.16)$$

While the closed form solution can be computed right away, it cannot work when the circles do not perfectly intersect as shown in figure 3.4 as it will invalidate the equations (3.8), (3.9) and (3.10). This happens frequently due to measurement errors in RSSI-distance calculation caused by noise and fading of signals.

iRingLA

Thaljaoui, Adel, et al [9] demonstrate the iRingLA algorithm. Instead of using circles, they propose using rings with a radius obtained experimentally considering the RSSI-distance error. The centroid of the intersection of the rings is the position estimate. While iRingLA is an improvement over the geometric cluster closed form, in practice it will still not work if one or more of the rings do not intersect. This problem can be solved by increasing the radius of the rings. However, the intersection of the cluster will then become large enough such that the centroid calculation may not necessarily be the most accurate estimate. The equation for the radius of the ring is given as:

$$R_{in} = d \; - \; E \qquad (3.17)$$

$$R_{out} = d + E \tag{3.18}$$

where

R_{in} is the inner radius

R_{out} is the outer radius

d is the distance from the beacon to the receiver computed using the standard log distance path loss formula from section 3.4.2

E is a precomputed average error between the actual coordinate and the estimated coordinates at different distances between the beacon and the receiver

Nonlinear Least-Squares

The distances between the point N and the nodes is never perfectly equal to the computed distances through RSSI as observed in figure 3.4. The difference between the actual distance and the computed distance will be the residual values and the goal will be to find a point which will minimize these residual values [11]. Based on figure 3.3, the equations can be written as:

$$(n_x - a_x)^2 + (n_y - a_y)^2 - a_r^2 = a_\Delta^2 \tag{3.19}$$

$$(n_x - b_x)^2 + (n_y - b_y)^2 - b_r^2 = b_\Delta^2 \tag{3.20}$$

$$(n_x - c_x)^2 + (n_y - c_y)^2 - c_r^2 = c_\Delta^2 \tag{3.21}$$

.....

This is a system of three or more equations with two unknowns (n_x , n_y). Since there are more equations than unknowns, the system is overdetermined, and in general there isn't a unique solution but there is a least squares one. The equation has been rewritten where the zeroes on the right hand sides have been replaced by nonzero residuals. The least squares solution is the unique solution (n_x , n_y) that minimizes the sum of the squares of the residuals ($a_\Delta^2 + b_\Delta^2 + c_\Delta^2 + \ldots$). While the least squares solution is computationally more intensive than the others, it will work even if the circles do not intersect at all. Thus, nonlinear least squares is used in the implementation to solve the multilateration problem. The Levenberg-Marquardt algorithm from apache commons library is used to solve this [12].

3.4.4 Presence Detection

BLE can also be used for presence detection and measuring when a person has entered or exited the clean room. As the coordinates for a moving object or a person can be obtained through multilateration, the distance between the desired location and the person can be used to evaluate whether the person is present inside the zone. The accuracy of BLE however will impact the accuracy of presence detection because of false positives and false negatives.

3.5 Experiment

3.5.1 Setup

The experiment is carried out in University of Illinois-Urbana Champaign, Siebel Center for Computer Science, lab 3113 SC in an area of 8 x 6 meters. While not a cleanroom, the experiment will still provide valuable insights into the accuracy of the implementation. The user is carrying an android smartphone running our localization mobile app with Bluetooth switched on. The smartphone which is the BLE reader is always held in the hand of the person. Android supports four different scan modes [13] and the one with the highest sampling rate which is scan_mode_low latency is used.

Four android mobile phones are used as beacons and placed along the edges of the room. The coordinate system is defined in meters and the positions of the beacons are recorded as shown in figure 3.5. The beacons use eddystone format which is an open beacon format for advertising a unique id. Android supports four different transmission powers and three frequency modes [14]. The lowest transmission power of ~ 75dBm is sufficient for our room. Even with the highest frequency mode of low latency (10 Hz), the sampling rate at the receiving mobile device is only 1 reading per second.

3.5.2 Test Setting

For evaluating movement tracking, the person is walking in the room at a reasonable pace of approximately 1.3 m/s. The actual coordinates for 5 different positions of the room are known. The moment the person enters one of these positions, its estimated coordinates from the RSSI

BLE algorithm is recorded by manually tapping the button on the android app. There were a total of 20 such recorded values. These values are then compared with the actual coordinates to measure the accuracy.

In order to evaluate presence detection, a similar approach to movement tracking is used. The actual (theoretical) centroid of the room is measured and a threshold of 1.3m is used to evaluate if a person entered this or not.

3.5.3 Results

Actual Coordinate (x, y, z)	Estimated Coordinate (x, y, z)	Distance (m)
(1, 1, 2)	(2, 2.93, 2.01)	2.17
(1, 5, 2)	(3, 3.79, 1.99)	2.33
(3, 1, 2)	(4.02, 3.79, 2.2)	2.97
(3, 5, 2)	(2.82, 3.82, 1.99)	1.19
(3, 4, 2)	(2.4, 2.7, 2)	1.43

Table 3.1: Movement Tracking Readings

A sample of five readings for the estimated coordinates along with its corresponding actual coordinates is shown in table 3.1. The coordinate system along with the room is the same as shown in figure 3.5. On comparing the user's actual position with the position estimated through multilateration for 20 such readings, the average difference, which is the localization accuracy is about 2.1m.

$$Accuracy = \frac{\sum_{1}^{n} Distance\,(Actual\ Position - Estimated\ Position)}{n} = 2.1\text{m} \qquad (3.22)$$

One of the major factors impacting the accuracy in our case is the measurement error. We did not have the exact dimensions of the room and so we had to manually measure it. This would have easily introduced a discrepancy of at least 0.5m while tracking the coordinates of the

beacons and the evaluation points in the room. The orientation of the person and the placement of the mobile phone such as whether it is inside the pocket or not also matters. However, with enough variations of the test, the average accuracy remained at 2.1m. The accuracy of our implementation is consistent with those of the previous works [8][10] without taking into account further enhancements. This implies that the movement tracking in cleanrooms with BLE in the best case will have a localization accuracy of 2.1m with this implementation. This can definitely improve with more beacons, precise coordinate measurement and consistent hardware devices for beacons.

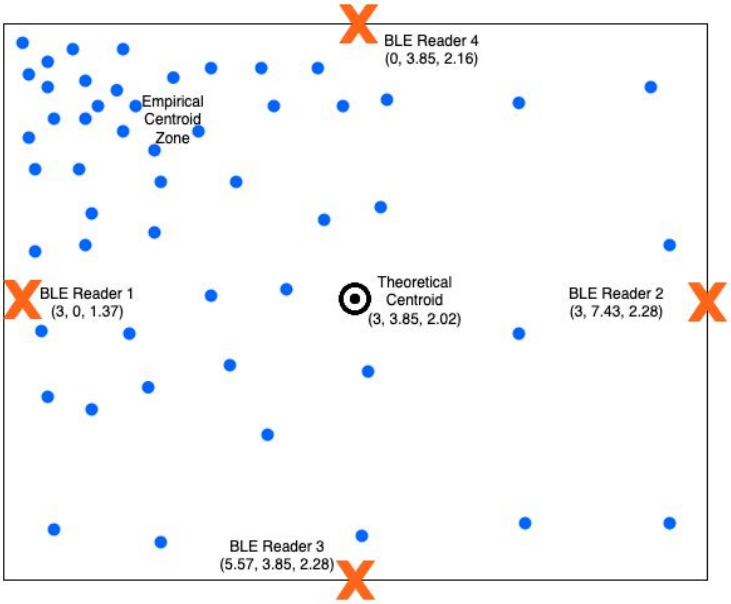


Figure 3.5: BLE Theoretical Centroid Vs Empirical Centroid

For presence detection, the distance between the estimated coordinate and the theoretical centroid is compared and if it is less than 1.3m, the position is recorded. Figure 3.5 shows the highlighted points which are recorded for 50 positions near the theoretical centroid. The impact of slow fading can be observed in this case as the highlighted points are not near the actual centroid but have a consistent shift to the top left. Instead of using theoretical coordinates for presence detection, we propose using empirical coordinates. The empirical coordinate can be measured as the person is standing at the desired location and the position estimate through BLE

RSSI is recorded. The average of these recorded estimated positions act as the empirical coordinate. In the case of our experiment, the distance between theoretical and empirical centroid is approximately 1.5 m.

The 50 positions from figure 3.5 are used which are recorded near the theoretical centroid. The distance between the estimated coordinate and the empirical centroid is measured and if it is within 1.3m, it is classified within the zone. With this in mind, about 31 out of 50 positions were classified correctly as within the zone yielding an accuracy of 0.62. This is less than the presence detection accuracy of RFID which is 0.825.

3.6 Future Work

The overall accuracy of BLE localization can be improved. The Kalman filtering can account for the users movement through additional sensors such as accelerometer from the users smartphone device as covered in [10] thus reducing noise even more. The position estimates can also be improved by post-processing it through corrective functions [8]. With Bluetooth 5.1 which has direction finding features, the localization accuracy is supposedly down to centimeters.

BLE for presence detection simply considered the weighted euclidean distances between the empirical coordinates and the estimated positions. This can be improved by tracing the path instead of checking for discrete values.

CHAPTER 4: PROXIMITY SENSING WITH COMPUTER VISION

4.1 Introduction

Computer vision based processing can provide more fine grained information of the environment where the traditional sensors fail to capture such details or be too cumbersome to deploy. In order to understand which person is working on which equipment in a cleanroom, BLE accuracy of ~1.3m is not enough as the equipment are even closer to each other and deploying low range RFID antennas near each equipment is infeasible. This problem of proximity sensing which measures the relative proximity of a person to different areas of interest within a room can be solved through computer vision.

In order to construct the world model from the camera's perspective and localize a particular object within it, it is necessary to obtain the depth information. The distances in the 2D image is not sufficient to reflect the actual proximity of the objects within the frame. The depth information can be recovered with the use of two camera's or a stereo camera with two or more lenses through binocular disparity. The depth cues can also be estimated with a single monocular camera through calibration, linear perspective, shading, occlusions, textures etc. As most of the cleanrooms already have a security camera installed, trying to replace all of them with stereo cameras will be cost prohibitive with additional deployment challenges. Thus, proximity sensing will have to be done with a single monocular camera through 2D to 3D conversion and tracking the movement of the person across frames. This can be used to evaluate whether a person is working on an equipment or not inside a cleanroom.

4.2 Challenges

The problems facing vision based processing are different from traditional sensors. Vision processing is significantly impacted due to change in lightning, occlusions, shading and reflection. The background and objects inside the frame also influence detection and tracking. In the case of cleanrooms, both the person wearing the suit and the background is white. Thus, it is

difficult to extract specific feature descriptors. As everyone is wearing white suit, it is not possible for vision to determine the identity of the person. The position of the camera plays an important role in getting complete coverage of the room. However, due to huge equipment, this coverage may be blocked warranting the use of more than one camera. Due to the closeness of the equipment and the potential of occlusion due to other people and equipment, measuring proximity sensing may not always be possible.

4.3 Related Work

Duan et all [15] combined camera and RFID for object tracking where the moving objects were tagged with passive tags with the camera mounted on the top and the RFID antenna having a complete field of coverage. The camera was calibrated for 2D to 3D conversion and used blob detection along with optical flow for tracking. This was combined with RFID data through a fusion algorithm matching trajectories with phase shift. While the results had an overall accuracy of 10mm, the experiment was carried out on a table of 400 x 800 cm^2 with small objects having 2D linear trajectories. For localization in a real world environment, the movement is going to be in 3D and it is very difficult for a single antenna to have a complete field of coverage.

Wang et all [16] used active RFID tags placed around the room with a camera to localize the people holding the mobile RFID reader within the room. The readings from the tags were used to obtain an overlapping region where the person could be standing. A simple coordinate transformation was done on vision positioning to get the 2D position within the room. The fusion algorithm was then used to combine both observations resulting in an accuracy of 91%. The coordinate transformation works very well for 2D localization but will not work as accurately for proximity sensing as the distances in 2D will not reflect the actual proximity between objects.

Llorca, David Fernández, et al [17] used a combination of two RFID antennas, two BLE readers and two stereo vision for tag association with people in outdoor environments. Besides RSSI-distance calibration, the rest of the setup parameters were automatically calculated. They were able to correctly associate the tag to the person with an accuracy of 93.7% and also identify

the individual with the help of BLE. The application environment was outdoor and accuracy was achieved mainly due to the combination of multiple sensor values.

4.4 Our Approach

The cleanroom needs to have a camera mounted on the ceiling with maximum field of view of the room, especially the floor. The specific camera used will have to be calibrated for intrinsic and extrinsic parameters as described in section 4.4.4. The video segments from this camera need to be sent to a server for processing. The server will run the vision algorithm on the video segment to evaluate at each frame whether a person is working on an equipment or not. The results which will contain the timestamp and the equipment being worked on can be stored inside a database. As everyone within cleanrooms has to wear a white suit, it is difficult to distinguish one from another. Thus, it is only possible to evaluate whether someone without knowing the identity is working on a specific equipment or not. The architecture for vision based processing is shown in figure 4.1.

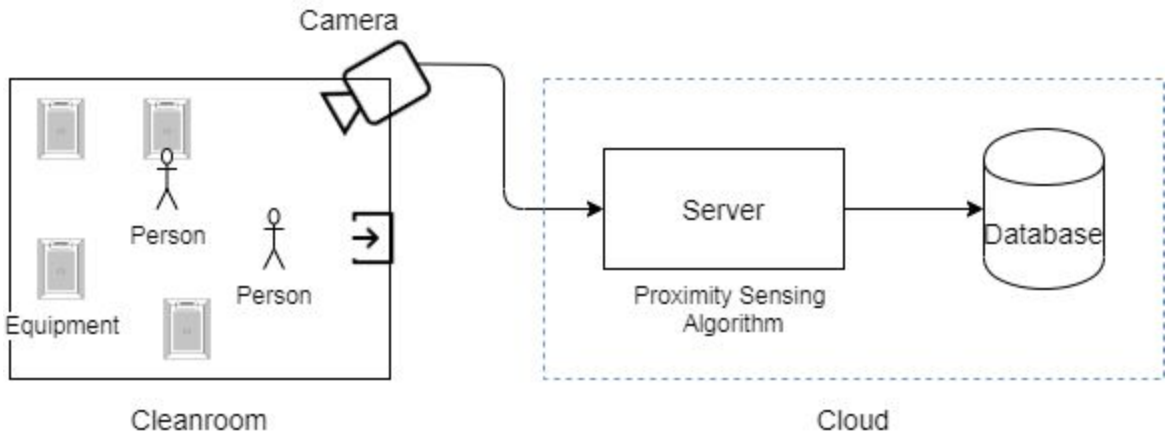


Figure 4.1: Vision based Proximity Sensing in cleanroom

The vision processing algorithm first performs background-foreground subtraction to highlight the blobs of the difference in the foreground and the background frame. The contour detection will then select all of the blobs that will correspond to persons. The floor coordinates of

the blobs will be used for tracking and checking the nearness to an equipment. The coordinates will all be converted from 2D to 3D which should maintain the relative distances in the room. The blob will either be a new person or will be assigned to the previously detected person based on distance. The floor coordinates of the equipment in the 2d image need to be manually selected. The proximity measurement in 3D between the person and the equipment will be used to sense the specific equipment that he is working on. The overall pipeline for the algorithm is described in figure 4.2.

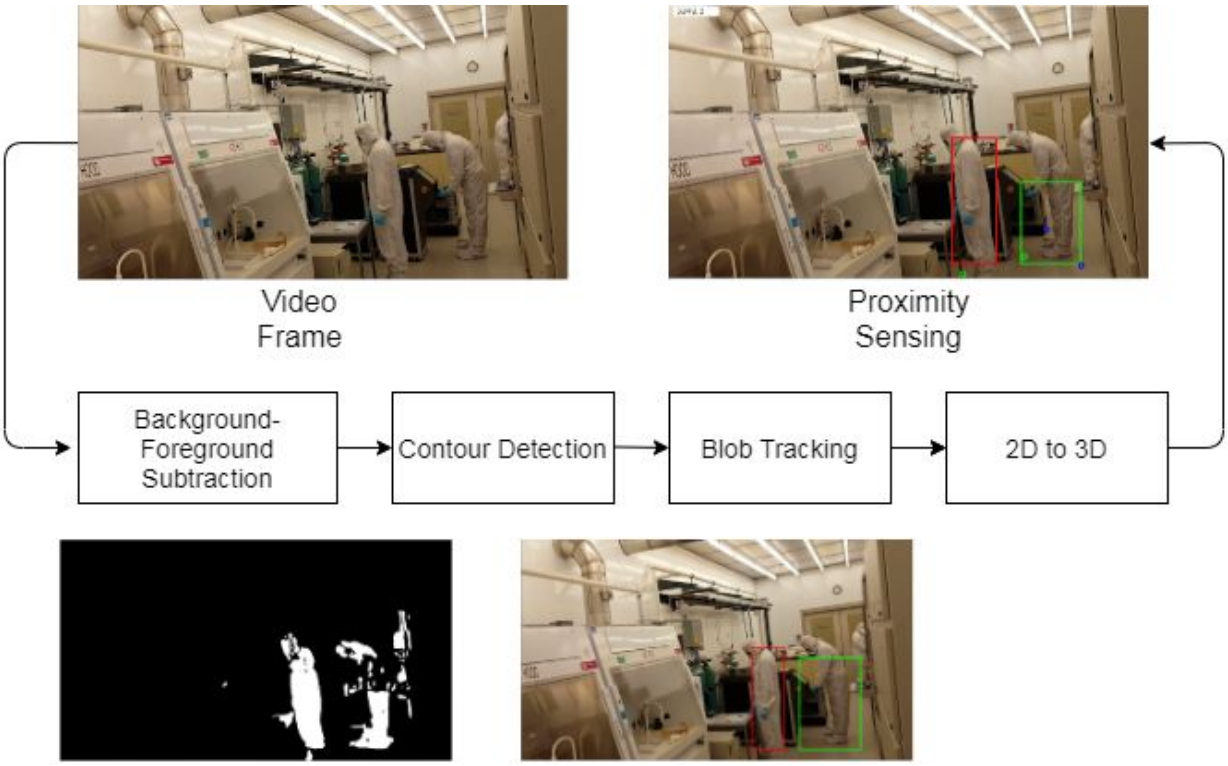


Figure 4.2: Processing Pipeline for Proximity Sensing

4.4.1 Background Foreground Subtraction

Figure 4.3: Background Image for a cleanroom

The background-foreground subtraction is used to isolate the person from the cleanroom. All of the frames inside the video segment including the background image are resized, smoothed and converted to grayscale. The resizing of the frame makes subsequent processing faster and smoothing reduces noise caused due to slight changes in lightning. The frames from the video segment are then subtracted from the background image setting each pixel value to the absolute difference. The difference in the value of each pixel is compared to a threshold and if the value is greater, the pixel is set to white color or else it is set to black. This binary mask image is then used for further processing.

Background	Foreground
	

Figure 4.4: Background - Foreground Subtraction

Algorithm 4.1 Background Foreground Subtraction

```
1:  global variables
2:      cv ← python cv2
3:  end global variables
4:
5:  function resizeGraySmoothFrame(f) :
6:      constants: fWidth, fHeight, kernelSize
7:      f ← cv.resize(f, (fWidth, fHeight), interpolation = cv.INTER_AREA))
8:      f ← cv.cvtColor(f, cv.COLOR_BGR2GRAY)
9:      f ← cv.GaussianBlur(f, (kernelSize, kernelSize), 0)
10:     return f
11: end function
12:
13: function backgroundForegroundSubtraction(bg, f) :
14:     constants: τDiff
15:     f ← cv.absdiff(bg, f)
16:     binaryMask ← cv.threshold(f, τDiff, 255, cv.THRESH_BINARY)
17:     return binaryMask
18: end function
```

4.4.2 Contour Detection

A blob is defined as a binary large object within an image which in the case of the binary mask image of cleanroom is characterized by a region containing white pixels. The continuous boundary points along these blobs having black pixels on either side are contours. The contour detection algorithm [18] treats all of the continuous white pixels as a single blob and returns the boundary points for each blob. The area of the contour is used to filter blobs that can correspond to a person. We used an area of 4000 pixel square on the frame size of 960 x 540 which is determined by taking a sample from one of the frames. The area corresponding to the person is dependent on the camera position. A bounding rectangle is then drawn along each of these blobs from its contour points taking the extreme x and y coordinates for the top left, top right, bottom left and bottom right. This ignores blobs that can be due to shadows and reflections. The contour detection along with bounding rect for the binary mask image is shown in figure 4.5.

Algorithm 4.2 Contour Detection

```
1:  global variables
2:      cv ← python cv2
3:  end global variables
4:
5:  function getContours(binaryMask) :
6:      constants: τP        : Threshold area for the person
7:      cnts ← cv.findContours(binaryMask)
8:      for each c ∈ cnts do
9:          if cv.contourArea(c) < τPerson then:
10:             cnts.remove(c)
11:     return cnts
12: end function
```

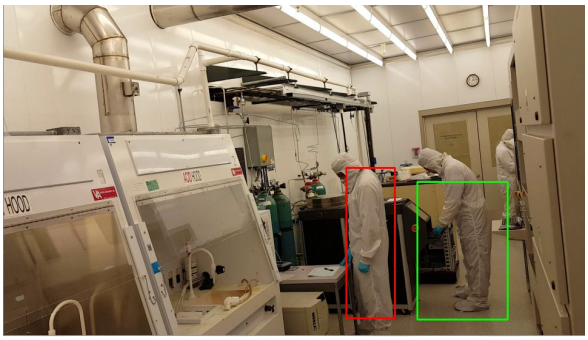
Binary Mask	Contour Detection
	

Figure 4.5: Contour Detection

Sometimes the blob corresponding to the person itself may split into two or more parts due to black pixels in between caused due to the imperfection of background-foreground subtraction. In this case, either the blobs may be ignored altogether due to the area being less than the threshold or it may result in 2 - 3 allowable blobs for the same person. The blob nearest to the previous blob of the person measured through 3D distance with some threshold as described below in section 4.4.4 is used to update the location of the person. If there are no blobs matching this criteria, the location does not get updated. In the case of a split, the closest blob is

used to update the location which mostly ends up being the lower of the body as used for floor coordinates. Figure 4.6 showcases a double split for the person into upper and lower body however only the lower body is used being nearer to the location in the previous frame.

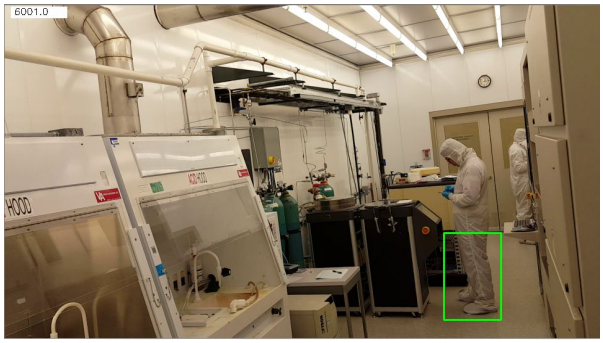
Binary Mask Image	Split in Contour Detection for a person
	

Figure 4.6: Contour Detection with Split

4.4.3 2D to 3D

The distances in 2D image is not sufficient to measure the proximity of a person to the equipment as the distance of a person to two equipment can almost be the same in 2D while being significantly different in 3D. Thus, the 2D image coordinates of the person and the equipment will have to be converted to 3D.

In order to convert the 2D points on the image to 3D, the traditional pin-hole model will be used. The image view is formed by projecting 3D points into the image plane using a perspective transformation which is given by:

$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{4.1}$$

where:

λ is a scalar constant

x, y are the coordinates of the projection point in image

f_x, f_y are the focal lengths expressed in pixel units

c, c_y is the principal point that is usually at the image center

[R | t] is the rotation and the translation matrix

X, Y, Z are the coordinates of the 3D point in the world coordinate space

Equation (4.1) can also be written as:

$$\lambda x = K [R \mid t] X \tag{4.2}$$

where:

K is the matrix of intrinsic parameters of the camera

[R | t] is the matrix of extrinsic parameters

x is the (x,y) coordinate of the projection point in image

X is the (X, Y, Z) coordinate of the 3D point in the world coordinate space

The last column of the rotation matrix can be ignored as it is simply the cross product of the first two columns of the pose. Thus the rotation of the extrinsic matrix becomes R' with size 3 x 2 with [R' | t] becoming a 3 x 3 matrix. Reverse projecting the points from 2D image plane to 3D can now be done through the inverse of the intrinsic and the extrinsic matrix. This will result into X' with dimension 3 x 1 and (4.2) becomes:

$$\lambda K^{-1}[R' \mid t]^{-1} x = X' \tag{4.3}$$

The homography for reverse projecting the 2D image points to 3D world coordinates can be set to:

$$H = K^{-1} [R' \mid t]^{-1} \tag{4.4}$$

Substituting (4.4) into (4.3) and ignoring the scalar constant λ:

$$X = H x \tag{4.5}$$

This equation can be used to obtain 3D coordinates X from 2D image points x. While this does not maintain the same units as the original world coordinates, the relative distances between the coordinates stay the same. The intrinsic parameters K of the camera can be estimated through the checkerboard pattern or the vanishing lines in the image [19]. The intrinsic parameters represent the inherent properties of the camera and do not depend on the scene viewed. The joint rotation-translation matrix [R | t] is called the matrix of extrinsic parameters. It is used to describe the camera position with respect to the scene and differs for each image. As the camera is static while capturing the video segments for a room, the extrinsic parameters stay the same. The extrinsic parameters can be computed through perspective-n-point. Perspective-n-Point [20] estimates the pose of a calibrated camera given a set of at least 3 or more 3D points and their corresponding 2D projections in the image. Once the intrinsic and the extrinsic parameters of the camera are estimated, it can be used as an homography as described in equation (4.4) for converting 2D image points to 3D.

4.4.4 Tracking & Sensing

2D to 3D estimation can be done through a homography consisting of the inverse of camera intrinsic and extrinsic parameters as described above in section 4.4.3. It is not possible to get real-time depth information of a person from a single monocular camera. The only plane that stays the same between the background cleanroom and the person inside the frame is the floor. In order to measure the proximity of the person to the equipment, floor coordinates of the person and the equipment are used. The average of the bottom two coordinates of the rectangle corresponding to the blobs are used to evaluate the distance. These floor coordinates are used to assign the detected blobs in the current frame to a person from the previous frame or treat it as a new person. The euclidean distance between the previous coordinates of the person and the floor coordinates of the blobs are compared and the closest ones within some limit are assigned respectively. This helps keep track of the person as he is moving throughout the frame. The floor distance between the person and the equipment within the threshold provides the proximity of a person to an equipment. The major caveat is occlusion when two or more people walk past each other. The tracking will not be able to isolate when two people walk past each other as it will

appear to be a single person to the camera at that point. This is a limitation right now which is discussed as part of the future work.

Algorithm 4.3 Tracking & Sensing

```
1:  global variables
2:      cv ← python cv2
3:      constant τDist                    : Threshold distance
4:      constant entryCoordinate          : Entry coordinate
5:  end global variables
6:
7:  function tracking(persons, blobs) :
8:      for each p ∈ persons do
9:          for each b ∈ blobs do
10:             if dist(p, b) < τDist then:
11:                 p.updateCoordinates(b)
12:                 blobs.remove(b)
13:                 break
14:
15:     for each b ∈ blobs do
16:         if dist(b, entryCoordinate) < τDist then:
17:             persons.add(new Person(b))
18: end function
19:
20: function proximitySensing(persons, equips) :
21:     for each p ∈ persons do
22:         for each e ∈ equips do
23:             if dist(p, e) < τDist then:
24:                 e.workedBy.append(p)
25:                 break
26: end function
```

The *dist* function in the above algorithm converts the 2D coordinates to 3D using the homography and then compares the euclidean distance between the two positions.

4.4.5 Algorithm

The complete logic for proximity sensing using all of the above defined functions is demonstrated in the following algorithm:

Algorithm 4.4 Proximity Sensing Using 2D Camera

```
1:  bg ← getBackgroundImage(cleanroom.id)
2:  bg ← resizeGraySmoothFrame(bg)
3:  equip ← getEquipments(cleanroom.id)
4:  frames ← video.readFrames()
5:  local variable: persons
6:
7:  for each f ∈ frames do
8:      f ← resizeGraySmoothFrame(f)
9:      binaryMask ← backgroundForegroundSubtraction(bg, f)
10:     cnts ← getContours(binaryMask)
11:     rectCnts ← boundingRects(cnts)
12:     rectsFloor ← getFloorCoordinates(rectCnts)
13:     persons ← tracking(persons, rectsFloor)
14:     resetEquipmentAvailability(equip)
15:     proximitySensing(persons, equip)
```

4.5 Experiment

4.5.1 Setup

We had captured video segments from two cleanrooms in Holonyak Micro and Nano Technology Laboratory at University of Illinois, Urbana-Champaign. The two video segments, one for each room were roughly about ~ 12 minutes each at 30 frames per second. These videos were captured using a Samsung S6 mobile phone camera placed on a tripod stand near the entrance. The videos were copied to the laptop and the vision algorithm written in python was then executed on these video segments.

Due to the vibration in the camera caused by one of the machines, the first video segment had too much shaking thus rendering it unusable. For the second video segment, there was a point when the camera position was moved. The background frame could not work on this and so some of the later portion of this segment had to be disregarded. Overall, about ~ 7 minutes of video segment from the second video was used for evaluation.

Figure 4.7: Video Snapshots of two cleanrooms

4.5.2 Ground Truth

The usable video segment was manually labeled in the interval of frames on whether someone is working on an equipment or not. This is defined as when someone is standing close to an equipment and may not even be necessarily working on it. A labeling sample is presented below in table 4.1 when two people are present inside the cleanroom. For example, from frame number number 3000 to 3480 from table 4.1, one person was working on a computer and another at the Black Rotator. From frame number 3526 to 3550, both the persons were not working on any equipment and were simply moving. Due to the problem of occlusion, the tracking aspect has not been evaluated and is left as part of future work.

Frame Interval	Equipments
3000 - 3480	Computer - Black Rotator
3481 - 3525	Computer - None
3526 - 3550	None - None

Table 4.1: Video Labeling Sample

4.5.3 Results

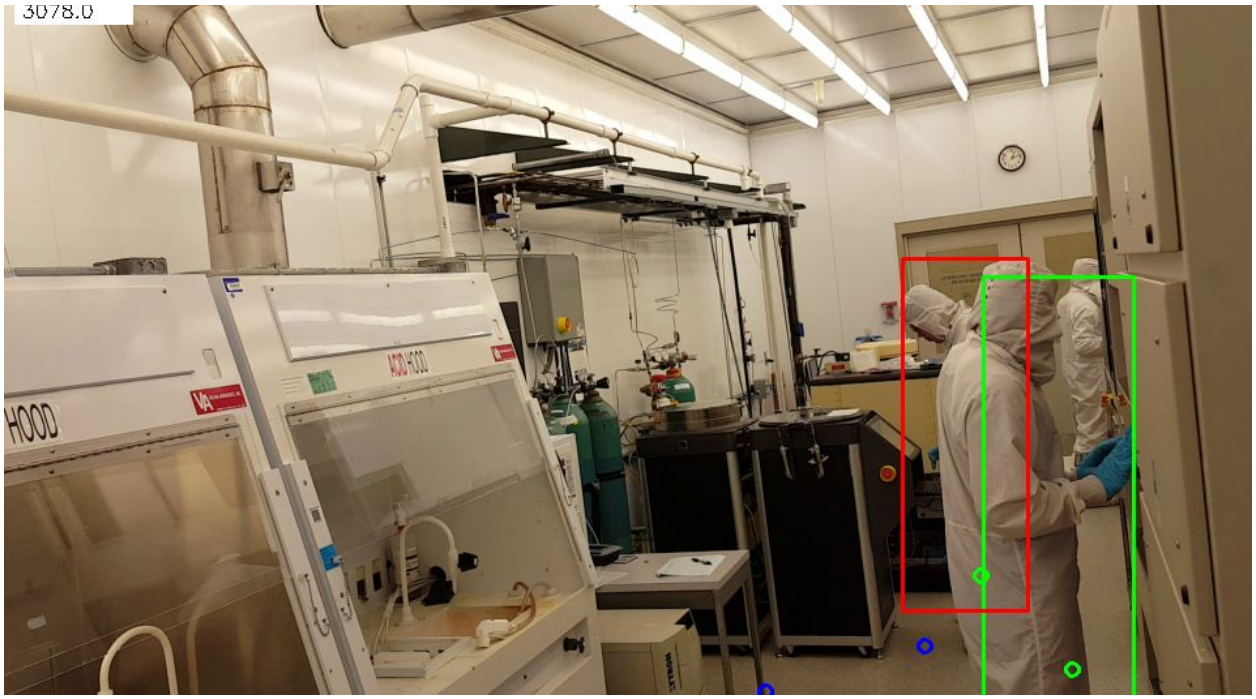


Figure 4.8: Two persons working on equipment. Purple indicates that the equipment is available and green means it is occupied.

The figure 4.8 shows a sample result of the proximity sensing algorithm visually which gets outputted to a file. The results are compared with the ground truth as labeled in section 4.5.2. The accuracy is measured as the number of correct proximity labels that the algorithm outputs summed over all of the frames divided by the total number of labels. For instance if the algorithm outputs *Computer - None* for two people while the correct label is *Computer - Switchboard*, the accuracy will be 0.50. This means that the algorithm classified one person as working on the computer and the other as not working on any equipment while the correct classification for the other person instead of *None* should have been *Switchboard*. Even if the algorithm simply detected a single person in this case and has correctly identified the equipment he is working on, the accuracy will still be 0.50. With this in mind, the overall accuracy comes to about 0.71 for the 7 minute video segment as captured in section 4.5.1. The accuracy is significantly impacted by false positives which is classifying the person as working on an equipment while the person is simply walking past it.

4.6 Future Work

In the actual scenarios, it is difficult to figure out whether a person is actually working on an equipment or simply standing next to it. This problem is known as human object interaction. While this has been out of scope from our work, there are many techniques that can be experimented with for checking if a person is actually working on an equipment or not.

The identification of a person cannot be determined with computer vision. RFID with presence detection as described in chapter 2 can be integrated with this vision system to solve this problem.

The implementation for proximity sensing does not take into account whether the person is standing or moving. Even when the person is moving, the proximity sensing gets triggered which results into a lot of false positives. The accuracy can be improved by only considering stationary positions.

While tracking is implemented, it does not work in the case of occlusion. This can be solved with the help of multiple cameras or through a fusion approach between computer vision and BLE used for movement tracking described in chapter 3.

Due to limited availability of videos, the implementation wasn't tested in different cleanrooms. Moreover external factors such as change in lighting, vibrations, camera movement impacts vision processing. The solution needs to be more resilient to these factors.

CHAPTER 5: CONCLUSION

5.1 Summary

The position of a person in a given space is an important element of contextual information. Indoor localization is the process of obtaining the device or users location in an indoor setting. The environmental context is an important consideration for localization as it can pose unique challenges. While there has been a lot of literature in applying localization in traditional environments, this thesis explores various avenues for localization in scientific labs called cleanrooms. Some of the applications of localization in cleanrooms involve detecting the entry / exit of the person to a cleanroom, tracking the movement across multiple cleanrooms and evaluating whether a person is working on an equipment or not.

The cleanrooms have a distinct environment with people wearing white suits, huge equipment made from different materials, congestion in the wireless channels etc. With a wide variety of sensors and technologies available for localization, RFID, BLE and Computer Vision are selected for its applicability in cleanrooms.

The comparison of these sensor devices is shown in table 5.1 which helps evaluate the usage of each of the sensor systems. The hardware column includes the essential hardware required to deploy the system. The installation effort column states the efforts involved in deploying the system to the physical environment. The calibration effort is specific to the algorithms and the implementations covered in this thesis. The maintenance while dependent on the specific model / built is derived from the industry standards.

It can be observed from table 5.1 that BLE is cost-effective and easy to deploy. One of the advantages of using BLE over other wireless technologies such as Wifi is the ease with which it can be deployed. As cleanrooms have huge metal objects that can attenuate electromagnetic signals, beacons can easily be placed in strategic locations to mitigate this problem. While RFID seems to have a high cost, most of it is only one-time associated with the antenna and the reader. The passive tags are extremely cheap costing only a few cents each. The advantage of the camera is that most of the cleanrooms may already have one for security

reasons. If this is already positioned properly for the vision algorithm, there is no need to worry about installation. While vision processing algorithms require extensive calibration, if the make / model of the camera is the same across multiple cleanrooms, the intrinsic calibration only needs to be done for one as it stays the same for cameras with the same built. From a holistic point of view, it can be observed that there isn't a perfect sensor that fairs well in all of the areas but has its own advantages and limitations.

	Hardware	**Cost**	**Installation Effort**	**Calibration Effort**	**Maintenance Level**
RFID	UHF RFID reader with 2 antennas and 8-12 passive tags	$1000	High 1) Antennas are big 2) Space constraints near the exit 3) Wiring for power	Low 1) Transmission power 2) Hyper parameters for classification	Low 1) Equipment sustainable over 5 year. 2) Washable tags ~ few months based on usage
BLE	4 BLE Beacons (Assuming user's mobile device is used as a reader)	$75	Low 1) Small Size 2) No wiring	Medium 1) Path loss index 2) Coordinate system	Low Replace once battery is dead ~ 5 years
Computer Vision	1 standard off-the-shelf camera	$100	Medium 1) Needs to go on ceiling 2) Wiring for power	High 1) Camera intrinsic parameters 2) Extrinsic parameters for each room	Low High lifespan of a camera. Generally > 5 years

Table 5.1: Comparison of RFID, BLE and Computer Vision

The notion behind localization can be further broken down into presence detection, movement tracking and proximity sensing. Presence detection merely involves detecting the existence of an object in an area. In the case of cleanrooms, this can be automatically tracking the entry / exit of the person. Both RFID and BLE can be used to solve this problem. RFID will be better suited due to its economic feasibility, range control and the portability of tag. Moreover, the accuracy of RFID for presence detection in our tests resulted in an accuracy of 82.5% while it was only 62% for BLE. BLE however can be effectively used for continuously tracking the movement of a person across multiple rooms with an average accuracy of 2.1m during our experiments. Fine grained tracking can be done through computer vision where traditional sensors lack this capability. While the exact coordinates may be difficult to get, the proximity can be sensed. The accuracy for evaluating whether a person is working on an equipment through vision is 71% in our experiment. The overall results of our experiments with each of the technologies is shown in table 5.2.

	Presence Detection	**Movement Tracking**	**Proximity Sensing**
RFID (Passive)	Accuracy ~ 82.5%	Technical Limitation due to occlusion	Physical Deployment constraints
BLE	Accuracy ~ 62%	Accuracy ~ 2.1m	Technical limitation due to low accuracy
Computer Vision	Identification not possible due to white suits	Possible with multiple cameras	Accuracy ~ 71%

Table 5.2: Localization functions with BLE, RFID & Vision

5.2 Discussions and Future Work

We observe from our results that it is not possible for a single sensor to perform all of the three functions of localization effectively. There are technical limitations that if overcome can expand the scope such as BLE for proximity sensing or RFID for movement tracking. For example, the accuracy of BLE can be improved with the Bluetooth 5.1 version which utilizes

angle of arrival. However, even if technical limitations are overcome, there are practical constraints as well. For instance, computer vision cannot ascertain the identity of the person as everyone is wearing the same white suit. It is not feasible to have an RFID antenna near each equipment for proximity sensing. The congestion in the wireless network will impact the BLE signal even if its accuracy is improved. Therefore, a hybrid sensor fusion approach will have to be used if localization needs to be solved comprehensively.

Localization is going to play an important role in our future as the number of IoT (Internet of Things) devices grow forming a network that can sense and respond. The wide scale applications of localization will involve heterogeneous technologies. Hybrid approaches will leverage the complementarity of several technologies with benefits and limitations of each as explored in this thesis. There are yet major challenges that must be overcome to realize the full potential of localization such as energy efficiency, scalability, security, availability etc. This is a step towards building that future.

REFERENCES

[1] “Radio-Frequency Identification.” *Wikipedia*, Wikimedia Foundation, 15 Apr. 2020, en.wikipedia.org/wiki/Radio-frequency_identification.

[2] Bottani, Eleonora, et al. "Performances of RFID, acousto-magnetic and radio frequency technologies for electronic article surveillance in the apparel industry in Europe: A quantitative study." *International Journal of RF Technologies* 3.2 (2012): 137-158.

[3] Hauser, Matthias, et al. "Pushing the limits of RFID: empowering RFID-based electronic article surveillance with data analytics techniques." (2015).

[4] Goller, Michael, Christoph Feichtenhofer, and Axel Pinz. "Fusing RFID and computer vision for probabilistic tag localization." *2014 IEEE International Conference on RFID (IEEE RFID)*. IEEE, 2014.

[5] Kellomäki, Tiiti, et al. "Towards washable wearable antennas: a comparison of coating materials for screen-printed textile-based UHF RFID tags." *International Journal of Antennas and Propagation* 2012 (2012).

[6] “Scipy.signal.find_peaks.” *Scipy.signal.find_peaks - SciPy v1.4.1 Reference Guide*, docs.scipy.org/doc/scipy/reference/generated/scipy.signal.find_peaks.html.

[7] “Bluetooth Low Energy.” *Wikipedia*, Wikimedia Foundation, 13 Apr. 2020, en.wikipedia.org/wiki/Bluetooth_Low_Energy.

[8] De Blas, Aitor, and Diego López-de-Ipiña. "Improving trilateration for indoors localization using BLE beacons." 2017 2nd International Multidisciplinary Conference on Computer and Energy Science (SpliTech). IEEE, 2017.

[9] Thaljaoui, Adel, et al. "BLE localization using RSSI measurements and iRingLA." *2015 IEEE International Conference on Industrial Technology (ICIT)*. IEEE, 2015.

[10] Röbesaat, Jenny, et al. "An improved BLE indoor localization with Kalman-based fusion: An experimental study." *Sensors* 17.5 (2017): 951.

[11] Kaminsky, Alan. *Trilateration*. www.cs.rit.edu/~ark/docs/trilateration.pdf.

[12] *LevenbergMarquardtOptimizer (Apache Commons Math 4.0-SNAPSHOT API)*, commons.apache.org/proper/commons-math/apidocs/org/apache/commons/math4/fitting/leastsquares/LevenbergMarquardtOptimizer.html.

[13] “Bluetooth Low Energy Scan Settings.” *Android API Guide*, developer.android.com/reference/android/bluetooth/le/ScanSettings.

[14] “Bluetooth Low Energy Advertise Settings.” *Android API Guide*, developer.android.com/reference/android/bluetooth/le/AdvertiseSettings.

[15] Duan, Chunhui, et al. "Fusing RFID and computer vision for fine-grained object tracking." *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 2017.

[16] Wang, Ching-Sheng, and Li-Chieh Cheng. "RFID & vision based indoor positioning and identification system." *2011 IEEE 3rd International Conference on Communication Software and Networks*. IEEE, 2011.

[17] Llorca, David Fernández, et al. "Recognizing individuals in groups in outdoor environments combining stereo vision, RFID and BLE." *Cluster Computing* 20.1 (2017): 769-779.

[18] “Contour Features.” *OpenCV*, opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_contours/py_contour_features/py_contour_features.html.

[19] “Camera Calibration.” *OpenCV*, opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_calib3d/py_calibration/py_calibration.html.

[20] “Camera Calibration and 3D Reconstruction.” *Camera Calibration and 3D Reconstruction - OpenCV 2.4.13.7 Documentation*, docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html.