Contents

Abstract

This master thesis will be centred in two main parts, the first one is the investigation of the metrics and techniques that can be used in the development of an engagement detection system for live events, this investigation includes both the literature research and a basic implementation of the techniques for the measuring of the metrics analysed, this implementation will be done with videos and not with the live data. The second part will be the design, implementation and validation of a system to measure the engagement from at least one metric of the ones analysed before, this part will make use of different different techniques to have a hybrid approach that can improve the measurements.

Because of the broad of the topic the implementation will be done in only one concert hall and it will be centred in developing an approximate measurement of the engagement, including possible extrapolation of the data from a smaller number of people than the total.

List of Figures

List of Tables

List of Acronyms

SVM:	Support Vector Machine.
HOG:	Histogram of Oriented Gradients.
SIFT:	Scale Invariant feature transform.
CPU:	Central Processing Unit.
GPU:	Graphical Processing Unit.
TPU:	Tensor Processing Unit.
API:	Application Programming Interface.
CNN:	Convolutional Neural Networks.
YOLO:	You Only Look Once.
SSD:	Single Shot MultiBox Detector.
R-CNN:	Region-CNN.
mAP:	mean Average Precision.
COCO:	Common Objects in COntext.
URL:	Uniform Resource Locator.
CUDA:	Compute Unified Device Architecture.
FDDB:	Face Detection Data set and Benchmark.
RSSI:	Received Signal Strength Indication.
MUSIC:	Multiple signal classification algorithm.
BLE:	Bluetooth Low Energy.
RFID:	Radio Frequency IDentification.
DBN:	Deep belief network.
DAE:	Deep auto-encoder.
RNN:	Recurrent neural network.
BPTT:	Backpropagation through time.
LSTM:	Long short-term memory.
ReLu:	Rectified Linear Unit.
ECG:	ElectroCardioGram.
RSSI:	Received Signal Strength Indication.
CLAHE:	Contrast Limited Adaptive Histogram Equalisation.
LOPD:	Ley Orgánica de Proteción de Datos.

Acknowledgements

Chapter 1

Introduction

The entertainment industry has experienced a huge development in the recent years, this development has been in it's majority due to the implementation of entertainment services in internet as well as the migration of services to the web. This transition has brought not only more users to the platforms that make use of new technologies in this industry, it also has brought more data of the people using that service. The data collected can be used in many different ways, but one of the most interesting uses of that techniques is to use that data to improve the service, in the case of the entertainment industry the data analysed is the reaction of the people to the service in different moments.

This industry can be divided into two very differentiated type of events:

- Traditional: This type of events has been developed for centuries and includes shows as concerts or theatre. This type of events has been characterised for being live and without any recording, although in recent years they has been recording in order to allow more people to see them. In this type of shows the technology has been slowly introduced in different steps, first to help to improve the experience, with the introduction of devices such as microphones or lights, after that type of devices the technology introduced was centred along bringing more people to the show, in this case cameras or big screens were introduced. Nowadays in order to improve the shows the industry wants to know the parts that the people like and not in order to modify the show to make it more appealing and interactive.
- Modern: This type of shows are centred on being able to be consumed in many places at the same time and at any time that the consumer wants, some of the events that are included are the Video On Demand (VOD) or the video-games. The main difference is that this type of entertainment was design with the technology in mind.

The traditional events implementation of technology has been related to the manager of the show idea, without knowing exactly the likes of the audience, this tendency has changed as nowadays with the implementation of technologies to detect the feeling of the audience.

In the traditional shows the implementation of the online audience has been very slow, and in some events it is very strange to have both audiences, one online and another one in the place. The introduction of online interaction brought many information such as time of maximum connections and disconnections to the show or the characteristics of the people interested in the event, information which was not obtained when the event was only in-person and can be very important for the manager of the event, which was one of the reasons because the streaming of live events has increase. With that information the manager of the events change the future events and with that use of the information a question appeared **Why not obtaining the same information from the in-person audience?**.

Obtaining that data is very interesting but it is also a very broad and ambitious project, which is normally implemented in different steps, this project will be centred around the technologies to measure the in-person audience in an event and their engagement with the event. The measure of the in-person audience and engagement can be very dependable on the event, including it's conditions such as place or time, reason to maintain a common event for testing, this testing field will be the weekend concerts in the Dabadaba concert place.

This master thesis will analyse the possibilities of that technology with the identification of the main characteristics and metrics that can be measured in the in-person audience of a live event and their relation with the engagement of the audience, some of the analysed metrics will be the number of people that are in the event, the position of each person in the room, the movements of the people, the attention of the people and the emotion that can be obtained. This project will also identify the technologies that can be develop in order to detect at least one of the characteristics analysed. This last part will consist in two part investigation with literature as papers and possible algorithms and the implementation with libraries.

Chapter 2

Objectives and methodology

In this chapter the objectives of the project, divided in smaller objectives to been able to complete them easily, and the methodology that will be followed.

2.1 Objectives

The main objective of this project can be sum up as the design, implementation and validation of a system capable of measuring the audience engagement in a live event, in order to fulfil that broad objective it has been divided in smaller objectives:

1. **Metric definition**: This objective is necessary as it analyses the different characteristics of both the people and live events in order to provide the best characteristics that can be used to know if the people likes the event or not. The analysis will be done choosing the characteristics that can be objectively analysed and select the techniques that can give information about that characteristics. To prove this objective has been fulfilled a list of techniques will be chosen by the metrics information given to be compared in performance.
2. **Existing technology identification and experimentation**: This objective because of the broad that it can be and the different technologies involved can be divided in pseudo-objectives:
   (a) **Usage of artificial vision to detect and track people**: This pseudo-objective make use of artificial vision techniques for the detection and tracking of people, the detection will include the pixels in the image or in the frame of the video where the person is, while the tracking, which will only be done in the videos, will store the movements done by the person detected. In order to prove that this pseudo-objective has been fulfilled a bounding box will be drawn over covering all the person area, as well as the path that the centre of the bonding box has followed in the case of a video.

(b) **Usage of artificial vision to detect the face of the people**: This pseudo-objective make use of artificial vision for detecting where the face of a person is, with special attention to being able to detect the face in strange angles and with different conditions. In order to prove that this pseudo-objective has been fulfilled a bounding box will be drawn covering the whole face area.

(c) **Usage of artificial vision to detect and track body parts on a human body**: The most important parts of the body for measuring the audience engagement in live events are head, chest and arms, hips and legs, the last two with lower detection rate as being part of the lower body. All that parts will be detected using artificial vision in both images and frames of videos, and in the case of the video their movement will be stored. In order to prove that this pseudo-objective has been fulfilled a bounding box will be drawn covering the whole body part area for each of the body parts, as well as the path that the centre of the bonding box has followed in the case of a video.

(d) **Detect the position of different parts of the face**: One of the part of the body where more expressions can be detected is the face, because of this reason, being able to detect the different parts of the face can bring many information,the most important parts are eyes, nose or mouth. In order to prove that this pseudo-objective has been fulfilled a point will be drawn over the frame of the video covering the different parts mentioned before.

(e) **Detect the direction of attention**: As the attention of the person is given normally to the most appealing thing on the room for that person detecting where the person is given the attention can bring much information about the engagement. In order to prove that this pseudo-objective has been fulfilled a variable will store the direction of the attention on each frame as well as with the combination of the position of the person detecting the zone of the room where the person is looking, such as bar, toilets or scenario.

(f) **Improve the artificial vision position results with other techniques**: This pseudo-objective is centred in improving the artificial vision person detection capabilities by covering it's downsides using other techniques for person detection in a room. In order to prove that this pseudo-objective has been fulfilled the number of people detected will be improved as the detection of the place in the room.

(g) **Count the number of people in the event**: This pseudo-objective is centred on being able to know in every moment with good reliability the approximate number of people in the room and differentiate between different zones of the room. In order to prove that this pseudo-objective has been fulfilled a the number of people detected will be stored in a variable and it will be shown in the terminal, as well as the number of people in each zone.

3. **Design and development of a solution using one metric**: The final objective of this master thesis is finishing it with a valid solution to measure the audience engagement, for this the necessary hardware has to be built and fully connected, with the possibility of being installed in the place where the testing is done. The software will be needed to be fully written, installed and working. This objective as being the last one can be taken as fully completed when the system can recognise the chosen metric with the technique or the group of techniques chosen.

2.2 Methodology

The methodology applied to this project will be to divide it in three parts:

1. Investigation: In this part of the project the methodology will be to have different possible metrics to measure the audience engagement, as well as different techniques to obtain that metrics. In this part the main focus will be to read information of ways that the audience engagement is done. In this part the main metrics to measure the engagement will be chosen, which will determine the next parts of the project. The different metrics to be analysed are:
    - Number of people: This metric make use of the detection to know the number of people that are in the room, it can be improved to detect the number of people in different parts of the room.
    - Position of people: This metric make use of the detection to know the approximate position of the people, this can be taken as an improvement of the previous one, but it was treated as another one because the improvement of the information given.
    - Movement of people: This metric make use of the position of the people to determine the movement of the person by using the position thought the time. This is a topic which is very related with the new LOPD(Ley Orgánica de Proteción de Datos) of Spain, which needs to be taken into account.
    - Attention direction: This metric make use of the position of the different parts of the face to determine the direction where the person is looking.
    - Emotion of people: This metric make use of the detection different signals in the body to determine the emotions of the people.
2. Technology tweaking: In this part the different technologies that were investigated in the previous part will be changed to improve the performance in the case of use chosen. In this part the most interesting techniques to obtain the chosen metrics will be used and changed to improve the out of the box performance, the techniques will be changed too to give more metrics and to cover more objectives. The two technology analysed are:
    - Computer vision: This technology uses the computer vision techniques make use of machine learning for the detection, in this case the technology will be adapted to the proposed case improving the performance and adding different functions. After the tweaking different test will be done in order to test the performance and the new functions. In this case the test will be done for two videos, one with good conditions and another with real conditions.
    - Radio waves: This technology uses the radio waves to detect the devices that all the people has, it needs that this type of communication is activated. As in the other technique some tweaking is done to the technique to improve the performance and the addition of new functions. In this case the tests are done to see if performance and new functions.

3. Creation of hybrid approach: This is the last part of the project, in this part several tweaked techniques will be used at the same time, this will not only use all the data to have the metrics chosen, the data of each of the techniques will be use to change some of the parameters of the other in order to have a better performance. The hybrid approach will make use of the two techniques at the same time using the results of the measurements taken from the results. This results are used to tweak the data introduced in the techniques in order to improve the results of the techniques.

Chapter 3

State of the art

This master thesis centred on measuring the audience and their reaction to a live event can implement different techniques, this techniques are very broad, the techniques can be divided into different categories depending on what is detected.

This chapter is organised in section with the different techniques that can be used to perform the objectives of the project, inside each of the sections the different ways of using this techniques are explained including the metrics of this part. In the case of the object detection as with the image method different people can be detected ignoring the rest of the object being it explained. The last section makes a comparison of the techniques analysing the accuracy, hardware, computational cost, user interaction and metrics obtained, this were sum up in different tables.

3.1 Object detection:

Object detection is the technique to obtain the position of an object, all techniques make use of different technologies, being the computer vision based techniques the most used ones. Each object has differentiating features, which are the ones used for the detection. The different technologies used for object detection are:

3.1.1 Image

The majority of the algorithms for object detection make use of this technology, using algorithms of artificial intelligence. This techniques consist on giving the computer the minimum knowledge for understanding the images, or the frames in the case of using videos, to automate the detection of objects. The system learn the necessary information to make the detection by going through a series of examples. [1]

The main work before 2012 was centred on developing new models and databases to reduce both the computation power and training time, the techniques used for object detection before 2012 were:

- Haar cascade: This is a classifier that has to be trained with images to work properly. It works by overlapping the image with some negatives. This model is highly dependable from the training data-set as it is used directly in the comparison. [2] The first detector to have good performance is the Viola-Jones detector, which is based on the haar cascade classifier, this detector although being created to solve the problem of face detection it can be applied to several classes of objects. The detector perform four operations:

  1. Haar feature selection.
  2. Integral image creation.
  3. Training.
  4. Cascading filters.

  It uses the principles of the haar cascading to detect the main features of objects as shape or colour. In the training the Adaboost algorithm is used to speed up the detection of the features, this algorithm takes the output of the slower haar one and used weights to boost the classification. The weights are given to each sample in each iteration, this weights represent the error that is currently in that part. [3]

- SVM: In this type of model the detection is perform by separating different features as colours, edges or textures for the detection. The computation of the features can be very high so a kernel function is used, which removes the necessity of mapping by computing the inner products between the pixels. This kernel function need to be done for each case as it has not been found an universal one. In order to find the object features the image is prepossessed with multiple sliding windows to compare the image and the possible features. [4]

- Histogram of oriented gradients(HOG): It is a descriptor of the main characteristics of the object to detect in the deep learning model.The majority of the features in an object can be described using the gradient intensity and the direction of the edges, this method divides the image in smaller sections, which are the parts where the gradients are calculated. [5] The method has 5 steps:

  1. Gradient computation: This step allow the elimination of a part of the prepossessing and normalisation. In this step a derivative mask for a 1D point is applied for both direction of the 2D space.
  2. Orientation binning: In this part the image is segmented into smaller sub-images called cells. After the segmentation the detection of the orientation is performed, this detection is based on the orientation of each pixel on the cell and then the orientations are weighted to have an unique one for the cell.
  3. Descriptor blocks: In this step the cells are grouped into blocks, making a descriptor by concatenating the data of the cells. The blocks can be either rectangular or circular.

4. Normalisation: In this step all the data from each block is changed to have the same reference including orientation, contrast and colours. All of the data is in the HOG descriptor.
5. Recognition: In this step the HOG descriptor is used, as it contains the features in the image, to be introduced in a machine learning algorithm as SVM or Haar.

- Scale Invariant feature transform(SIFT): This is used to detect and describe local features in images. It works by extracting the keypoints from reference images, the objects are detected by comparing each feature to the database, and it is taken as a candidate when the euclidean distance from one feature keypoints to the other is lower than a minimum.

One library very useful to work in machine learning is Tensorflow, developed by Google. [6] It's flexible architecture allows for an easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices. It has stable APIs for both python and c. Tensorflow also has different tools, some for new users and other for more experienced ones. It has the power to bring machine learning to mobile devices with tensorflow lite.[7]

Normally on top of tensorflow Keras is used, Keras is an open source neural network library that can be used with python. It was build to enable a fast way for experimenting with neural networks. It focuses on being user-friendly, modular, and extensible. Keras was conceived to be an interface rather than a standalone machine-learning framework. It offers a higher-level, more intuitive set of abstraction that make it easy to develop deep learning models regardless of the computational back-end used. It contains numerous implementations of commonly used neural-network building blocks such as layers, objectives, activation functions, optimisers, and a host of tools to make working with image and text data easier. [8]

After 2012 neural networks started to been used, The neural networks are computational models that tries to mimic the structure of the human brain with layers of neurons and the connections between them. Nowadays the type of neural network that is most used for artificial vision are the Convolutional Neural Networks (CNN), although there are other methods based on deep learning as You Only Look Once (YOLO) [9] or Single Shot MultiBox Detector (SSD) [10]. The CNN type of networks are usually used for image and video as they take advantage of 2D structures with the structure of the layers. They are easier to train as it needs less parameters. An improvement from this type of neural networks are the Capsule Neural Networks that add capsules to the convolutional network to reuse the output from several capsules to make a more stable representation. This type of networks are divided on different types as Region-CNN(R-CNN), Fast R-CNN and Faster R-CNN.

The neural network techniques are dependable on the knowledge of the computer, as the computer has to know what are the main key features of the object that it is looking to be able to look for them. The steps that the computer has to perform are:

- Training: this part is divided into the leaning and validation, in the first part a model of the object is given to the neural network, this allow the neural network to know what to find. In the verification part some real images are given to the network to validate the results obtained.

- Detection: This is the part where the actual data is given to the network, in this part the detection is done by looking for the same features that was learnt in the first part.

There are several types of learning techniques for neural networks, supervised, unsupervised and reinforcement, in the object detection field the most used one is the supervised learning, which uses a set of data as examples, normally giving the input and output that the network should give. This way of learning works by providing feedback for every solution of the network about the quality of the outputs.

The main problems with neural networks are the time needed to be trained to perform the detection. The neural networks normally need huge computation power, reason to be normally used with GPUs instead of normal CPU, which reduce computation time.[11] Other methods used with neural networks are:

- You only look once (YOLO): is a state-of-the-art, real-time object detection system, the main difference with CNN is the usage of the model with a sliding window to the image or frame, considering detection the ones with better probability as other models, it uses only one neural network for the whole image, the neural network is the one that divide the image and predicts the objects with bounding boxes and probabilities for each segment, at the end some weights are used to smooth the result and eliminate recursive bounding boxes and false positives.The current version improve training and increase performance, including: multi-scale predictions and a better backbone classifier.[9]
- Single Shot MultiBox Detector (SSD): This method scored over 74% mAP (mean Average Precision) at 59 frames per second on standard data-sets. By it's name it can be seen that it is a network to detect objects (detector), although it also classifies them, the localisation and classification of the objects are done with one forward pass of the network (Single Shot), and that it uses a bounding box regression technique called multibox[12] (Multibox). SSD's architecture is built on VGG-16 architecture. The reason VGG-16 was used as the base network is because of it's performance in high quality image classification tasks. Instead of the original VGG fully connected layers, a set of auxiliary convolutional layers were added, this enable the feature extraction at multiple scales and decrease the size of the input to each subsequent layer.[10]

Some data-sets are:

- COCO[13]: COCO is a large-scale detection, segmentation and captioning data-set. It has several features:
  - Object segmentation
  - Recognition in context
  - Super-pixel stuff segmentation
  - 330K images (more than 200K labelled)
  - 1.5 million object instances

  - 80 object categories
  - 5 captions per image

- ImageNet[14]: ImageNet is a data-set of images that are organised according to the WordNet hierarchy. WordNet contains approximately 100,000 phrases and ImageNet has provided around 1000 images on average to illustrate each phrase.

- OpenImages[15]: Open Images is a data-set of almost 9 million URLs for images. These images have been annotated with image-level labels bounding boxes spanning thousands of classes. The data-set contains a training set of 9,011,219 images, a validation set of 41,260 images and a test set of 125,436 images.

The main libraries and wrappers for object detection are:

- DLib[16]: this is a modern C++ toolkit, which has been also ported to other programming languages as python, this toolkit contains machine learning algorithms and tools for creating complex software to solve real world problems. Dlib's open source licensing allows it's use in any application, free of charge. For object detection it can use either a sliding window classifier or a CNN.

- CVLib[17]: It is a high level easy-to-use open source Computer Vision wrapper for Python, the guiding principles of cvlib are heavily inspired from Keras, simplicity, user friendliness, modularity, and extensibility. It has functions for detecting faces, gender and objects, with different characteristics in each, but in the object detection section it uses Yolov3 model trained with COCO data-set for detecting up to 80 object.

- ImageAI[18]: It is a python library built to allow developers an easier way to make the deep learning and computer vision applications. It main benefits are:

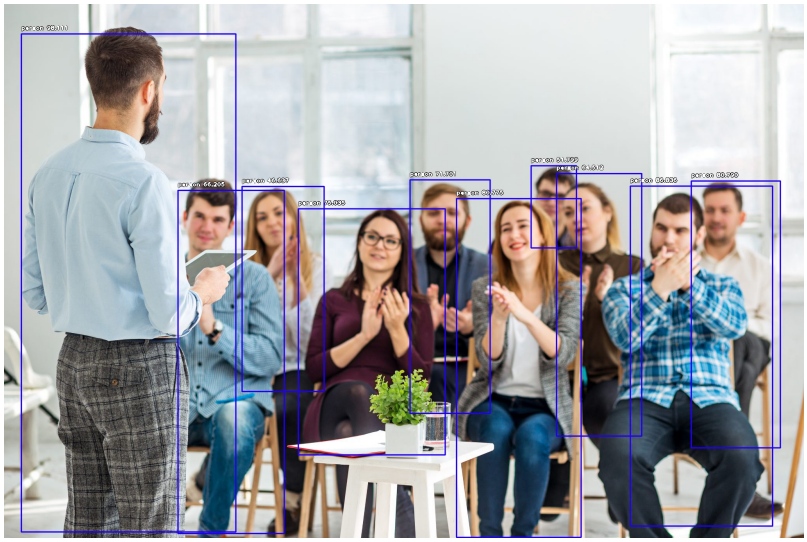
  - Support for 4 machine learning algorithms trained with ImageNet-1000 data-set and 3 trained on COCO data-set.
  - Custom image and video prediction.
  - Basic object tracking.

  For object detection it can use:

  - Retinanet: A medium size model with high performance and accuracy and longer detection time.
  - YoloV3: A huge size model with moderate performance and accuracy and detection time.
  - TinyYoloV3: A low size model with optimisation for speed and moderate performance with fast detection time.

  This library is highly customisable allowing to control both the objects and the minimum probability to be detected, as well as the speed of the detection between five possible, normal fast, faster, fastest, flash, and allowing the use of the image or only the numpy array.

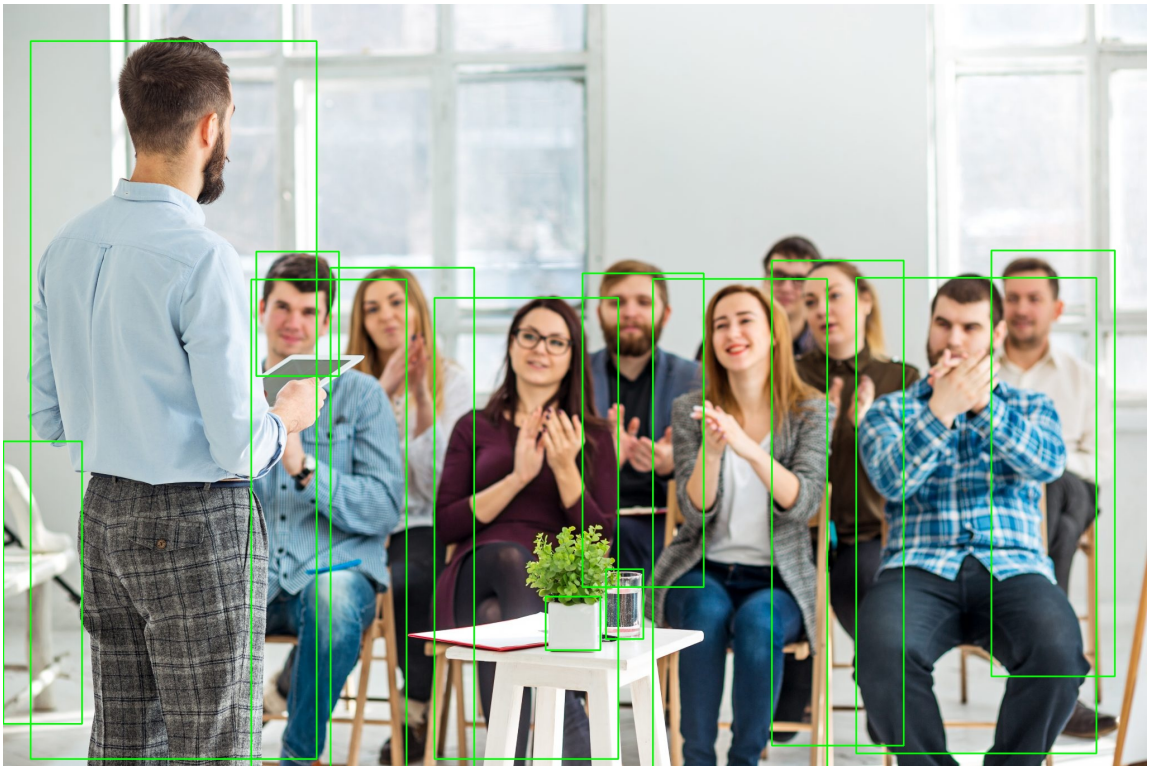
- Darknet[19]: It is an open source neural network framework. It is fast, easy to install, and supports CPU and GPU computation. It uses popular models as ResNet and ResNeXt combined with Recurrent Neural Networks. It has the possibility of using a reduced version to improve speed with the trade-of of accuracy[9].



(a) Image AI

(b) Fast Image AI

Figure 3.1: Object Detection with images



(a) CVlib

(b) Darknet

Figure 3.2: Object Detection with images

Face detection:

A part of the object detection with vision techniques that has been heavily developed is the detection of different parts of the body, one of the most studied one has been the face. In the case of face detection the main source for the detection is the image, although the 3D re-composition from radio-frequency signals can be done. Nearly all the techniques and algorithms for face detection make use of machine learning with computer vision.

The main work done in this field has been around the develop of better training model, which normally consist on having more labelled data or better model of computation. There are several algorithms and libraries for face detection, but they use similar detection models as in object detection. This type of detection is only centred on detecting frontal faces, and normally the training is done with that type of faces, but with some techniques and libraries it allow the detection of faces in different angles.

The main uses for this techniques are recognition of faces and motion capture, being the first one the ability to relate an image of a face with the information of a person, and the second one being the ability to record the movement of a face and reproduce them in a virtual environment.

The main data-sets used for face detection are:

- Labelled face in the wild[20]: This database has the faces labelled with the name, but as the previous step for that is the face detection it can be used ignoring the labels, being able to be used for face detection and recognition. This set contains 13000 images with faces. The main drawback of this database is that the majority of the faces comes from caucasian people. All the faces had been detected with the Viola-Jones and labelled.
- FDDB[21]: This database from the university of Michigan contain annotations of 51171 faces inside 2845 images.
- Wider Face[22]: This database is a face detection benchmark database. It contains 32,203 images and 393,703 labelled faces. This data-set has different scales, poses, occlusions, expressions, colours and makeups and illuminations.

The main libraries and wrappers for face detection are:

- DLib[16]: as it was explained before is a modern C++ toolkit, this toolkit is used in object detection so with the proper training, only with faces, it can be used for the detection of faces. As with object detection for face detection it can use either a sliding window classifier or a CNN.
- CVLib[17]: as with object detection the cvlib wrapper can be used when it has been trained with images of only faces.
- Face_recognition[23]: This library has been developed mainly for face detection, it provides commands for face detection, facial feature manipulation and face recognition. It uses DLib face recognition build over a CNN deep neural network.

(a) face_recognition

(b) CVLib

Figure 3.3: Face Detection with images

3.1.2 Wi-Fi

There has been several ways of using waves for the detection of objects in which the reflection gives the position and shape of the object. Nowadays people are surrounded by waves being the most common the ones in the 2.4GHz frequency. That frequency is one of the Wi-Fi bands[24, 25, 26], so this can be used to perform the detection of objects. The majority of the work done for detecting objects with Wi-Fi has been centred on detecting people because every person carries a Wi-Fi capable device, this allow the person to be easily detected, but this same principle can be applied to the detection of any object, the only pre-requisite is to have a Wi-Fi capable device.[27]

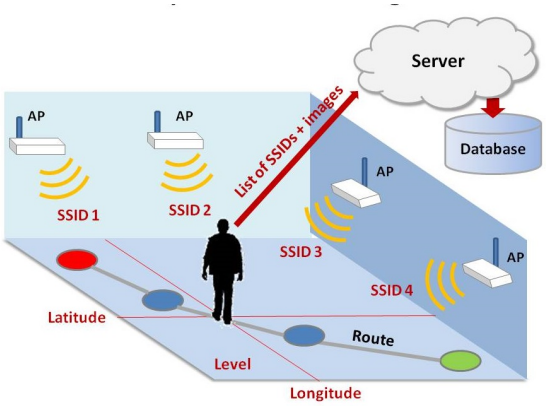


Figure 3.4: Wi-Fi indoor location

The most used techniques make use of Wi-Fi capable devices to emit and receive the Wi-Fi signals, this is an easy method but not the only one as there has been new techniques that does not need the object to be detected to carry a Wi-Fi capable device, this techniques are based on the reflection of signals as the sonar.[27, 28, 29]The main methods for object detection with Wi-Fi are:

- Wi-Fi RSSI (Received Signal Strength Indication): This method is based on measuring the strength of the signal in the receptor, it works better combining the RSSI of different access points to calculate the distance. This method can be improved by using trilateration with several access points. It's main benefit is the simplicity with the trade-of of the accuracy, typically 3 meters.
- Wi-Fi fingerprint: This method make use of a database of the different coordinates related to the signals strengths to simplify the previous method. It's main disadvantage is the necessity of new measures for any change of the room.
- Wi-Fi time of flight: This method uses the time between the emission and the reception of the signal to calculate the distance between them. The main problem with this method is that the measurements needs to be in the range of nanoseconds to have a decent accuracy, lower than 3 meters.

- Wi-Fi angle of arrival: For this method several antennas are used in array to perform an estimation of the angle in which the signal is coming. This method uses the multipath signal and apply triangulation to them. The main part of the algorithm is the computation of the angle that is done with the Multiple signal classification algorithm (MUSIC) to estimate the frequency of the waves and then the radio direction.[30]

3.1.3 Bluetooth

Bluetooth technology is a radio based communication system centred in proximity not in exact position. This technology has been developed to use lower energy and have more precise control on the emitting power and distance, with that the object detection can be performed with a small deviation. The main improvements of this technology has lead to being able to detect objects with two methods:

- Beacons: This method is based on the usage of a Bluetooth Low Energy beacon to send a signal to all the possible receivers in the range, those receivers are normally phones. This method was used first with a companion app that when received an identifier performed an action[31]. The communication between the parts is only a 1-way communication between the emitter and receiver. The position can be obtained by two methods:
  - Fingerprinting[32]: This method is similar to the ones used in Wi-Fi, where several signals, are received and by knowing the position of the emitters, the position of the receiver can be calculated. It normally allow accuracy between 0.5 and 1.5 metres.
  - Approximation: This method is highly dependable on the number of beacons as the position is calculated by approximating to the position of the beacon with higher signal. By doing this the computation needed and the dependency to the environment changes are reduced.

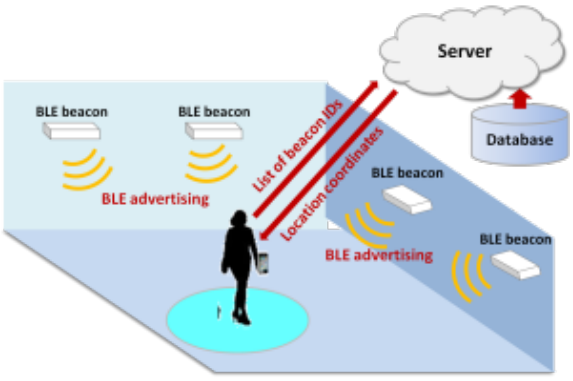


Figure 3.5: Bluetooth Low Energy indoor positioning

- Bluetooth 5.1: The last version of Bluetooth make use of the angle of arrival, which with the angle of the signal can determine the position of the device with the MUSIC algorithm.

3.1.4 RFID

Radio-Frequency IDentification is a protocol designed mainly for proximity identification the development and the improvements allow to detect the position of the object with a medium accuracy. The main usage of this technology is to make invisible barriers from which when crossed the position is changed, but it can be implemented an active tracking. This technology is composed by a emitter and a receptor (tag), which can be either active or passive. With the active tags the distance for the detection can be increased so an active detection can be performed, while with the passive RFID tags it need method very similar as the Bluetooth beacons[33, 34].

3.2 Object tracking:

Knowing the movement of an object has been a very interesting subject for the localisation of object, as they can be followed thought time, this can be done either with sensors in a suit, which gives good reliability, or with non-invasive techniques as computer vision. The main objective for the tracking is to know the movement of different parts of the body.

In the first uses of this technology the use of markers were mandatory, the markers indicate the position of the parts. The first development in this type of technology were the increase of tracked points, including the facial landmarks. This technique is nowadays very used in the cinematic industry in order to transfer the emotions and movements from a real person to an animated character using a technique called Computer-generated imagery (CGI), which has been used in films as Tron, Matrix or Avatar. The usage of markers has a problem, they need to be carried by the person tracked. That problem was solved with the development of the detection technology, this development was done by using computer vision combined with neural networks that allow the detection without any marker.

There has been a lot of comparison between marker and marker-less techniques [35], the main conclusion of the comparisons were that the marker-less techniques had less reliability but the reduction on the reliability was not comparable with the improvements that this type of development can give. The main methods for object tracking are:

3.2.1 Motion capture

This method consist on capturing the movement of a person many times per second, although it started with using cameras for having the position of every joint [36], nowadays it is done by using special suits with sensors. The main advantage is the quality of the final result of the capture, but the disadvantage is the price and material necessary for the detection, another disadvantage is the low quantity of the subjects that can be tracked.

3.2.2 Object Detection

The object tracking can be done with the information retrieved from the position of the whole object, by storing it's position on a list and comparing it in different movements it can be done a movement tracking, the data of the object can be retrieved from any of he methods in the section "Object detection". In the figure 3.6 it can be seen the path followed by the person, being the blue points the most recent data and the red the most outdated points.

Figure 3.6: Movement tracking with object detection

3.2.3 Accelerometers

This method use the the acceleration created when a person is moving to detect the direction of movement and sent that information to a server. This method has several problems calculating the direction based only on the acceleration, reason from which this method normally complements other to have a more reliable reading with lower movements. This devices show a relatively good behaviour in short movements while it gets worse with longer movements. This method needs to have a starting point for reference and then it calculates the relative movement from the previous time-stamp. [37, 38, 39, 40]

3.2.4 Pose detection

This technique is used to know the position and orientation of a 3D object from a 2D source. This technique is applied to movement detection by detecting the joints in each person and computing the movement of each point in a different list. This method has better accuracy by having more points detected than image object detection[41]. Traditional pose detection has been done in single person environment with probabilistic model[42], but it has been improved with the inclusion of different methods:

- Estimating key-points in heat-maps: This is the most used method, the estimation is based on having a sub-network to classify the heat, in which first it reduces the resolution to match the input and then it estimate the heat-maps with a regression, finally the key-points are multiplied to be shown in the full image.
- Regressing position of key-points: This method is based on supposing a value of a variable, this makes a variation on the values of the parameters in the regression function.

There are several frameworks to do pose detection:

- High to low: This process aims to generate low resolution and high level of representation. In order to obtain better results this process can be done several times. Some examples for this process can be implemented using the ImageNet classification.
- Low to high: This process aims to generate high resolution and high level of representation. In order to obtain better results this process can be done several times.Some examples for this process can be implemented using the with bi-linear up-sampling.
- Multi-scale fusion: This process is based on giving images with different resolutions to different networks and compute the output by doing the union of them. This type of method often combines the two previous process. Some example of this method are the Hourglass and the cascaded pyramid network.
- Intermediate supervision: This method was develop in first place for image classification and used later for helping deep neural networks to improve the estimation and reduce the training. The most used approach with this method is the Convolutional pose machine one.

Some of the data-sets for this are:

- Human Pose[43]: this is a state of the art benchmark which include 25000 images with 40000 people doing different 410 human activities with the proper label for each image
- YouTube Pose[44]: This data-set comes from 50 YouTube videos and covers several activities that normal people do. It has one hundred frames from each video annotated with the joints, it only has the annotations for the upper body joints.

- Extended BBC Pose[45]: this data-set contains videos from the British television channel BBC from the BBC pose data-set as well as 72 training videos, in total it has 92 videos with around 7 millions frames.
- Pose track[46]: This is a data-set normally used for bench-marking contains pose estimation and articulated tracking video. It has both training and validation sets. It has more than 500 video sequences, more than 20000 frames and more than 150000 annotations.

Some of the python libraries used in pose detection are:

- deep-high-resolution-net.pytorch[47]: This library is tries to solve the problem of human pose estimation with high resolution representation without any resizing. It has a high resolution sub-network and adds high to low sub-networks to have more stages, having the multi-resolution networks in parallel [48].
- OpenPose[49]: It was the first real-time multi-person system to have the joint detection. It was developed only for non-commercial research by the Carnegie Mellon University.[50]
- Pose-net[51]: It was first one to develop the pose detection inside the browser although it has implementations in python. It works both for single or multiple people. It uses the fast greedy decoding[41].

Figure 3.7: Pose Detection

3.3 Emotion detection:

The emotions are the psychological experiences that affects our life, this are complex process that involves large number of components and can be externally shown in different ways. Modelling the emotions has been a very difficult task, with various models proposed, but no model has been taken as universally acceptable. Emotion detection is the process of knowing an approximate emotion that a human has. This is a very difficult task as each human act different with the emotions so there is a huge variability between the people. This task is normally based on detecting the basic emotions, they are defined as the main emotions that the humans can have,being the rest of the emotions a combination of the other. This emotions are:

- Happiness: Is defined as a pleased emotional state, it is normally expressed through facial expressions as smiles, relaxed stance body language, upbeat tone of voice.
- Sadness: Is defined as a transient emotional state it can be expressed through quietness, lethargy, cry and facial features as excessive blinking.
- Fear: Is defined as a powerful and important function to protect the person from danger, it is very related with the fight or flight response. It is characterised by tense muscles, increases in the heart and respiration rate, facial expression as widening eyes or pulling back the chin.
- Disgust: Is defined as a revulsion or disapproval to something. It is characterised by going away from that something, it is characterised with reactions as vomiting and facial expressions as nose wrinkling or lip curling.
- Anger: It is a powerful emotion that can be related to the fight or flight response. It is characterised by facial expressions as frowning or glaring, strong stance with the body and sweating.
- Surprise: Is a brief emotion related to something unexpected, it is often characterised by facial expressions as raising the eyebrows or opening the mouth, and physical responses as jumping back.

In the emotion detection the data taken is translated to two components, arousal and valence, from which the emotion can be translated as it can be seen in the figure 3.8, this components are used to differentiate the emotions in a subjective way, this allow the emotions to be detected objectively. It also allow to work with the emotions as any other data, which can be used for predictions or engagement. The centre zone is taken as neutral as the differentiation of emotions in that part will be very difficult because the proximity of the borders.

- Valence: As it is defined in "The emotions" [52] the valence in psychology means the intrinsic attractiveness (positive valence) or adverseness (negative valence) of an event, object, or situation for a person, this is used to complete the arousal theory to distinguish from positive to negative emotions as some of the emotions can appear very similar.

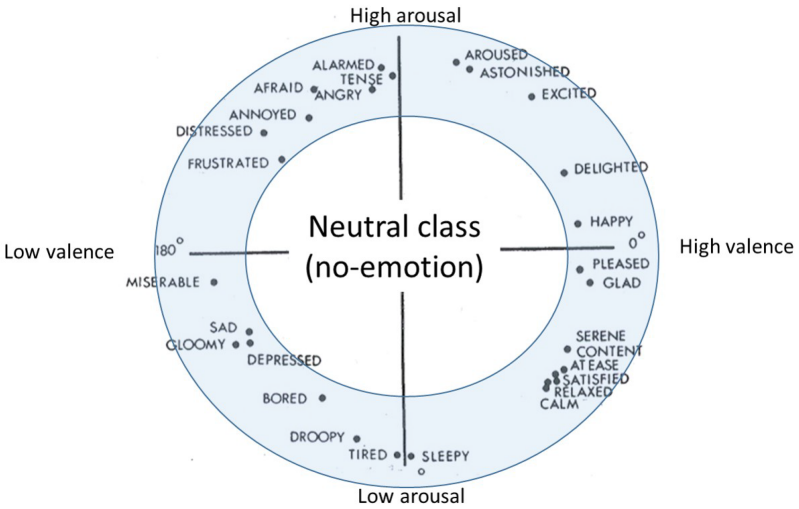


Figure 3.8: Arousal vs valence chart

- Arousal: This component is the state of being able to sense stimuli and perceive the data around the person, it involves the activation of the part of the brain in charge of the automic nervous system, which is the one that controls the hearth rate and blood pressure. The arousal is very related to the attention, alert and consciousness as well as primal instincts as fight-or-flight response. The James-Lange theory[53] was the first time that the emotions has been related to the arousal, which with the six primal emotions previously explained can be related as it can be seen that they are affected by the same behaviours.

The automatic recognition uses different models as:

- Bayesian networks: This is a probabilistic graphical model that represent the dependencies of some variables with a graph. For this type of techniques to work perfectly it need to be specified the probability distribution function on each part, normally represented by Gaussian functions. The main algorithm used for implementing this technique in machine learning was developed by Rebane and Pearl, it uses 3 nodes, with 2 representing the same dependencies, another methods make use of optimisation based search, which make use of a function for rewarding.

- Gaussian Models: this is a probabilistic model for representing small quantities of objects inside a bigger group, it is include inside the family of mixture models, it represent the probabilistic distribution of observation in the big group.

- Markov models: It is an statistical model which assumed to have a Markov process without observed states. This technique is especially used in reinforcement learning and pattern recognition, it can be considered a generalisation of the Gaussian mixture model.

3.3.1 Image

Artificial vision has been the main framework for the study of emotion detection, this is due to being a relatively easy and non invasive method. This method is based on detecting the facial landmarks on the people and comparing them with machine learning to the main emotions[54]. As it can be seen in the descriptions of the basic emotions they are very related to the facial expressions as well as the movements of the person, this is one of the reasons to use vision techniques and to develop a model to detect the emotions of the people. As in other task every time that the neural networks are involved databases are needed, some of the most important ones are:

- Cohn-Kanade extended (CK+)[55]: the extended Cohnkanade databases is the one that is most used for research purposes, it contains 593 videos from 123 different people with a variable duration. Only 327 sequences are labelled with 7 emotions (anger, contempt,disgust, fear, happiness, sadness, and surprise).[56, 57]
- MMI[58]: This database contains 326 videos of 32 people with 213 videos labelled with the 6 basic emotions mentioned above.[59]
- Toronto face data-set (TFD): This is is the union of several facial expression databases, it contains 112234 images, one third of them is annotated with the same 7 expressions as the first data-set, all the faces has the same size and put at the same distance.[60]

The main steps for the expression recognition are:

1. Pre-processing: This step is done to reduce the variation from different faces that are not important to the emotion detection, some of this features are colour, position, illumination and background. There are three parts that are done to eliminate this features:
    - Face alignment: This part can be done with several methods as Holistic, cascaded regression or deep learning method. This works by detecting different points on the faces and then rotating the face to be in the desired position, with this known, the methods that can perform better with less number of points are the deep learning based ones, although the cascaded regression can be faster and with nearly the same results.
    - Data augmentation: As the next steps will be going to be performed using neural networks they need enough training data, as most of the times they do not have enough training data the input images can be changed to help the neural network to have more training material. Normally this part is embedded in the neural learning tool-kits. In this step the images are randomly cropped, flipped, and colour changed, this can allow the neural network to have up to ten times more data to work with.
    - Normalisation: This task is responsible for the performance being as good as possible by reducing the variation in illumination and position. In the case of the illumination can affect to the intra-class variations. The most used illumination normalisation techniques are based on the isotropic diffusion, the discrete cosine transform or the the Gaussian difference. In the case of the pose normalisation it tries to compensate the movements of the poses, the most used technique is based on making a generate all the facial components from the facial landmarks in the position that is desired.

2. Deep feature learning: As it has been covered in the state of the art the neural networks had been highly used in the and researched as they allow high level of performance with different architectures, CNN, which has been explained before and as in other computer vision this type of neural networks are the most used ones, but not the only ones as it can be used:

    - Deep belief network (DBN): It make use of a graphical model which learns to have the representation of the data. It is built with two layer generative models, with one visible layer and another one a hidden one, the units in the higher layer are the ones that learn the dependencies between the layers. In this networks the training start with a layer greedy learn to improve the results with lower labelled data, while the contrastive divergence is used to approximate the likelihood.
    - Deep auto-encoder (DAE): It was the first to learn efficient coding for dimensional reduction. This type of network in contrast to the previous ones is trained to minimise the input error, .
    - Recurrent neural network (RNN): This model captures temporal information being more used for sequential data prediction, because of that the learning need the time steps that share the same parameters, for that it is normally trained with BPTT.

3. Feature classification: After the pre-processing and training of the network the final part is to assign each emotion to the face this can be either performed at the same time of the feature extraction or at a different time. If it is done after the feature extraction it has to add a loss layer to reduce back error propagation.It can be also done using a neural network, normally a CNN to have feature extraction and then use a SVM or random forest to make the classification.

The detection of emotions can benefit from the time correlation as while locking at an image can be outputted a result that with some temporal correlation can be changed. In that case it consist in three steps:

- Frame aggregation: As the emotion facial expression is built through the time, it is not suitable to measure the emotions for each frame. There are two methods depending on the step where the aggregation is performed, the feature level, where the information for the facial features has to be put in a time diagram, and the decision level, where the information which has to be put in correlation with the time is the output of the features and comparison. The problem of this part is having constant length vectors as the number of frames in which a face appear is variable. To solve that problem an averaging of the frames data can be done or the data can be repeated, but there are a way in which it is not needed to have a constant length, this is statistical coding which act calculating probabilities in each frame.
- Expression intensity: Normally in the recognition methods the expressions are detected when there is a high peak in the intensity of that emotion, for detecting different expressions intensities a neural network can be trained to detect the low peaks in the emotion expression.
- Spatial-temporal networks: Taking into account that the aggregation takes all the frames and computes the information in some temporal order it has not a direct temporal dependency, in this part the frames in a temporal window are seen as one input and is used to detect more subtle expressions. The neural networks that can perform such operation are:

  - RNN: this neural networks can have the needed information using the connection from the features in different frames, there are an improved version LSTM which can handle variable length data. This is composed with ReLUs and gives a simpler mechanism for solving the correlation of features in different frames.
  - Cascaded: It is based on the combination on the vision representations from CNN and combine it with LSTM for the inputs to been able to have a variable-length. Instead of the CNN it can be more flexible if it is used a Resnet network which allow contact between the lower layers of the CNN and LSTM
  - Network ensemble: It use a CNN with two streams training one of them for temporal information and the other for feature extraction. It can be performed with a SVM-fusion and a neural network fusion as well.

3.3.2 Body signal measurement

This method which has been traditionally a more invasive method is based on using the body signals to determine the emotion of the person, although there has been a development on ways of obtaining the same data from distance, as it can be the infrared cameras for the temperature or the reflection of Wi-Fi signals, it is not very usual to obtain the data without direct contact with the person as the data will have noise[61]. The relation of emotion with the body signals can be seen as the bodies have two types of systems sympathetic and para-sympathetic, the first system is highly related to some of the topics that has been pointed before as the fly or fight response and very related to the whole emotions subconscious. Some of the organs that this system controls are the eye, hearth, lungs, vessels, sweat glands or the digestive track. The main body signals that are used for the emotion detection are:

- Hearth beat: This, along with the breathing rate, is the main focus for this topic, this is because they provide very reliable signals and are very easy to obtain. As it was said before it is controlled by the sympathetic system so it has a valid relation with the emotions. In this method the ECG signal is computed with the separation of the PQRST components, which can be seen in the figure 3.9[62], this points are the ones that can give the information of the heart states and because of that the type of beat. The signal of the heartbeat for different emotions are computed in the diagram 3.8, as it can be seen it depends on the valence and arousal of the signal, with this the ECG can give the data of the heart and with that data the levels of both the arousal and valence can be computed.

  The computation is done in three steps when using neural networks:

  - Preprocessing: The preprocessing tries to eliminate the noise that comes from the act of measuring the data, it also perform an spatial and temporal filtering.
  - Feature extraction: The most important information from the measurements are extracted, as with another neural network the features needs to be normalised and synchronised in phase. Prior to the classification the features needs to be taken with an historical log, as the frames in the videos, the signals in the body does not appear instantaneously they are built thought the time, and that data can determine different emotions of the person.

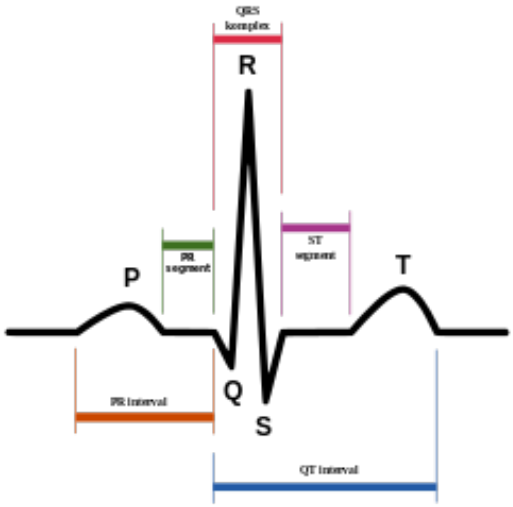


Figure 3.9: Electrocardiogram with P, Q, R S and T points

  - Feature classification: In this part the features can be related to the emotion. As it was explained before the features extracted with the neural networks are the arousal and valence, which are classified to obtained the emotion as it can be seen in the figure 3.8.

- Breathing frequency: The breathing rate is regulated by the sympathetic system. Breathing rate can be very easy to see when someone has fear or is surprised but no so well when it is happy. The method for the detection of emotions are very similar to the hearth rate detection. This method, although being able to work alone is normally used in conjunction with the heart rate to reduce the error.

- Galvanic skin response: This is the measure of the sweat gland activity, which are directly reflection of our emotional state, it includes also the skin conductance as a measure, this has not a very good way to differentiate the exact emotion but the main benefit is to have a good differentiation of the intensity of the emotion. This signals are obtained with electrodes that measure the electrical activity of the skin and the sweat glands. This method normally is a companion, normally to the two before to have a more reliable measure of the arousal.

- Body temperature: The signal that the humans had used to determine their state is the temperature. The first uses for this type of measurement was to detect some diseases as fever, but with the development in both the measurement hardware and the research between this data and the state of the body and mind. As it can be seen in the figure 3.10 the temperature changes not only on the quantity also it change in the distribution, this allow to be detected with temperature cameras but requires that the people are set apart with some space between them as the temperature is transmitted from one to another. In this case this method has it's own classification as the features extracted are the quantity of heat and the position of that heat.

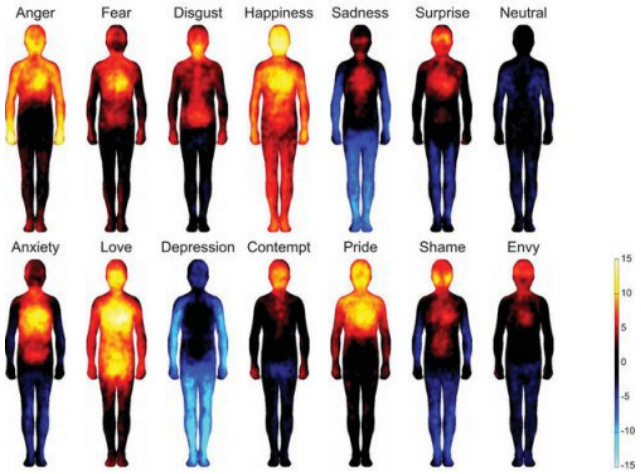


Figure 3.10: temperature for different emotions vs valence chart

The majority of the techniques uses direct sensors places in the people but this is not a good method when several people needs to be detected and also, to solve that there are a new technique that make use of wireless signals to detect the hearth and breathing beats. This technique develop by the MIT team "EQRadio"[63] make use of the reflection of the waves to capture the data of the movement of the chest, this data contains a wave with mainly two frequencies:

- Hearth rate: This is a high frequency signal that shows all the movements of the hearth, it is not as reliable as the ECG but with some computation it can give a reading of the four states of the hearth.
- Breathing rate: This is a low frequency signal that shows the movement of the chest that is induced by the inflation of the lungs

This technique had an accuracy of detecting the same hearth and breathing data of 87%, which means having a good reliability with a technique that is not invasive.

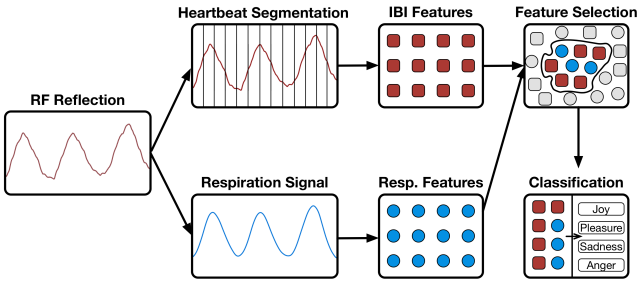


Figure 3.11: EQ-Radio

3.4 Attention analysis:

The place where a subject has it's attention is a very useful data as it can be used in many fields, being the most important ones security and audience measurement, the fist one implementations are mainly related to driving, as it is very important to detect if the driver is distracted, in the second topic the attention analysis is used detect the place where the object or action that takes the interest of people the most. The main techniques can be divided from the method used to obtain the data:

- Vision techniques: This method obtain the data with cameras, the image of the cameras is computed as in the other parts of the project to obtain data, in this case the data wanted are the position of different parts of the face. This method uses artificial vision to compute the position of those parts and then compute the direction of attention with that data.

- Sensor techniques: This method was the first one used, and it has been hardly used in recent years. It obtained the data of the position of different parts of the face using sensors, this sensors can track the position and communicate the data to a computer to determine the direction of attention.

The detection of attention in images has been a heavily studied topic in recent years, but there has not been huge development with the spatial-temporal data in the videos for attention analysis. The majority of techniques that are applied to videos make uses of vision techniques for images in each frame, depreciating the information that can be obtained by the correlation of different frames. The information of the correlation of frames can be used to know the habits of the people and extrapolate behaviours from it, as when a person is looking to the toilets there are a huge probability of him going to the toilet in the near future.[64]

For the detection of attention with sensors there is the necessity of having the sensors in the people, which is not suitable for real world applications, so from here the main focus of attention analysis will be with vision techniques. This techniques are based on detection the main points of the face, with more points detected the reading will be better, but also require higher computation.[65] The main parts of the face to perform the attention detection are:

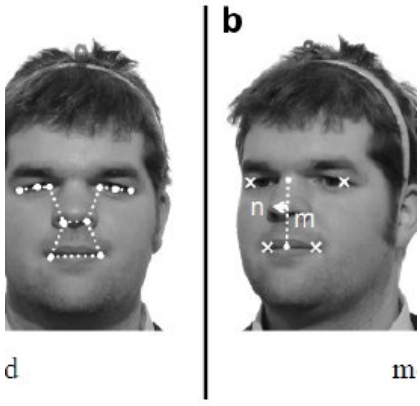
- Eyes: This points are ones of the most important, this can be used to either detect the pupil and extrapolate the direction from it, which is very difficult due to it's size, or to compute the inclination of the face and the midpoint of the eyes for a less difficult approach.

- Nose: This point is used to detect the direction of the attention in an axis, differentiating from right, left or front.

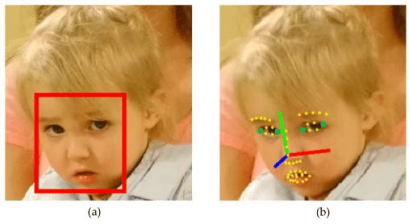
- Mouth: In this case several points can be detected and it can be used to improve the data of the previous points in order to perform a better approximation and to detect not only one axis but two.

- Ears: This points are use for the same purpose as the mouth points, but in this case they are not as used as the previous ones because the difficulty of detecting the ears compared to the mouth.

- Eyebrows: In this case several points can be detected and it can be used to improve the data of the previous points in order to perform an even better approximation and to detect more than two axis.



(a) Head pose with some points



(b) Head pose with all the points

The main techniques and models are the same as for object detection using artificial vision as they are mainly based on detecting different parts of the faces. In this case the object that are looking for are the centre of the faces or the facial landmarks, and with the position and distances of them been able to calculate the direction of the face. [66]

More developed techniques use the eye's pupil to have the direction where the pupil is pointed at, the drawback of this techniques are the computational power necessary for the pupil detection,as it is a smaller object than a whole face so the detection needs a higher segmentation to perform equally, although the results are better as the head pose is not always pointing to the same place as the eyes.

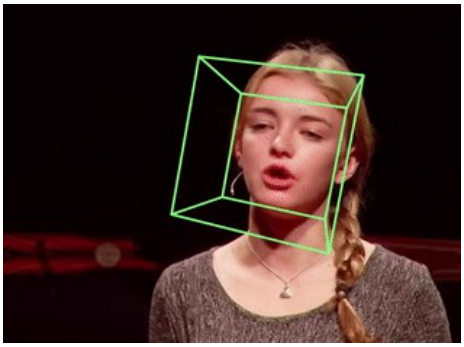


Figure 3.13: Head pose

3.5 Technology comparison

In this section a comparison of all the techniques will be done for each of the parts of the project, in the comparison it will be taken into account the following parameters:

- Accuracy: This parameter will analyse the overall detection taking into account the ratio of detected vs non-detected, the ratio of false detection vs good detection and the necessary characteristics for the object to be detected.
- Hardware to be implemented: This parameter will analyse the minimum hardware to implement, this comparison will only include hardware that other methods do not use to have a easier comparison.
- Computational cost: This parameter will analyse the computational necessities for the method, this comparison will be important to determine the minimum hardware to be implemented.
- User interaction: This parameter will analyse the necessities for the person detected to carry some gadget or perform determined actions prior to the detection.
- Possibility of usage for others parts: This parameter will analyse if the same method can be used in different parts of the project, reducing the number of methods and so on simplifying the overall solution. This will include also the tweaks to be used in other techniques including the reduction of information, operation with the results or storing the information in variables for a later computation.

3.5.1 Object detection

The main methods for the detection of objects are divided into five categories depending on the technology used, the order correspond to the best technique taking in account the previous parameters.

1. Artificial vision:
    - Accuracy: The accuracy of this method depends on the technique used, it can go from a good accuracy in general situations to a very good accuracy in very deterministic situations by the use of neural networks. Normally this techniques can be easily adapted for the situation.
    - Hardware: Only the cameras are necessary for this method to work.
    - Computation: It depends on the technique used, but normally this is a very high computation method.
    - User: The user does not have to perform any action or carry any gadget with him to be detected.
    - Other uses: The majority of the techniques in this method can be used to perform movement tracking, but it can not distinguish different parts of the person.
2. Wi-fi and Bluetooth:
    - Accuracy: The accuracy of this method is very dependable on the technique used, having around one or two meters in some techniques and tens of centimetres in others.
    - Hardware: It needs that the person or object to be detected carry a device that can detect the radio-waves and in the majority of the cases the device needs to connect to a server to send that information.
    - Computation: This is closely related to the accuracy, the techniques with higher accuracy needs higher computation, but in all the cases it needs a dedicated server to have all the information and compute the position of the person.
    - User: There are methods which requires an app in the device, but also passive methods which does not require anything apart from having that connections enabled.
    - Other uses: The majority of the techniques in this method can be used to perform movement tracking, but it can not distinguish different parts of the person.
3. RFID:
    - Accuracy: This method was created to detect if an object has gone through a determined place, so the exact position accuracy is not very good but the zone accuracy is quite good. There has been some experiments to have the exact position but the accuracy goes to several meters.
    - Hardware: It needs that the person or object to be detected carry a device that receive the RFID signals and in the majority of the cases the device needs to connect to a server to send that information.
    - Computation: This method because the simplistic approach that takes does not need a huge computation power.
    - User: The user needs to carry an RFID capable device that can connect to a server.
    - Other uses: The majority of the techniques in this method can be used to perform movement tracking, but it can not distinguish different parts of the person.

3.5.2 Movement tracking

The main methods for the tracking of movements are divided into two categories depending on the technology used.

1. Accelerometers:
    - Accuracy: This method is the one that has lower accuracy due to the fact that it computes the position from the movement.
    - Hardware: It needs that the person to be detected carry an accelerometer with connection to a server.
    - Computation: This method is quite computational heavy as it needs to take into account an historical of both movements and positions.
    - User: The user needs to perform a position calibration prior to the detection, this is because the position is computed by the movement and for that it needs a starting point.
    - Other uses: The majority of the techniques in this method can be used to perform position detection.
2. Pose detection:
    - Accuracy: The accuracy of this method depends on the technique used, it can go from a good accuracy in general situations to a very good accuracy in very deterministic situations by the use of neural networks. Normally this techniques can be easily adapted for the situation.
    - Hardware: Only the cameras are necessary for this method to work.
    - Computation: It depends on the technique used, but normally this is a very high computation method.
    - User: The user does not have to perform any action or carry any gadget with him to be detected.
    - Other uses: The detection of pose is done by detecting different joints, this allow the method to be used in person detection, face detection or attention analysis. If the face points is increased it can also perform emotion detection.
3. Motion capture from position:
    - Accuracy: The accuracy of this method depends on the accuracy of the technique for the position detection.
    - Hardware: This is completely dependable from the technique of detection.
    - Computation: Apart from the computation of the position it carries the computation of the historical analysis of the position.
    - User: This is completely dependable from the position detection.
    - Other uses: this technique can be used for object detection, other uses are dependable of the technique used.

3.5.3 Attention detection

The main methods for the detection of attention are divided into two categories depending on the technology used.

1. Facial landmarks:
    - Accuracy: This method has a very good accuracy as it detects more than 60 points in the face.
    - Hardware: It is a vision technique so it needs cameras to obtain the images.
    - Computation: It has been developed mainly for single person usage, and with that is a very heavy computational method, for multiple people it will be even higher.
    - User: The user does not have to perform any action or carry any gadget.
    - Other uses: It can be used for face detection, object detection and movement detection as it detects the face and points in it.
2. Facial points:
    - Accuracy: This method has a good accuracy depending on the number of points detected, for a good detection in one axis (right, front, left) it can be performed with three points, eyes and nose.
    - Hardware: It is a vision technique so it needs cameras to obtain the images.
    - Computation: It needs lower computation than the one above as it compute less points in the face.
    - User: The user does not have to perform any action or carry any gadget with him to be detected.
    - Other uses: It can be used for face detection, object detection and movement detection as it detects the face and points in it.

3.5.4 Emotion detection

The main methods for the detection of emotions are divided into two categories depending on the technology used.

1. Artificial vision:
    - Accuracy: The accuracy of this method is quite disappointing, it is normally applied for single person detection and the detection is made for only six or seven emotions. The detection of those emotions is quite low and need very exaggerated expressions.
    - Hardware: As this is a computer vision technique it needs cameras to obtain the images.

- Computation: Depending on the number of points detected from the face that are introduced in the neural network to compute the emotions it can have different computation needed. This method normally is performed with one person and need very huge computation so for several people it will be even greater.
- User: The user does not have to perform any action or carry any gadget with him to be detected.
- Other uses: As this method make use of the facial landmarks it can be used for face detection and person detection. With the data of the relative position of the facial landmarks the attention detection can be computed. With the data of the position of the face in different moments the movement of the face can be tracked.

2. Body signal:

- Accuracy: The accuracy of this method is enough for detecting the 6 basic emotions. In the case of obtaining the signals wirelessly it has lower accuracy because of the interferences of the data obtained.
- Hardware: The hardware needed depends if the method obtains the signals with sensors, which will be needed, or it is obtained wireless so the emitter and receptor will be needed.
- Computation: The computation mainly depends on the number of signals that are used for the detection, increasing the number of signals will increase the computational power needed. Cleaning the signals will also increase the computational power needed when they are obtained wireless to get rid off the noise.
- User: The user does not have to perform any action, but in the case of the wired sensors it needs to carry them with him to be detected.
- Other uses: It can not be used in any other part of the project.

Actions	Objective	Method	Techniques	Observations
Person Detection	The objectives of this part are: - Know how many people are in the room - Know the position of all the people detected as either: + Coordinates in the room + Pixels of the picture.	Vision	Neural network with CVlib	In normal videos with good quality it has good performance, but it detect other objects as it lacks for a object discretization, which can be solve by either implementing it or using a leaning model with only people to detect. In the night club videos because of the conditions of dark and moving lights the detection of object is very low, around 4% but the object discretization is no longer needed.
			Neural network with Image AI	In normal videos with good quality it has medium performance, but it detect other objects as it lacks for a object discretization, which can be solve by either implementing it or using a leaning model with only people to detect. It has the feature to change the speed of the detection with the trade of the accuracy. In the night club videos because of the conditions of dark and moving lights the detection of object is very low, around 2% but the object discretization is no longer needed.
			Neural network Darknet	In normal videos with good quality it has medium performance, but it need a very long time to compute the frames. It has the feature of object dicretization built-in. In the night club videos because of the conditions of dark and moving lights the detection of object is very low, around 4%.
			Neural network For Pose detection	Although the libraries for this techniques were designed to detect joints the majority of them order the data per person, with this the number of people detected can be known and the position of the center of the joints can be taken as the person position.
			Neural network for Face detection	As the case for pose detection the libraries in this case were not designed for that but with the data of the face position and number the faces it can compute the required information.
		Wifi	Fingerprinting (FIND3) with triangulation in server	It requires an app in the user device to have it's position, it computes the wi-fi signals strength and with the data of the position of the 3 strongest access points it has the position of the device
			Fingerprinting (FIND3) with approximation to the beacon position	it requires an app in the user device to have it's position, it computes the wi-fi signals strength and with the data of the position of the strongest access point it has the position of the device
			Angle of arrival	The access point has several antennas to detect the angle where the signal is coming, with that info it has the exact position.
		Bluetooth	Beacon fingerprinting (FIND3) with triangulation in server	It requires an app in the user device to have it's position, it computes the wi-fi signals strength and with the data of the position of the 3 strongest access points it has the position of the device
			Beacon fingerprinting (FIND3) with approximation to the beacon position	it requires an app in the user device to have it's position, it computes the wi-fi signals strength and with the data of the position of the strongest access point it has the position of the device
			Bluetooth 5.1: Angle of arrival	The access point has several antennas to detect the angle where the signal is coming, with that info it has the exact position
		RFID	Door approximation	It has low reliability as it either needs an active tag or has low accurracy, it is because of consisting on detecting if it has gone trough a place
		Accelerometers	Movement variation	It consist on measuring the orientation and acceleration to determine where to move it needs an inicialization point to compute all the movements

Table 3.1: Comparison table

Actions	Objective	Method	Techniques	Observations
Movement detection	The objectives of this part are: - Know a log of previous positions of the person - Detect the movement of parts of the body separately: + Head + Chest + Arms + Legs	Vision	Vision techniques for object detection + tracking	When triying to calculate the movement the main part is the position detection with that information in a log the movement can be computed, with that information it can be seen the type of movement as walk, run or dance. The main problem with this is having only 1 point that the movement detected is the overall movement, this has a low reliability and a limited detection but is a good aproximation
			Vision techniques for face detection + tracking	When triying to calculate the movement the main part is the position detection with that information in a log the movement can be computed, with that information it can be seen the type of movement as walk, run or dance. The main problem with this is having only 1 point that the movement detected is the overall movement, this has a low reliability and a limited detection but is a good aproximation, this one has even lower reliability than whole body detection because the type of movements of the human body.
			Joint detection + tracking with Pose-net	This type of detection is based on having several points detected on the body, 17 in total (eyes, nose, ears, shoulders, hips, elbows, wrists,ankles and knees), with this the movement is much more precise and allow the distinction of more movement as there are a discretization of different body parts.
			Joint detection +tracking with Deep high resolution pytorch	This type of detection is based on having several points detected on the body, 15 in total (head, shoulders, hips, elbows, wrists,ankles and knees), with this the movement is much more precise and allow the distinction of more movement as there are a discretization of different body parts.
			Joint detection + tracking with OpenPose	This type of detection is based on having several points detected on the body, 70 in total, with this the movement is much more precise and allow the distinction of more movement as there are a discretization of different body parts. This is a propietary system with no other possible use than appart from the investigation.
		Wifi	Position detection + tracking	When trying to calculate the movement the main part is the position detection with that information in a log the movement can be computed, with that information it can be seen the type of movement as walk, run or dance. The main problem with this is having only 1 point that the movement detected is the overall movement, this has a low reliability and a limited detection but is a good aproximation
		Bluetooth	Position detection + tracking	When trying to calculate the movement the main part is the position detection with that information in a log the movement can be computed, with that information it can be seen the type of movement as walk, run or dance. The main problem with this is having only 1 point that the movement detected is the overall movement, this has a low reliability and a limited detection but is a good aproximation
		RFID	Position detection + tracking	When trying to calculate the movement the main part is the position detection with that information in a log the movement can be computed, with that information it can be seen the type of movement as walk, run or dance. The main problem with this is having only 1 point that the movement detected is the overall movement, this has a low reliability and a limited detection but is a good aproximation.
		Accelerometers	Position detection + tracking	When trying to calculate the movement the main part is the position detection with that information in a log the movement can be computed, with that information it can be seen the type of movement as walk, run or dance. The main problem with this is having only 1 point that the movement detected is the overall movement, this has a low reliability and a limited detection but is a good aproximation

Table 3.2: Comparison table

Actions	Objective	Method	Techniques	Observations
Face detection	The objectives of this part are: - Know the position of the faces in the room - Know the main parts of the face: + Eyes + Nose + Ears + Mouth	Vision	Neural networks with CVlib	In normal videos with good quality it has good performance. In the night club videos because of the conditions of dark and moving lights the detection of object is very low, around 0.5% , which can be improved by using a leaning model with faces in that environment.
			Neural networks with face_recognition	In normal videos with good quality it has good performance. In the night club videos because of the conditions of dark and moving lights the detection of object is very low, around 0.5% , which can be improved by using a leaning model with faces in that environment.
			Neural networks for Pose detection with some facial landmarks	Although the libraries for this techniques were designed to detect joints the majority of them also detect different parts of the face such as eyes or nose, with this the faces of the people can be detected as well as the position of the faces for the position.
Attention detection	The objectives of this part are: - Know the head pose - Know if the person is looking to a place	Vision	Detection of facial features and extrapolate x and y axis	Depending on the accuracy needed this method is based on detecting the eye position as well as other face parts as nose or ears to detect the direction of atention, This method can be easily implemented with that 3 data and depending on the relation between the center of the eyes and the nose as well as the inclination of the eyes it can extrapolate if the person is looking to the front, left or right, combining that with the person position it can extrapolate to which place in the room the person is giving it's atention.
			neural network to extrapolate the x,y and z axis	This method is more computational demanding than the one above as it tries to detect either the facial landmarks and from that extrapolate atention direction or directly from the image of the face extrapolate with the neural network the direction. It gives much more precision and add another axis of detection.
Emotion dettection	The objectives of this part are: - Detect the main sources of emotion for each technique. - Detect the emotion and mood of the people in the room	Vision	Use neural networks with the facial features	This method has been very used, it consist on using the facial landmarks of the face to extrapolate the emotion, the extrapolation is done with a neural network trained with the data of that landmarks for the 6/7 basic emotions, the problem with this method is the reliabily as it has a medium detection and to achive that it needs very exagerated expresions.
		Body signals	Use direct body sensors in the body and neural networks to extrapolate the emotion	This is based on extrapolating from the data of different body signals the emotion from the 6/7 basic ones. This need direct contact with the body wich can be a bit anoying, The data is introduced in a neural network to know the emotion.
			Use reflection of waves to obtain the body signal and neural networks to extrapolate the emotion	It uses reflection of waves to detect the body signals previously detected with sensors, it detects both the hearth rate and the breath rate, then the process is the same as before to detect the emotion.

Table 3.3: Comparison table

Technique	Position detection	Movement detection	Attention detection	Emotion detection
Neural network with CVLib	✓	✓		
Neural network with Image_AI	✓	✓		
Neural network with Darknet	✓	✓		
Neural network with Face_recognition	✓	∼		
Neural network with Pose-net	✓	✓	✓	
Neural network with OpenPose	✓	✓	✓	
Neural network facial landmarks detection	✓	∼	✓	✓
Deep high resolution pytorch	✓	✓	✓	
Radio-waves with triangulation (WI-FI +Bluetooth)	✓	✓		
Radio-waves with approximation (WI-FI +Bluetooth)	✓	✓		
Radio-waves with angle of arrival (WI-FI +Bluetooth)	✓	✓		
RFID door approximation	✓	∼		
Accelerometer movement variation	✓	✓		
Body signal sensors				✓
Body signals with radio-waves reflection				✓

Table 3.4: Techniques vs utils

Chapter 4

Proposed case and technology adaptation

This chapter will be divided in two parts, the first one will analyse the case in which case the project will be used, explaining the different parts of the case and the conditions of it, and the second part will analyse the changes done to the technologies in order to be suitable to be used on the proposed case, this analysis will be divided in the implementation done, the different test to prove the behaviour of the technique with the implementation tweaks, and the conclusion of each one of the techniques.

4.1 Proposed case

The proposed case for the project is to develop a system that, when placed in a room in which a show is being made, is able to detect the engagement of the people in the room, for that purpose as the existing technologies have some disadvantages for being used a hybrid approach has been chosen. This hybrid approach will be implemented using the combination of vision and radio-waves techniques.

Vision techniques are some of the most developed techniques nowadays and require low quantity of specialised hardware, the main disadvantage of this type of technique is the requirement of some characteristics in the room to detect the people, although when the people are detected the metrics obtained are very useful.

In the case of the radio-wave techniques a combination of Wi-Fi and Bluetooth waves techniques will be used, in this case the technique require some specialised hardware, not very different than the normal hardware to have that communication in the room, the main disadvantage is that for the people to be detected they need to carry a Wi-Fi or Bluetooth device.

The system will be divided into three parts:

- Vision: This part of the system is in charge of the detection and tracking of the people to obtain the different metrics that this technique can give, this part include hardware as well as software. The hardware needed are the video cameras, which need to have a good image quality and a transmission method for the video while it is being recorded,for the processing in real time it needs a computer with a GPU, supposing a 30 frames per second recording a Nvidia GT 1080Ti is enough. The software make use of machine learning for the detection of the different points, in order to improve the detection pre-processing is done, this pre-processing includes the hybrid improvements such as dividing the image received in different parts. To make it able to run in real time a tracking of the most important points is done to reduce the area of detection to reduce the processing time. All the metrics for a person will be related with a reference and sent to the main server.
- Radio-Waves: This part of the system is in charge of the detection of mobile devices to have the owner position. The hardware needed for this part is a server to store the position and MAC address of the device and process the measurements of the RSSI to calculate the position of the device and several raspberry pis, this devices will monitor the devices and send the RSSI measurements and MAC address to the server. The software is based on the Find3 code [67], also an own server has been made changing the way of processing the data for the position. All the metrics for a person's device will be related with a reference which will be both sent to the central server.
- Server: This part is the place where all the position data is obtained and computed, the hardware needed is a processing unit such a computer. The main objective of this part is to perform the relation of the vision and radio-wave systems, this will relate both of the references which has very similar positions. Another task of this part of the system is to communicate the changes to be made in the processing of both the systems.

4.2 Technology adaptation to the proposed case

In this section the implementation of the vision and radio-waves techniques are analysed, the analysis will be done explaining both the implementation and the results that the technique has. In the implementation it will be analyse the changes tweaks made to the technique as well as the reason for making that changes, including the problems that the out of the box technique has. In the results the testing of the technique alone will be done and the results of the different tests will be analysed.

4.2.1 Computer vision

One of the most used ways of locating people is the usage of cameras to record the room and locate in the image the people, as a human does the location, this location is done by computer vision using machine learning. The cameras used can be the ones used for security but they use to have low resolutions. This method uses the frames that are introduced in a machine learning framework for the detection.

The method chosen is detecting the people by the pose detection technique, where different joints are detected and with that the position of the person is calculated inside the room. This method has been used because the possibility of improving the project with other metrics obtained from this. The joints detected are eyes, ears, nose, shoulders, elbow, writs, hip, knees and feet, although for the different metrics only some of them are used. This method has been implemented to be used in different measurements:

- **Person detection**: This is the part of the method where the whole person is detected, this will retrieve the number of people and the position in the frame. For this detection only one point per person is needed, the same for all the person. It has been chosen to have the whole upper part and make the mean of all the points, this increase the detection as when one point is not detected others are used to compute the position.
- **Body part**: In this part one body part is located, this is done by locating all the points of that part and ignoring the rest, this can be done for face detection.
- **Several body parts**: In this part several body parts are located, this is only done to locate the arms or legs, as in the previous part, this is done by having all the points of that parts.
- **Face parts**: This is an important part for all the information that it can bring, this parts are the eyes, ears and nose. As in the previous parts it is done by having only the location of the interested points and ignoring the rest.
- **Attention detection**: This part uses the data from the previous part to perform the calculation of the direction the person is looking to, this can be used along the position in the room to pinpoint the exact thing the person is looking to.

The part of computer vision in this project can be divided into two different parts:

- **Machine learning**: As it has been explained in the state of the art machine learning is a very powerful method to allow a computer "think" and detect the joints, for that it has been used to make the comparison between the frames and the training used. This part is the main part for the joint detection and allow the computer to know the position of the joints. It is highly dependable from the hardware as, because of making use of a machine learning framework, is very computational hungry, running the system on a normal CPU (CPU name) it perform at a rate of 1.5 computed frames per second and with a Nvidia 1080 it performs at a rate of 15 frames per second.

- **Tracking**: This part is responsible of assigning a reference to a person detected to follow them across the room, while being detected, and reduce the computation by analysing less part of the figure, which will allow to process a higher number of frames with the same hardware. With this technique implemented an the processed frames can be double or even triple depending on the number of people to be processed

The location the system can be divided into:

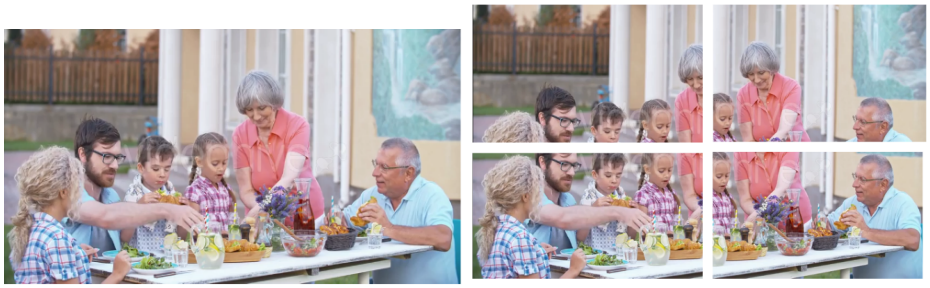
- **Processing**: The processing unit receive the images and uses a machine learning framework, preciously trained, to search for the joint points, this is done using a heatmap and then using only the points with a probability higher than a minimum, defined by the user. Then the points are filtered having only the important ones for the metric selected, being the rest ignored. At the end there is a list where it is stored the pixels for the joint in a predefined position.
- **Cameras**: This devices are only used to record the images of the room and transmit them to the processing unit. The cameras require to have good quality, at least 720p, and a method to transmit the images to the processing unit as they are being recorded. The cameras are recommended to have a manual mode and be selected, this is to have constant conditions in the images, if they were changing the detection will be even more difficult.

4.2.1.1 Implementation

The implementation of this technique is based on the posenet repository and the posenet-pytorch repository from Ross Wightman, which has been downloaded and adapted to the case, to adapt the code the main part for the detection has been maintained, erasing the non used parts to improve the performance. In order to simplify the implementation and because the working hours using the real-time video has been impossible, it has been chosen to use a pre-recoded video instead. The main problems with the implementation that has been solved are:

- **Room conditions** As a vision technique is being used, by using images of the room, some of the characteristics of it are very important and can highly determine the results, some of the characteristics that has this importance are the luminosity, contrast and resolution of the images recorded. In order to improve the detection of the joints some changes to the code has been introduced, this changes can be divided into three parts:
  - **Joints detected**: The use case is a show, because of that the people use to be standing and close one to another, this fact makes very difficult to see the legs and sometimes the hips of the people. Seing the images the most detected part are the head and the shoulders, the least one are the legs that are only detected on the people closer to the scenary. For this fact there has been programmed several modes of detection where it can be selected the parts to be looking for, this improves performance, this improvement is due to a minor number of searching and comparison on the frame.

– **Image partition**: The conditions of the room can differ between the different parts of the room, so in order to analyse the image it has to be partition in different parts in order to analyse each part separately. The act of make the division is done automatically with the only data given being the image and the number of rows and columns. In order to not lose information between the division some threshold has been introduced as it can be seen in the figure 4.1.



(a) Whole frame (b) Divided frame

Figure 4.1: Frame division

– **Preprocessing**: The preprocessing of the image is done on each division, in each one of them is done. The prepocessing can be done in three different ways:
  * **Brightness**: One of the image characteristics that can be changed is the luminosity of the image, this characteristic can affect to the detection as with low luminosity the people are more difficult to locate. In the case of changing the luminosity the frame is transformed into the YUV colour space, as the Y is the luma component, or the brightness, changing that component to the desired value. After the change the frame has to be change to the default colour space.
  * **Contrast**: This characteristic is the difference in luminance or colour that makes an object distinguishable. It is determined by the difference in the colour and brightness of the object the nearby objects. An adaptive histogram equalisation is chosen to perform an automatic contrast change, for that CLAHE(Contrast Limited Adaptive Histogram Equalisation) is used. This has been chosen as traditional Adaptive Histogram Equalisation amplify the noise in constant regions. The CLAHE method can be seen in the image 4.2, in this method a threshold is specified and the contrast above the threshold is shared evenly all along the image. This method allow better differentiation of objects as it can be seen in the image 4.3 where although it can appear to reduce the quality it can be seen the separation between the different objects easier than before.
  * **Gamma**: Gamma correction is a nonlinear operation used to encode and decode luminance or tristimulus values in video or still image systems. Gamma correction is, in the simplest cases, defined by the power-law expression ($V_{out} = AV_{in}^{\gamma}$). This correction is used to optimise the usage of bits when encoding an image, or bandwidth used to transport an image, by taking advantage of the non-linear manner in which humans perceive light and colour. In the figure 4.3 it can be seen that this step is done to improve the CLAHE performance.

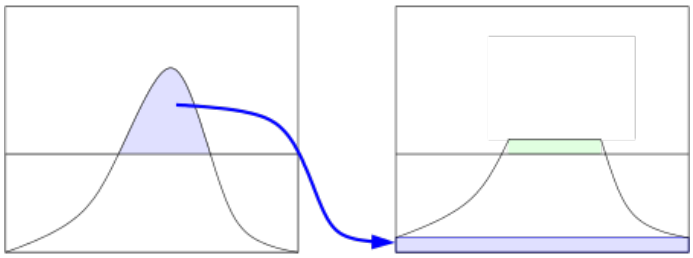


Figure 4.2: CLAHE behaviour

(a) Original partition

(b) Gamma corrected partition

(c) CLAHE applied partition

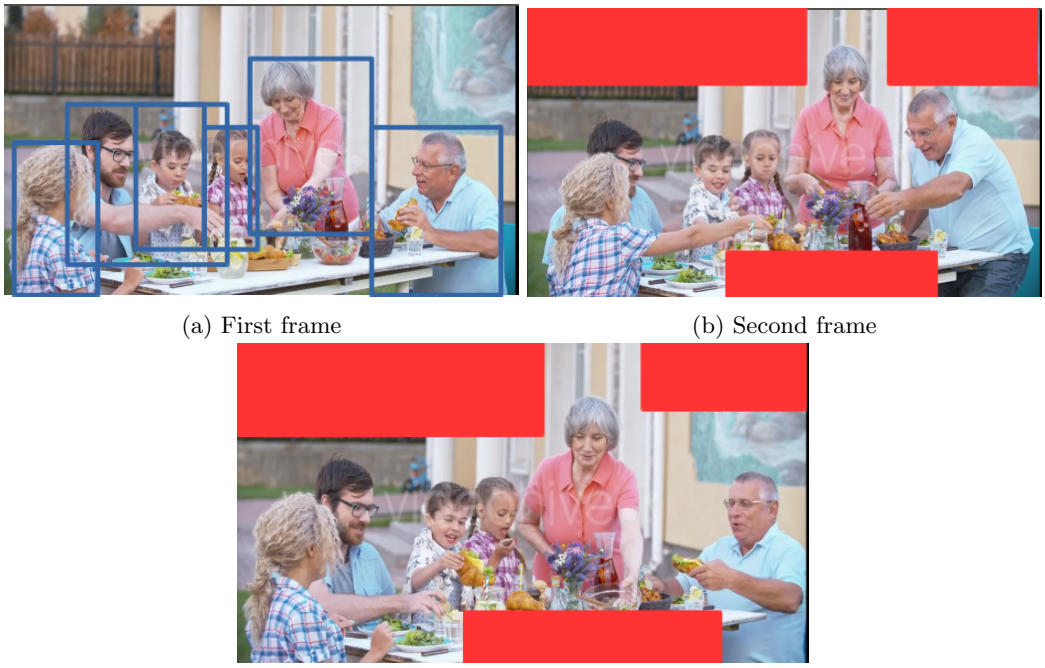
Figure 4.3: Preprocessing of a partition

- **Track people** In order to improve the performance and to be able to have information about the movement of the people a tracking functionality has been implemented. The tracking is considered to be an important part of the implementation as it both gives more functionality and improve the performance. In order to include this in the code there are in python, inside the OpenCV library, some function to been able to track an object, this is the multitracker function, which allow to track several objects at the same time.

  The improving on the performance is due to the change in the searching algorithm of the machine learning, which instead of searching in the whole frame, allow to search in smaller areas, which when there is a lower number of objects tracked highly improves the performance, as it does not look in the majority of the frame. In order to not lose new defections a whole search in the frame is done once each 10 frames.

The tracking can be done to follow a whole person thought the room in the time or can be used to track some parts of the person, this is allowed as what is really tracked is the part of the image inside a determined bounding box. this bounding box is what is given to the multitracker function. In order to follow the object it firstly compared with the tracked objects to see if they are already been tracked. In the case they are being tracked the position is updated, in the case they are not a new tracker is created.

In order been able to do the tracking a threshold is defined, this threshold is done to define the area around the bounding box where the machine learning framework search for the object. In the figure 4.4 it can be seen the tracking process.In the first frame it can be seen the people detected, which defines the bonding boxes for the tracking, in the rest of the frames it can be seen the parts where there is no searching



(a) First frame

(b) Second frame

(c) Third frame

Figure 4.4: Frames for tracking

- **Performance**

  One of the most important changes to be done are related to the performance, this is because, at it has been said before, using computer vision techniques with machine learning requires a huge computing capability, both including the training and the detection. In order to have a real time application is at least required to have 30 frames per second, or more depending on the camera. With a good processor as the i5-2300 CPU @ 2.80GHz it can run at 3.1 frames per second and with an Nvidia 1080 it can run at 32 frames per second. This performance has been reached with some changes in the code, which has been:

- **Minimum probability**: There is an option to change the minimum probability for the detection, this option has been selected for all the possible defections with the same variable. This option is very important for the performance and detection quality trade-off because it ignores the points with a probability lower than the minimum. It has been selected a variable minimum probability, which can be used for the detection improvement with the hybrid method.
- **Partition**: As it has been said before the input frame is divided in order to make different changes to the parts. In the case that there are zones where is impossible to detect a person in a part of the frame this part can be eliminated, this crop of the frame improves the performance as there are less pixels to analyse. An example of the image division was seen in the figure 4.1 and one example of the cropping can be seen in the figure 4.5.
- **Preprocessing**: As it has been explained before a preprocessing of each one of the division of the frame is done, with this preprocessing, although it requires some power it improves the performance as it make easier the detection with the changes in brightness, contrast and gamma correction.
- **Scaling**: As it was said in the partition part when analysing a lower quantity of pixels the process is speed up, but with a trade of the detection rate. In this case the input frame is reduced to speed the process.

(a) Original frame (b) Cropping process

Figure 4.5: Before and after cropping

- **Drawing** As it was said before to improve the performance by seeing the room conditions some of the joints can be ignored, and it can be selected the joints to detect. As the way of storing the positions is in a vector with the position, if detected, or a null value, if not detected, to improve the drawing several function has been done to speed up the drawing of the parts.

  In the original repository the only drawing function was to draw a whole skeleton, while in the case done it has been written other drawing functions, the output of this function can be seen in the figure 4.6.

(a) Original function

(b) Draw only keypoints function

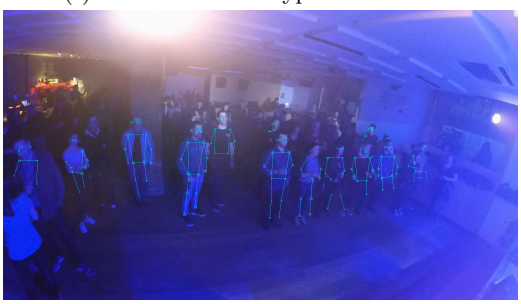
(c) Draw only skeleton function

(d) Draw face keypoints function

(e) Draw right arm keypoints function

(f) Draw left arm keypoints function

(g) Draw chest keypoints function



(h) Draw mean position of a part function

(i) Draw attention keypoints functions

Figure 4.6: Drawing functions

- **Draw only keypoints**: This function is done to eliminate the lines between the joints, process that is only done to prove that the detection of the points is correct, as the unions are pre-defined.
- **Draw only skeleton**: This function is done to draw only the connections and the face points to prove that the right and left parts are correctly detected.
- **Draw face keypoints**: As it has been said the face is one of the parts of a person that gives more information, and it is the easiest to see by the cameras, this function was written to show only the eyes, nose and ears.
- **Draw right arm keypoints**: This function has been written to analyse the movement of one of the arms.
- **Draw left arm keypoints**: This function has been written to analyse the movement of one of the arms.
- **Draw chest keypoints**: This function has been written to analyse the chest, this is because the chest is in the middle of the body and can give normalise measurements.
- **Draw mean position of a part**: This function draw a point with the mean position between the specified keypoints.
- **Draw attention keypoints**: This function has been written to draw the keypoints used in the attention analysis.

In the figure 4.6 it can be seen the difference on the drawing functions, although all of them can be used to have the number of people in each part of the room they bring different information with them. Some of them are only used to simplify the information to speed up the process, while other add new points.

It has been implemented a method to calculate the direction where the people are looking, this has been implemented using the position of the eyes and the nose. An algorithm is done instead of having a machine learning algorithm in order to simplify the process. This algorithm is based on the relative position of the nose with the eyes, analysing the relation of those parts it can be said the following:

- Drawing a line connecting both eyes and another between the nose and the half point between the eyes. If those lines are perpendicular it can be said that the person is looking to the front.
- Drawing a line connecting both eyes and another between the nose and the half point between the eyes. If those lines have an angle greater than 0.261799 radians, 15 degrees,it can be said that the person is looking to the right.
- Drawing a line connecting both eyes and another between the nose and the half point between the eyes. If those lines have an angle lower than -0.261799 radians, -15 degrees,it can be said that the person is looking to the right.

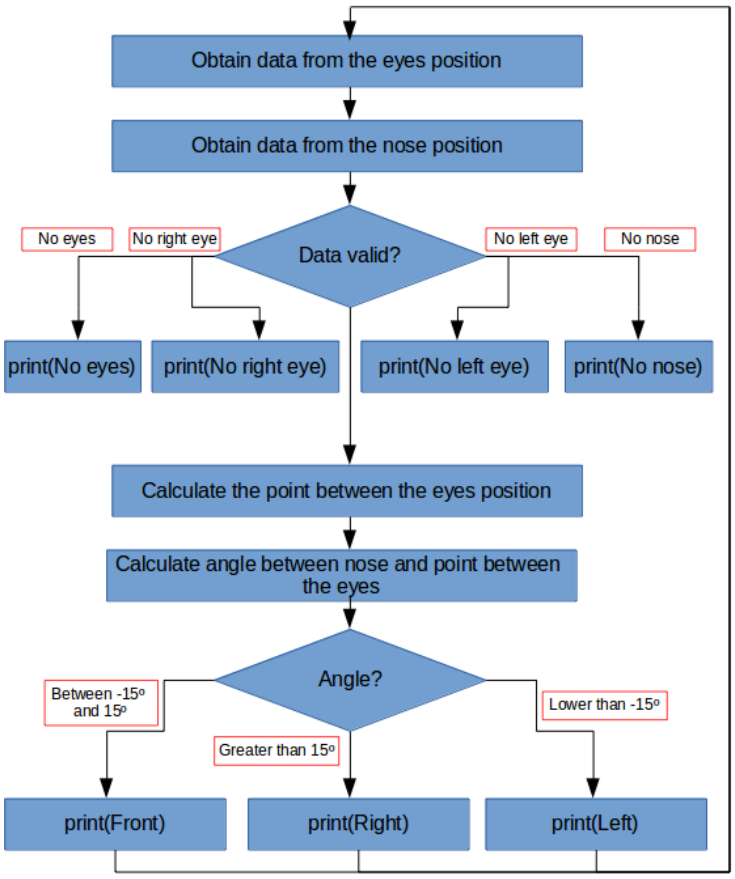


Figure 4.7: Work-flow for attention detection

The directions has been separated with 15 degrees in order to allow some deviations, in order to cope with some error that can happen in the detection. The algorithm to calculate the attention direction is based in the previous principles and can be seen in the work-flow of the figure 4.7, in that it can be seen first the information from the position of the eyes and nose needs to be in a list, then it is analysed if all that information is valid, in the case it is not valid a message will be shown, in the case all the data is valid the tangent definition will be used to calculate the angles between the lines. the formula is

$$angle = \arctan\left(\frac{Nose(Y) - Mid\_eyes(Y)}{Nose(X) - Mid\_eyes(X)}\right) \tag{4.1}$$

4.2.1.2 Testing

In the case of the computer vision the testing is done by applying the detection to several videos, this testing has been done with both the i5-2300 CPU @ 2.80GHz and the Nvidia 1080 to compare the performance. In order to see the degree in which the characteristics of the room affect the performance and detection rate. The tests done are five, each one of them has a different reason and will give different metrics.

- **Person detection**: In the person detection there are two main metrics to obtain, the first one is the number of people and the second one is the position of the detected person. This two metrics are important because with both of them it can done the same process as with the Wi-Fi, know the number of people inside a determine zones.

  For the person detection only one point is needed from a person, but is needed to know which point is it as the elbow and the nose are not in the same part of the body.

  - **Good conditions**: In the case of the first video there are six people, with proper illumination and contrast, this video has been taken with professional cameras. In this test it will be measured the maximum possible detection with perfect condition. As it can be seen in the figure 4.8 all the people are detected in the majority of the video. With that information it can be supposed that the detection is very good and can be performed in the majority of the situations. This test has been done to obtain the following metrics:

Figure 4.8: Person detection with good conditions

* Position of the people: This metric is based on obtaining the position of the person in the frame. This metric can be obtained in two modes, in the frame or in the room, in the first case the position is given by the pixels in the frame where the person is, being a relative position and dependent from the camera position, in the second case what is given is the absolute position in the room, this position is calculated with three factors, the position in the frame, the position of the camera and the size of the person.

* Number of people: This metric is very important as is one of the metrics that bring most of the information with less effort. This metric is obtained by adding 1 to a counter when a new person is detected in the frame. In the code it has been done with a list so by knowing the length of that list it can be know the total number of people in the room.

In this test it can be proved that the detection can be done, and with that the calculation of the metrics of the number and position of the people. In the figure 4.8 it can be seen that the people can be detected easier if they are looking in the direction of the camera. This is seen as the majority of the people are perfectly detected, in the case of the woman facing back the camera the detection is bad, but at least is detected. For the metrics to be calculated, both the number of people and an approximate position of the joints and so on the whole person.

– **Bad conditions**: In the case of the second video there are more than forty people, in this case with a similar illumination, contrast and conditions with the proposed case, this video has been taken with sports cameras. In this test it will be measured as the one with good conditions the maximum possible detection but with the case conditions. As it can be seen in the figure 4.9 between 10 and 20 people are detected. With that information it can be supposed that the detection is quite bad without the proper preprocessing. This test has been done to obtain the following metrics:

* Position of the people: This metric is based on obtaining the position of the person in the frame. This metric can be obtained in two modes, in the frame or in the room, in the first case the position is given by the pixels in the frame where the person is, being a relative position and dependent from the camera position, in the second case what is given is the absolute position in the room, this position is calculated with three factors, the position in the frame, the position of the camera and the size of the person.

* Number of people: This metric is very important as is one of the metrics that bring most of the information with less effort. This metric is obtained by adding 1 to a counter when a new person is detected in the frame. In the code it has been done with a list so by knowing the length of that list it can be know the total number of people in the room.

Figure 4.9: Person detection with bad conditions

In this test it can be proved that the detection can be done, and with that the calculation of the metrics of the number and position of the people. It can also be seen that a preprocessing is needed and that the conditions are very important for the detection.

In the figure 4.9 it can be seen that the closer the person is the detection is better, and also that the people which are illuminated with the spotlight are also detected, this means that the brightness of the image needs a tuning, this tuning is very difficult as the spotlights are dynamic so the tuning has to be done in real time, for that the preprocessing in the divided image is done to be able to change the characteristics of the parts of the image. In this case it has been seen that the detection is low in number of people, but when the person is detected all the visible parts are detected.

- **Face detection**:In the face detection the metrics to obtain are very similar to the one in the person detection, number of people and position of the detected person. In this case the metrics are more reliable as the area is smaller and because the variability of the head size being smaller than the variability of the whole person size.

  For the face detection only one point is needed from the face, and in this case the knowledge of which point is it as the distance of the face points are not very high.

  - **Good conditions**: In the case of the first video there are six people, with proper illumination and contrast, this video has been taken with professional cameras. In this test it will be measured the maximum possible detection with perfect condition and compare that detection with the person detection. As it can be seen in the figure 4.10 6 people are detected the majority of the time. With that information it can be supposed that the detection is very good and can be performed in the majority of the situations. This test has been done to obtain the following metrics:

Figure 4.10: Face detection with good conditions

    * Position of the people: This metric is based on obtaining the position of the person in the frame. This metric can be obtained in two modes, in the frame or in the room, in the first case the position is given by the pixels in the frame where the person is, being a relative position and dependent from the camera position, in the second case what is given is the absolute position in the room, this position is calculated with three factors, the position in the frame, the position of the camera and the size of the person.

* Number of people: This metric is very important as is one of the metrics that bring most of the information with less effort. This metric is obtained by adding 1 to a counter when a new person is detected in the frame. In the code it has been done with a list so by knowing the length of that list it can be know the total number of people in the room.

In this test it can be proved that the detection can be done, and with that the calculation of the metrics of the number and position of the people. In the figure 4.10 it can be seen that compared to the person detection the detection is very similar, but there is a requirement the person must be slightly facing the camera, in other case the person is not detected. In this case the majority of the people are detected when their face is being seen, including the ears. For the metrics to be calculated, both the number of people and an approximate position of the joints and so on the whole person.

– **Bad conditions**: In the case of the second video there are more than forty people, in this case with a similar illumination, contrast and conditions with the proposed case, this video has been taken with sports cameras. In this test it will be measured as the one with good conditions the maximum possible detection but with the case conditions. As it can be seen in the figure 4.11 between 8 to 20 people are detected. With that information it can be supposed that the detection is quite bad without the proper preprocessing. This test has been done to obtain the following metrics:

Figure 4.11: Face detection with bad conditions

* Position of the people: This metric is based on obtaining the position of the person in the frame. This metric can be obtained in two modes, in the frame or in the room, in the first case the position is given by the pixels in the frame where the person is, being a relative position and dependent from the camera position, in the second case what is given is the absolute position in the room, this position is calculated with three factors, the position in the frame, the position of the camera and the size of the person.
* Number of people: This metric is very important as is one of the metrics that bring most of the information with less effort. This metric is obtained by adding 1 to a counter when a new person is detected in the frame. In the code it has been done with a list so by knowing the length of that list it can be know the total number of people in the room.

In this test it can be proved that the detection can be done, and with that the calculation of the metrics of the number and position of the people. It can also be seen that a preprocessing is needed and that the conditions are very important for the detection.

In the figure 4.11 it can be seen that compared to the person detection the detection is very similar and that the closer the person is the detection is better, also that the people which are illuminated with the spotlight are also detected, this means that the brightness of the image needs a tuning, this tuning is very difficult as the spotlights are dynamic so the tuning has to be done in real time, for that the preprocessing in the divided image is done to be able to change the characteristics of the parts of the image. In this case it has been seen that the detection is low in number of people, but when the person is detected all the visible parts are detected.

Taking into account both of the conditions in the cases of the face and person detection it can be said that both methods are very similar in performance when facing the camera, and a bit worse when not facing it. For that because of the similar performance and the benefit of the lower area the face detection method is preferred than the person detection one.

- **Chest detection**:In the chest detection the metrics to obtain are the same as in the two previous methods. For the chest detection only one point is needed from the chest, but is needed to know which point is it as the lower and higher part are separated.

  - **Good conditions**: In the case of the first video there are six people, with proper illumination and contrast, this video has been taken with professional cameras. In this test it will be measured the maximum possible detection with perfect condition. As it can be seen in the figure 4.12 6 people are detected. With that information it can be supposed that the detection is very bad, mostly due to the low quantity of chests on line of sight. This test has been done to obtain the following metrics:

Figure 4.12: Chest detection with good conditions

* Position of the people: This metric is based on obtaining the position of the person in the frame. This metric can be obtained in two modes, in the frame or in the room, in the first case the position is given by the pixels in the frame where the person is, being a relative position and dependent from the camera position, in the second case what is given is the absolute position in the room, this position is calculated with three factors, the position in the frame, the position of the camera and the size of the person.
* Number of people: This metric is very important as is one of the metrics that bring most of the information with less effort. This metric is obtained by adding 1 to a counter when a new person is detected in the frame. In the code it has been done with a list so by knowing the length of that list it can be know the total number of people in the room.

In this test it can be proved that the detection can be done, but need line of sight with the thing to be detected. In the figure 4.12 it can be seen that the people can be detected easier if they are looking in the direction of the camera and have at least the upper part of the chest. In this case the detection of at least one point is performed in all the person, the main drawback is the lose of information as in the majority of the frames only the upper part, shoulders, of the chest are detected. In the figure the detected points can be drawn in two different ways, a small point or a big one, the difference is based on the probability of the detection, being a bigger circle when having a higher probability.

– **Bad conditions**: In the case of the second video there are more than forty people, in this case with a similar illumination, contrast and conditions with the proposed case, this video has been taken with sports cameras. In this test it will be measured as the one with good conditions the maximum possible detection but with the case conditions. As it can be seen in the figure 4.13 between 12 an 20 people are detected. With that information it can be supposed that the detection is quite bad without the proper preprocessing. This test has been done to obtain the following metrics:

Figure 4.13: Chest detection with bad conditions

∗ Position of the people: This metric is based on obtaining the position of the person. This metric can be obtained in two modes, in the frame or in the room, in the first case position is given by the pixels in the frame where the person is, being a relative position and dependent from the camera position, in the second case what is given is the absolute position in the room, this position is calculated with three factors, the position in the frame, the position of the camera and the size of the person.

* Number of people: This metric is very important as is one of the metrics that bring most of the information with less effort. This metric is obtained by adding 1 to a counter when a new person is detected in the frame. In the code it has been done with a list so by knowing the length of that list it can be know the total number of people in the room.

In this test it can be proved that the detection can be done when at least is seen half of the chest, and with that the calculation of the metrics of the number and position of the people. It can also be seen that a preprocessing is needed and that the conditions are very important for the detection.

In the figure 4.13 it can be seen that compared to the face detection the detection is worse, needing more part of the body to be seen, and that the closer the person is the detection is better, also that the people which are illuminated with the spotlight are also detected, this means that the brightness of the image needs a tuning, this tuning is very difficult as the spotlights are dynamic so the tuning has to be done in real time, for that the preprocessing in the divided image is done to be able to change the characteristics of the parts of the image. In this case the people on the front rows are easily detected with all the points of the chest, but in the case of the other rows only the upper points are detected.

Taking into account both of the conditions in the cases of the face and chest detection it can be said that the first method is way better so the chest method will not be used.

- **Movement tracking**:In the movement tracking there one metric to be detected, the movement of the person, this mean knowing the quantity of movement of each person and the position of the person at each time. In this case a detection method, one from the previous ones, is needed and apart from that a log of the positions are needed for each person.

  - **Good conditions**: In the case of the first video there are six people, with proper illumination and contrast, this video has been taken with professional cameras. In this test it will be measured the maximum possible detection with perfect condition. As it can be seen in the figure 4.14 6 people are detected and tracked. With that information it can be supposed that the tracking is very good and can be performed in the majority of the situations. This test has been done to obtain quantity of movement, which is based on obtaining the position of the person at each movement, making a relation between the time of the measurement, the position and a identifier.

    In this test it can be proved that the tracking can be done, and with that the calculation of the movement of the different people. In the figure 4.14 it can be seen that the people can be tracked easier when they are detected, in the case the person is no longer detected the next time it is taken as a new person. In this case it can be seen the path that each person has been followed, the detection method chosen has been the face detection. The movement tracking is not only based on having the path of the last positions, the total and maximum movement for each person has been computed, and that information is displayed each 5 seconds.

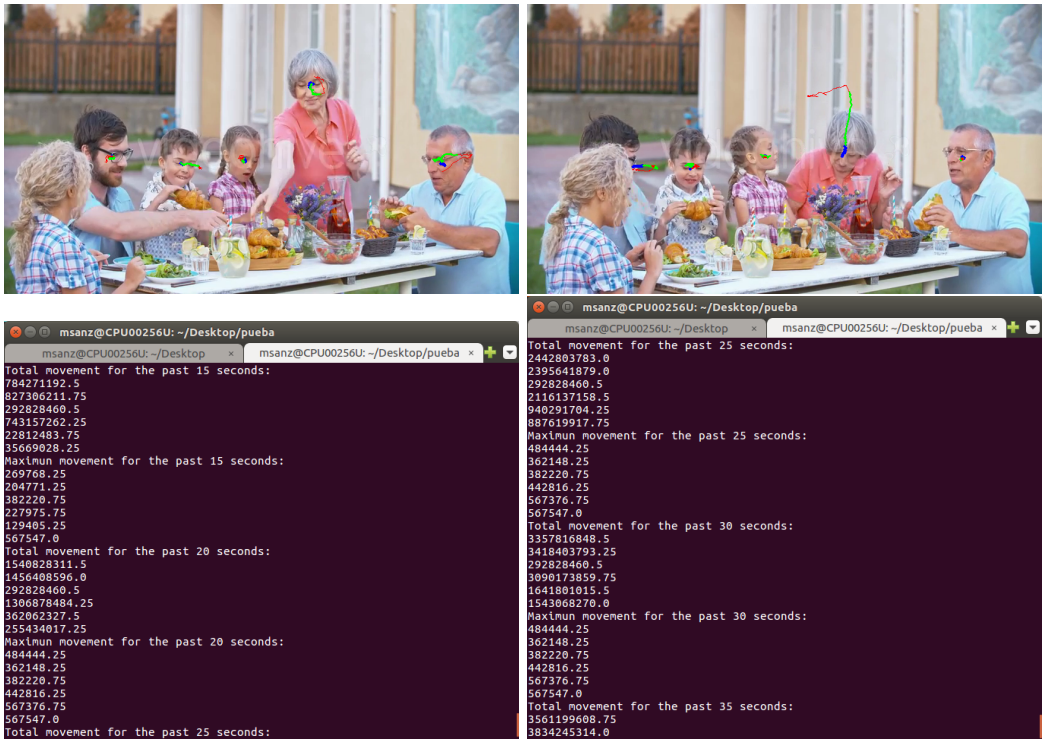


Figure 4.14: Movement tracking with good conditions

– **Bad conditions**: In the case of the first video there are six people, with proper illumination and contrast, this video has been taken with professional cameras. In this test it will be measured the maximum possible detection with perfect condition. As it can be seen in the figure 4.15 between 15 and 20 people are detected and tracked. With that information it can be supposed that the tracking is very good and can be performed in the majority of the situations. This test has been done to obtain quantity of movement, which is based on obtaining the position of the person at each movement, making a relation between the time of the measurement, the position and a identifier.

  In this test it can be proved that the detection can be done, and with that the calculation of the movement of the people. It can also be seen that a preprocessing is needed and that the conditions are very important for the detection. In the figure 4.15 it can be seen that the tracking is very dependable from the detection and that when a person is no longer detected the next time it is taken as a new person. In this case it can be seen the path that each person has been followed, although the movement is very low, the detection method chosen has been the face detection. The movement tracking is not only based on having the path of the last positions, the total and maximum movement for each person has been computed, and that information is displayed each 5 seconds.

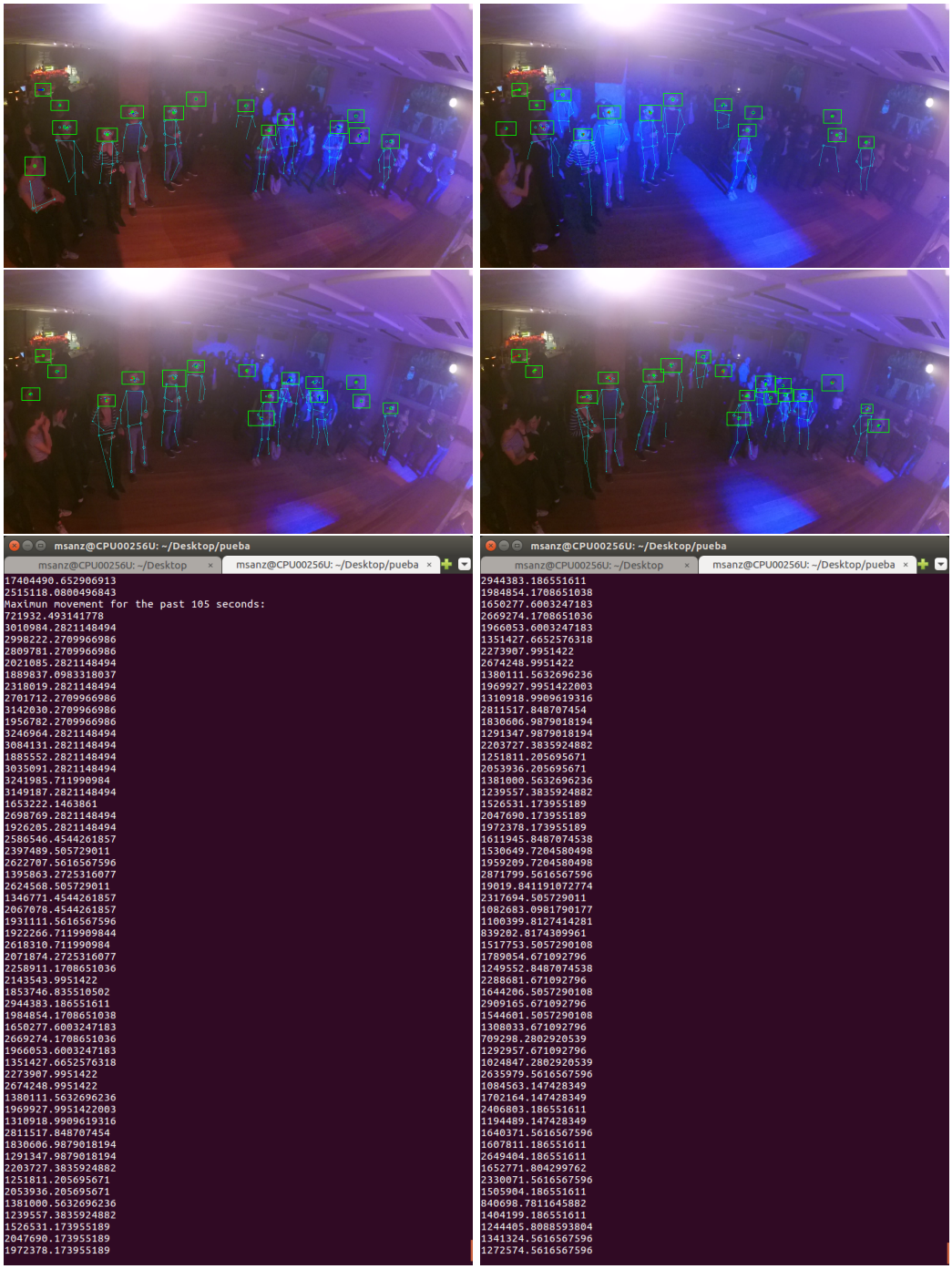


Figure 4.15: Movement tracking with bad conditions

Taking into account both of the conditions in the cases of the tracking it can be said that it is dependable to the method of the detection as the main principle is to store the position and a time-stamp of the detection.

- **Attention detection**: In the attention detection some parts of the faces, such as eyes and nose, are used as it has been previously explained to calculate the direction where the people are looking.
  - **Good conditions**: In the case of the first video there are six people, with proper illumination and contrast, this video has been taken with professional cameras. In this test it will be measured the direction of the detection with perfect condition. As it can be seen in the figure 4.16 6 people are detected and the direction of attention is correct in approximate 40% of the cases.

Figure 4.16: Attention detection with good conditions

1 recto	3179 izquierda	3201 izquierda
2 izquierda	3180 recto	3202 recto
3 recto	3181 recto	3203 derecha
4 recto	3182 derecha	3204 izquierda
5 recto	3183 derecha	3205 derecha
6	3184 recto	3206 recto
7 recto	3185	3207
8 izquierda	3186 izquierda	3208 izquierda
9 derecha	3187 recto	3209 recto
10 recto	3188 recto	3210 derecha
11 recto	3189 recto	3211 recto
	3190 recto	3212 derecha
	3191 izquierda	3213 recto

Figure 4.17: Attention detection data with good conditions

In this test it can be proved that the detection can be done, and with that the calculation of the position of the face parts the direction of the attention can be known although it is improved by weighting the measurement with the past measurements. In this case in the image the points used for the attention detection are drawn, as well as a file is written with the direction where the the attention is given.

– **Bad conditions**: In the case of the first video there are more than forty people, in this case with a similar illumination, contrast and conditions with the proposed case, this video has been taken with sports cameras. In this test it will be measured the direction of the detection with perfect condition. As it can be seen in the figure 4.18 between 5 to 15 people are detected and the direction of attention is correct in 30% of the cases, with the possibility of being improved with weighted methods.

Figure 4.18: Attention detection with bad conditions

In this test it can be proved that the detection can be done but as with the tracking method is very dependable on the detection method, and with that the calculation of the position of the face parts the direction of the attention can be known although it is improved by weighting the measurement with the past measurements. In this case in the image the points used for the attention detection are drawn, as well as a file is written with the direction where the the attention is given.

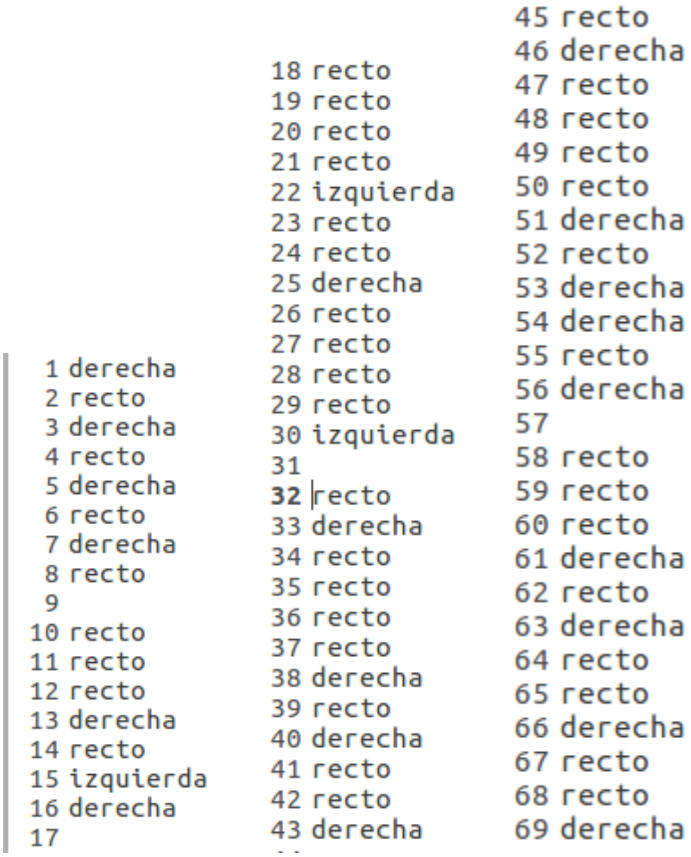


Figure 4.19: Attention detection data with bad conditions

4.2.1.3 Conclusions for computer vision

The conclusions for using the computer vision method alone are:

- With a low number of cameras with direct line of sight with the people it's number can be approximated.
- The condition of the room are very important, as it was seen in the comparison of the testing in the videos, having better conditions with professional videos.

- A preprocessing can heavily improve the detection. This preprocessing is the change of the brightness, contrast and a gamma correction. The contrast correction is done with the CLAHE method.
- The attention detection works better when the people are closer but it can be used at a medium distance.
- The movement detection works well when the detection is done correctly, with the only drawback of the refreshing of the information when the person is not detected in a frame. To eliminate old measurements the vector with the movement can be erased, or when the person is no longer detected the movement vector position is erased.
- The detection can be done on different parts of the body, the best detection has been obtained with the face or the face plus the shoulders.

4.2.2 Radio-waves (Wi-Fi + Bluetooth)

One way of locating people is to use devices that they carry with them, one of those devices are the mobile phones, those devices have wireless capabilities which can be used to obtain the position as it was explained in previous chapters. The method implemented to calculate the position of the device is based on measuring the RSSI (Received Signal Strength Indication), and comparing it with a database of locations related with the RSSI.

RSSI is an indication of the power level being received by the receive radio after the antenna and possible cable loss, so when this number is higher means that the signal received has more power, the method to calculate the RSSI of the signal from different devices is based on the connection probes. Once each couple of minutes when a device has the Wi-Fi turned on , in order to connect to the known networks, they broadcast some important information:

- **Mac address**: As it will be explained more in detail later there are some methods to protect this information from being known, but because of the way that the Wi-Fi probes was created a MAC address has to be transmitted.
- **Network name history**: Each device openly broadcasts incredibly identifiable network name history, such as "Miguel's Oneplus 6" or "Home".

Although the broadcast of all this information is a very huge concern, that broadcast is a transmission, because of that the power of the signal can be measure and related to a MAC address. This characteristic of the Wi-Fi connection method is the one used to perform the localisation of the devices. As this method is based on the Wi-Fi probes in order to receive the information and compute the RSSI it does not require the devices to be connected to the same network, although it has some advantages in precision and information, the only requirement for the device is to either have the Wi-Fi or the Bluetooth connection active.

In order to locate the devices the system can be divided into two parts:

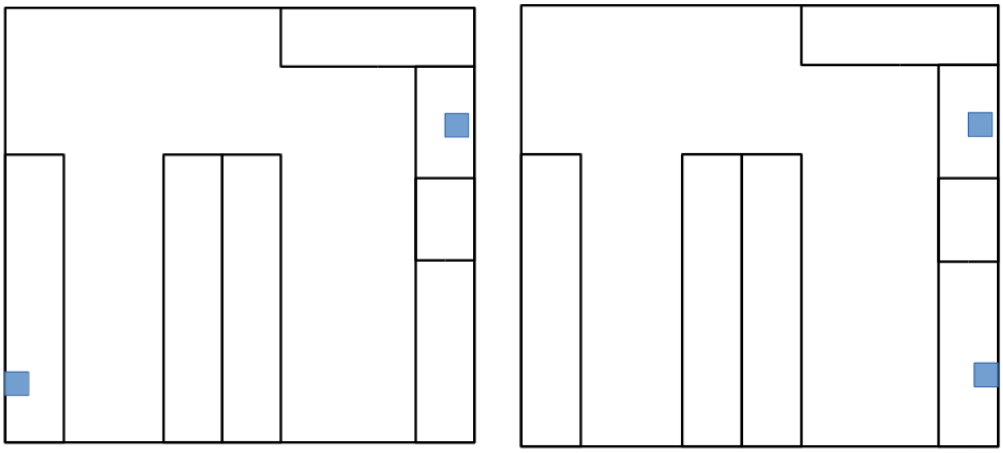
- **Server**: This part is in charge of storing a database of all the RSSIs received, MAC address and the tracking device who obtained the measurements. This server has two main tables, and other complementary ones, the main tables are devices, which contains the mac of the device detected and an identifier as it can be seen in table 4.1, and the table sensors, which contains the time-stamp of the measurement, the identifier, the location if it's in training and wifi values, being this last field created by putting together the identifier of the tracking device and the RSSI measured, this can be seen in the table 4.2. This server is created in a PC running the Ubuntu Linux distribution, and it is based on the server of the Find3 repository.
- **Tracking devices**: This type of devices are the ones monitoring the Wi-Fi network, the devices used are Raspberry Pis, one 2B and one 3B, with TP-Link TL-WN722N Wireless dongle, a Wi-Fi dongle with support for promiscuous mode. In the testing this type of devices have been placed as it can be seen in the figure 4.20.

Identifier	MAC Address
b	wifi-f4:60:e2:e5:3c:ce
c	wifi-da:a1:19:e7:2f:87

Table 4.1: Devices table

Timestamp	Identifier	Location	Wi-Fi text
1555418541129	r	desk	”a”:-62,”b”:-40
1555418541140	c	desk	”b”:-74’

Table 4.2: Devices table



(a) Tracking devices placement a (b) Tracking devices placement b

Figure 4.20: Tracking devices placement

The room that can be seen in the figure 4.20 has an approximate dimensions of 9 by 7 metres, it has several tables with computers, and a rack in the middle of the right side, the tables have similar separation between them. The first placement of the tacking devices has been chosen to maximise the separation of the tracking device, in order to cover the maximum area of detection and have the most different RSSI possible between the tracking devices. In the second placement of the tracking devices the idea has been to prove if the position of the tracking devices can affect to the precision of the localisation.

4.2.2.1 Implementation

The implementation of this technique is based on the Find3 repository, from which it was downloaded and adapted to the used case, to adapt the server the non Wi-Fi related processes has been erased such as the IA part. The tracking devices code has been based on the Find3-cli-scanner with the proper adaptation to the used case, this adaptation is similar to the one done in the server, eliminating the non used process. As the tracking devices work by sending probe signals to every Wi-Fi capable device the database in the server filled up very fast, this problem has been related to:

- **MAC Randomisation**: The MAC address is an unique identifier for each device that can be connected to a net, either Wi-Fi, Bluetooth or any other, the main problem was that this identifier was send when a probe request was sent in some nets as Wi-Fi, which result in lack of privacy. To solve that the MAC randomisation was developed, this randomisation has been implemented in mobile phones from Android 8 in Android devices and IOS 8 for Apple devices. The randomisation of the devices only happens when the device is not connected to the network and the change is done for every request, increasing the number of devices in every scan.
- **Different Wi-Fi capable devices**: Mobile phones are the main devices to track for the used case, this means that other devices as computers, laptops, routers or wireless storage devices are not desired devices to be tracked. This is more noticeable in the laboratory where the testing has been done in Vicomtech, where because of being an investigation centre situated in a technological park it has a higher density of wifi capable devices, this devices are not only the expected ones, such as mobile phones, laptops or routers, they can be prototypes or development boards that are not expected to be found in the normal use case.

The mac assignation to the different devices is done by buying blocks of addresses to the IEEE, with 16777216 mac addresses in each block, in their website there is a list containing the type of assignation (normally a whole block MA-L), the block assigned, organisation name and organisation address, as it can be seen in the table 4.3. In order to attenuate the problem of devices different than mobile phones the IEEE list can be used to eliminate the address from organisations that does not manufacture mobile phones, this problem is not solve as the MAC address can be from any device of the organisation. The MAC randomisation makes impossible to track the movement of the devices in the room, but allow to know the number of devices and the part of the room where the device is in a determined moment. In order to solve the problem on the MAC behaviour the

main code adaptation has been to implement a MAC filter that only allow the devices from known mobile phone manufacturers.

Registry	Assignment	Organization Name	Organization Address
MA-L	807ABF	HTC Corporation	No. 23, Xinghua Rd., Taoyuan City Taoyuan County Taiwan TW 330
MA-L	E005C5	TP-LINK TECHNOLOGIES	Building 7, Second Part,CO.,LTD Honghualing Industrial Zone Shenzhen Guangdong CN 518000

Table 4.3: Mac assignation table

As the MAC randomisation is done by software the method is different in Android and IOS devices, in the case of the Android ones the macs transmitted are inside the blocks own by Google, so by including the Google blocks in the filter the Android devices are detected. In the case of IOS devices the MAC address is randomised in the range of all the possible MAC addresses, although Apple is not the owner of them, this last part of the MAC randomisation problem has not been solve because the complexity of the situation compared with the percentage of IOS users in the used case.

Without taking into account the minor changes to make it work and the changes mentioned before to improve the performance there were three mayor changes implemented to improve the performance, add new functions and solve partially the previous problems mentioned:

- **Mac differentiation**: This part implements an algorithm to compare the macs detected with the a list of MACs accepted. As the list of MAC blocks is huge and it has no proper order an algorithm has been done to have all the blocks from the mobile phone manufacturers. The reduction on the possible MAC address is noticeable going from 26160 MAC address blocks to 2607, a reduction of more than 90%. The next part of the mac differentiation is being able to develop an algorithm to compare the mac blocks and mac address detected. In this part the algorithm has been develop in four steps:
  - **Mac vendor relation in server in python**: in this first step the algorithm was implemented in python, a language used to develop fast codes but not very resource efficient, in order to know if the reduction on the number of MAC address detected affect to the performance of the server. This algorithm is placed after the download of the data from the database to treat the data, being a previous step to the zone differentiation. This part also implements an algorithm to obtain the variables from the MAC list, having all the macs are listed in a variable and all the blocks from the mobile phone vendors in another, two nested for loops are used to make the comparison. In order to select only the important information from the information downloaded from the database a regular expression is used to search for the proper words to obtain the MAC address and the RSSI of both devices.

- **Mac vendor relation in scanner in python**: in this step the algorithm is implemented in python in order to know if there is a proper reduction in the number of macs detected by the device and if the performance of the tracking devices is improved, it can be seen in the table 4.4 that the required time is higher than ten seconds, time between the scans, this means that in order to be used the performance needs to be improved. Another problem with this part is the place to put the code on the Golang scanner.
- **Mac vendor relation in scanner in Golang**: as the previous step works well but it needs to perform the tasks between each scan the code has been implemented in Golang, language centred on the efficiency, in order to reduce the computation time of the algorithm. Another problem solve with the usage of Golang is the place to put the code in the scanner by writing between the scan and the creation of the database packets. As it can be seen in the table 4.4 in this case the performance is suitable for the use case reducing the time necessary by a factor of 11.33.
- **Mac vendor relation in Server in Golang**: with the improvement seen in the previous step it has been chosen to put the differentiation in the server written in Golang to speed up the computation and reduce the number of files. As it can be seen in the table 4.4 the main advantage of using Golang is the improvement on the time as it reduce the time by a factor of 4.555.

Server with python	Server with Golang	Scanner with python	Scanner with Golang
0.16454	0.03668	28.36	2.5

Table 4.4: Time(s) spent in the mac comparison

- **Data treatment**: this part is centred on downloading and give format the data in the database to make an easier the work with it.in order to perform this operation automatically it has been divided into two parts.
  - **Data download**: in order to communicate with the server the simplest way is by shell commands, the commands indicate the part of the database that will be transmitted as well as the maximum time for the operation, the parts of the database that are interesting in the used case are:
    * Locations: it includes the MAC address, time-stamp and Wi-Fi measurements, those three data are the base of the algorithm.
    * Devices: it includes only the devices MAC address in the database, without any other information.
    * Database: it include all the information of the server, this is only taken as a backup in case the other files are not transmitted.

    The commands has a configurable timeout, it has been chosen as 300 seconds, in the case that the information is not given on that time it will mean that the database has many devices and needs to be cleaned. This timeout has been calculated when the scanning has been running for weeks without any MAC filter.The work-flow of the downloading process can be seen on the figure 4.21.

– **Data formatting**: the data downloaded has only some important information, the rest can be discard, to do that a formatting is needed, this process will look for the important information and put it in a known format to work with it.

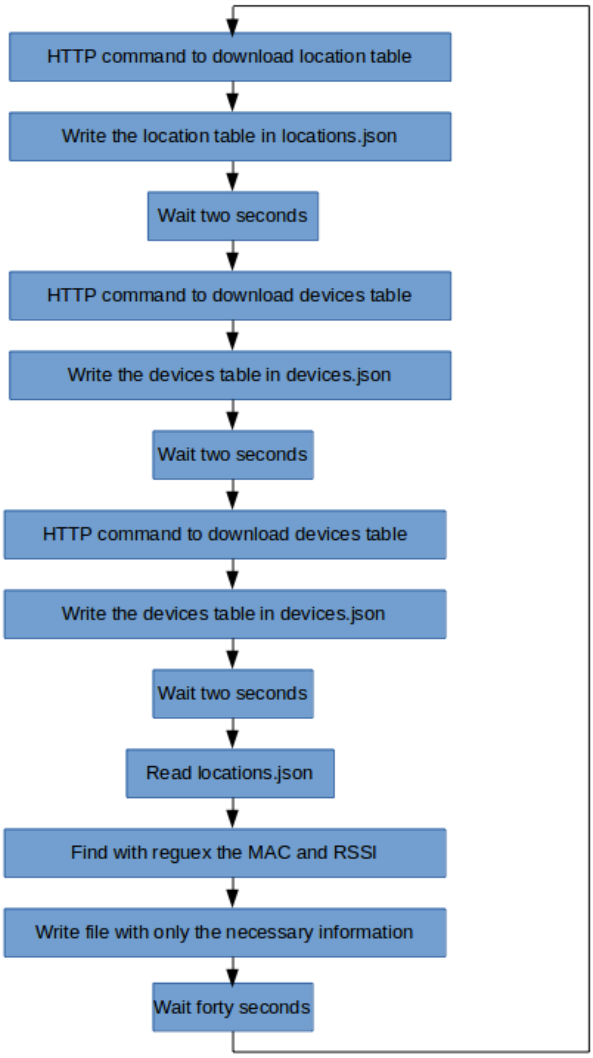


Figure 4.21: Work-flow for the data treatment

- **Zone differentiation**: the main objective of this part of the project is to been able to know the zone where the mobile devices are placed, for that the program will need to be able to know the Wi-Fi RSSI of the different zones inside the room. The process of differentiating the zones in the room can be done in three ways:

- **Manual**: this method is based on having the room RSSI mapped, and compare the measurements of each device with that to position each of the devices, it has low computation requirements. The main problem with this approach is related with the wireless signal characteristics, which change with the environment, mostly due to obstacles and reflections, for that reason the precision of this method is not very high.
- **Manual with weighted values**: this method tries to improve the precision by using a group of measurements instead of only one, it has been chosen to use the mean of the last ten values. The increase on precision can be seen in the table 4.12 but it also increase the time to perform the computation.
- **Automatic with reference devices and weighted values**: the main focus of this method is to make the zone differentiation non dependable on the conditions of the room, this is done by using reference devices in certain points of the room to obtain the measurements of known positions and compare them with the non known devices. The known devices can be distinguish by their mac address as long as they stay connected to the net allowing a direct identification using the mac differentiation code previously explained. As it can be seen in the table 4.12 it requires more computation but it's precision outperforms the other by a huge margin

The table 4.12 has been done with the computation of 1008 values from a hundred different devices taken for 12 minutes, in the weighted methods it has been taken the last ten measurements to perform a mean of the values, and in the last method three devices where used as a reference.

4.2.2.2 Testing

In order to test this solution the used case it has been emulated in a laboratory, this laboratory is the one seen in the figure 4.20, it has been taken this lab because of the open areas to work and also because it has a constant flow of people and device which can easily emulate the use case. The testing is done with two raspberry pis as scanner devices, which will be placed in two positions across the room as it can be seen in the figure 4.20, The tracked devices will be three Android devices:

1. **Motorola mobile phone**: This is the model Moto G 2º generation running on Android 6.
2. **Samsung galaxy tablet**: This is the model Galaxy Tab Pro 8 running on Android 5.
3. **Samsung galaxy tablet**: This is the model Galaxy Tab Pro 8 running on Android 4.

The test were develop to measure the RSSI in different zones of the room as well as the RSSI while moving the devices. This tests will be done by running the scanner in loop with sleeps of 10 seconds and downloading the data when there has been four uploads, this is done to have stable data and minimise the number of peaks appearing. The devices will be with the screen turned on to been able to take data of the device each 20 seconds instead of the 120 seconds that will be necessary when the phone is blocked. The test taken were:

1. **Android devices placed in the different positions with the scanning devices in position 1**:

   In this first test the Android devices were placed in several places across the room, each device were measured in each one of the positions shown in the scenario. The order of the devices in each position can be seen in the table 4.5, each of the measurements is given by the download of at least ten RSSIs.

	Motorola	Tablet 1(MAC 7E)	Tablet 2(Mac 78)
First measure	x	y	z
Second measure	y	w	x
Third measure	w	z	y
Fourth measure	z	x	w

Table 4.5: Time(s) spent in the mac comparison

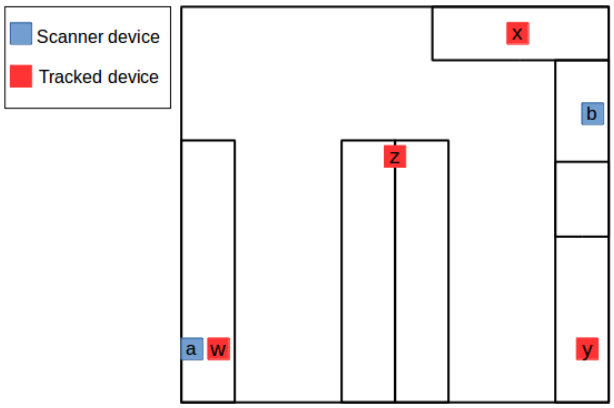


Figure 4.22: Test 1 scenario

The positions of the table 4.12 has been chosen for being accessible and cover main points of the room, those positions are placed at different distances from the scanner devices, being, the w position at a distance lower than 40cm to the scanner a, position z at a distance of 2.5 metres from scanner b, position x at a distance from scanner b of 1.5 metres and the y position at 8.5 metres from scanner a and 5 metres from scanner b with a rack in the middle.

Information about the measurements can be seen in the tables of the appendix, the information shown is the mean, variance and maximum difference. In general it can be seen that the mean change depending on the part of the room it is placed as it was supposed, from this data it can be seen that the mean of the different devices in the same place is very similar but not the same, having general deviations due to the change of the characteristics of the room, it also can be seen that the variance and maximum difference changes a lot, which can also be due to the change of the characteristics of the room.

Looking closely to the tables A.1 and A.2 it can be seen that when the device is place near the scanning device has a higher RSSI as it was supposed, but in some measures the difference between the data is so high that some weights should be added, as it was done in the second zone differentiation method. With the information from the test it can be said that with two scanning devices enough precision can be obtained to divide the room into at least 3 zones. This behaviour is sum up by positions in the table 4.6, where it can be seen clearly the difference between the regions, increasing the number of scanning devices will also increase the number of zones that could be distinguish, the number of zones will depend on the area of them as well as the position of the scanning devices.

Raspberry Pi 2B	w	z	x	y
Mean	-38.42	-51.22	-55.045	-47.66
Variance	2.72	13.86	9.31	11.60
Maximum difference	4.0	11.0	11.0	13.0
Raspberry Pi 3B	w	z	x	y
Mean	-65.45	-63.33	-38.65	-59.33
Variance	4.67	15.82	23.95	16.0
Maximum difference	8.0	15.0	13.0	13.0

Table 4.6: Data from Test 1

In the table 4.6 can be seen the measurements of each one of the scanner devices for the whole zone whereas when the measure was taken and it can be seen that the difference is higher than for each device, this means that each one of the device has different sensitivity to the Wi-Fi signal.

2. **Android devices placed in the same position with the scanning devices in position 1**:

In this tests, as it can be seen in the figure 4.23 all the devices are placed in the same part of the room at the same time, this is done to measure if the deviation of the data from the test 1 is because the device sensitivity or the characteristics of the room. The positions of the figure 4.23 has been chosen for being accessible and because they cover main points of the room, those positions also are placed at different distances from the scanner devices, being the first position at a distance lower than 40cm to the scanner a for all the devices, second position at a distance from scanner b of 1.5 metres and the last position at 8.5 metres from scanner a and 5 metres from scanner b with a rack in the middle.

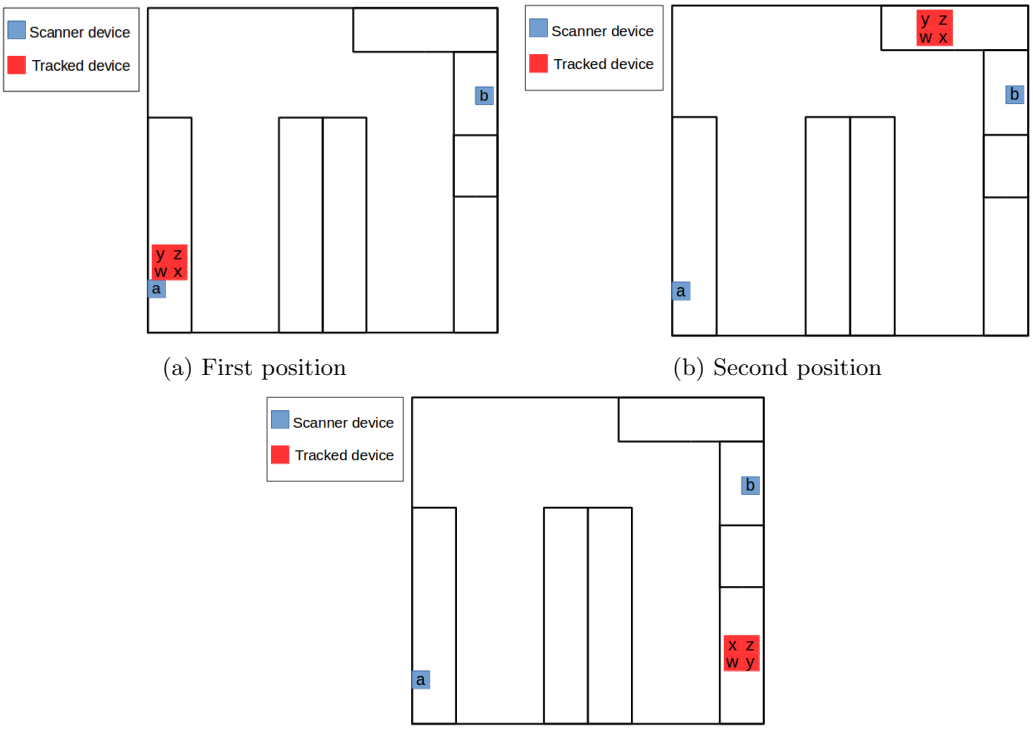


Figure 4.23: Test 2 scenario

As it can be seen in the tables A.3 and A.4 that the measurements were a bit different although they were in similar positions, so it can be said that the difference previously seen in the devices were not only caused by the room characteristics, it was caused in part by the device used, and as there are two devices which are the same model it can be supposed that the device characteristics such a usage patterns, battery and age of the components also affects to the RSSI.

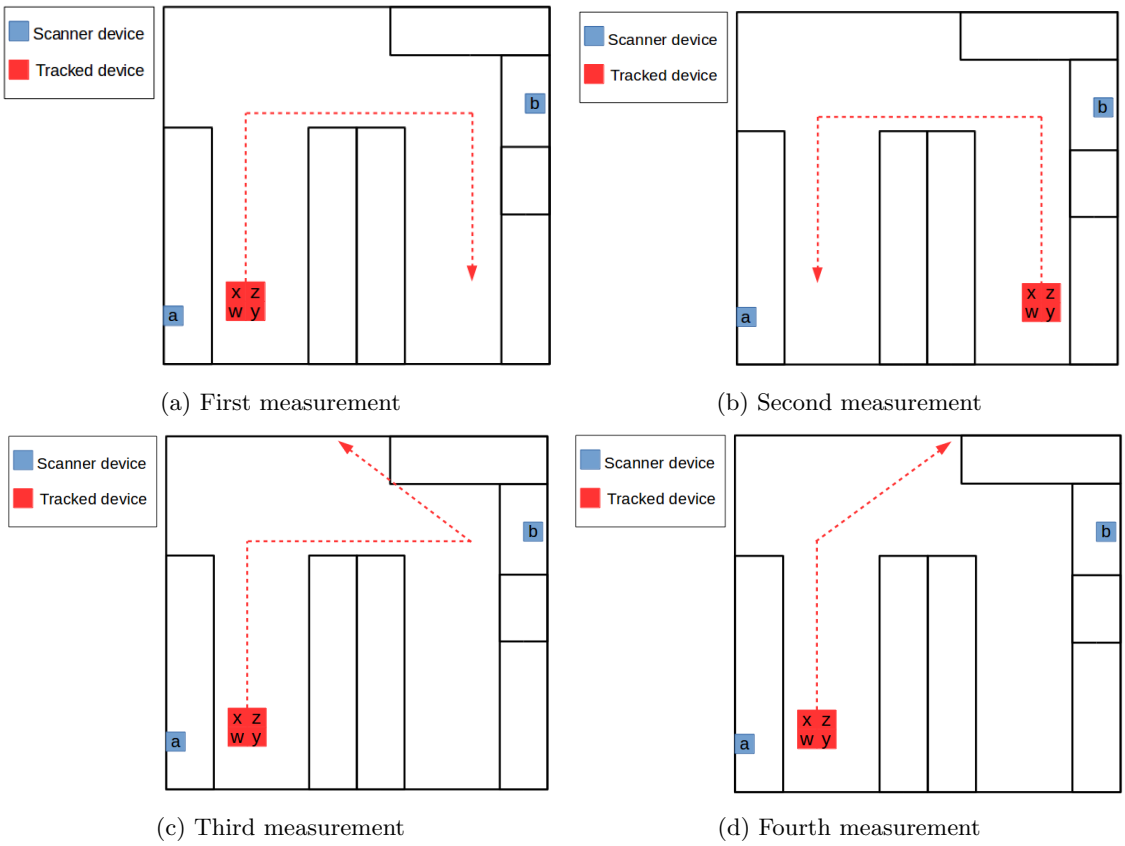
Raspberry Pi 2B	w	z	x
Mean	-51.88	-38.75	-57.09
Variance	5.165	9.51	17.66
Maximum difference	12.0	9.0	14.0

Raspberry Pi 3B	w	z	x
Mean	-61.22	-65.18	-44.57
Variance	10.28	6.876	29.88
Maximum difference	10.0	11.0	23

Table 4.7: Data from Test 2

In the case of the table 4.7 it can be seen that the different zones can be easily differentiate using the weighted method on the last RSSIs of the devices, this shows the same results as in the previous test, the room can be easily divided in at least 3 zones with only two scanning devices with the possibility of increasing the number of zones by increasing the number of scanning devices.

3. **Android devices moving together with the scanning devices in position 1**:



(a) First measurement

(b) Second measurement

(c) Third measurement

(d) Fourth measurement

Figure 4.24: Test 3 scenario

In this test the precision of the localisation of a moving device will be analysed, for that all the devices will be put together and moved in four paths to emulate different movements of the devices, the paths chosen are the ones shown in the figure 4.24. The change to a dynamic scenario is is a big change but can indicate the time for a device to be detected as being in another position. The paths were chosen to maximise the distance moved and to cover the following changes:

- **Direction of movement**: This is the change between the paths of the first and second measurements, if there is a difference between both it could be seen if when the MAC address is known the direction of movement can be known or the past measurements need to be stored.

- **Exact path vs relative path**: This is the change between the paths of the third and fourth measurements, if there is a change in the measurements it will indicate a good precision in the detection while the device is moving, allowing the detection of different paths that can be related with the engagement of the people.

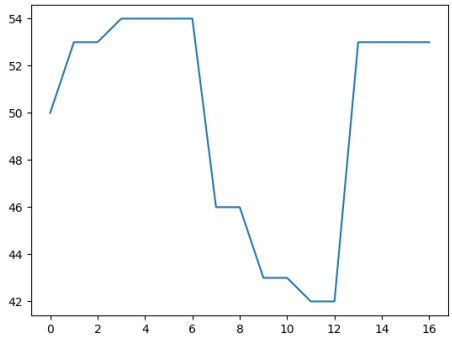
- **Percentage of line of sight**: This will cover if a change on the percentage of direct line of sight affect drastically to the signal, this happens in last movements of the third and fourth measurements, where a wall is placed covering the majority of the line of sight in last measure.

In the tables A.5 and A.6 it can be seen that, as expected, the measurement have very similar means from one path to the other and to the mean of the measurements of the room, the variance and maximum difference are very high, which indicates a movement, sadly the quantity of movement can not be seen with that data having the necessity of using the raw data to determine the place where the device is in each moment, the different graphs of the raw data can be seen in the figures 4.25 to 4.28, from this data the following information can be extracted:

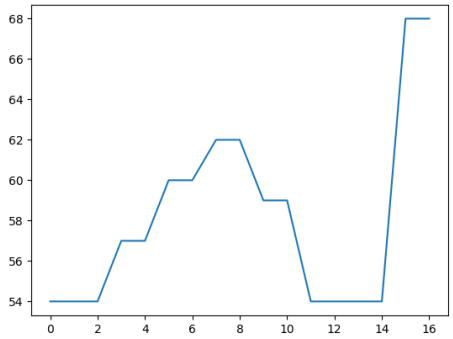
- The general behaviour is the same from the device Motorola to the device Samsung tablet when both are measured by the same scanning device. This could be used to perform the detection of the movement.

- The exact behaviour from the device Motorola to the device Samsung tablet when both are measured by the same scanning device suffer small changes, this make that the detection of movement can change from one device to another having some deviation and reducing the precision.

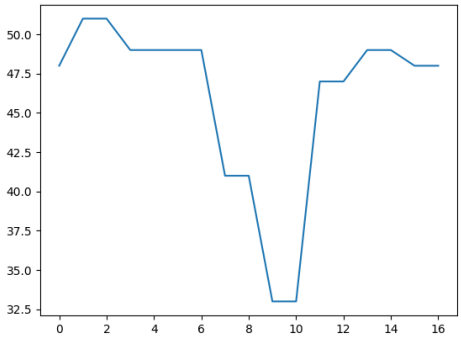
This indicates that the movement detection has to be done using other algorithm than the weighted zone detection, as this method performs a mean of the last values. The data from the figures 4.25 and 4.26 indicates that the direction of movement can be distinguish as the graphs are different between them. The data from the figures 4.27 and 4.28 indicates that there is a difference between exact path and the relative one as there is a change in the behaviour of the measurements, the increase of the values in the last part of the figures 4.27 and 4.28 indicates that the decrease on the line of sight is directly related with the change on the RSSI.



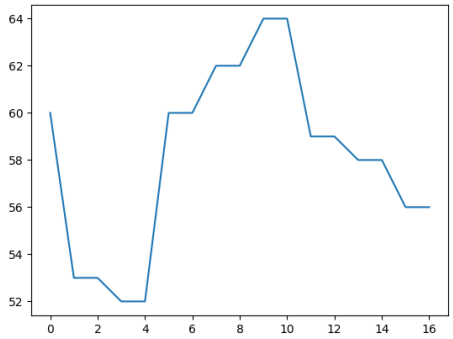
(a) Motorola RSSI A



(b) Motorola RSSI B

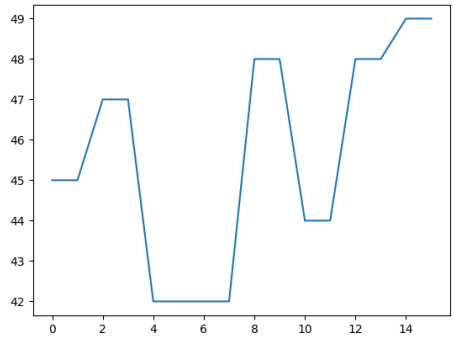


(c) Samsung tablet RSSI A

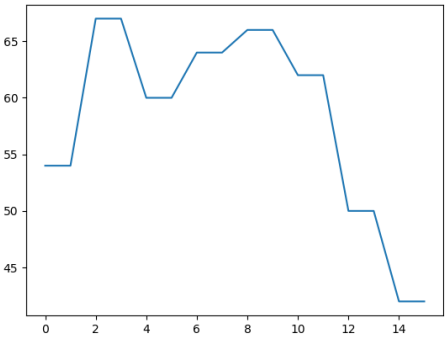


(d) Samsung tablet RSSI A

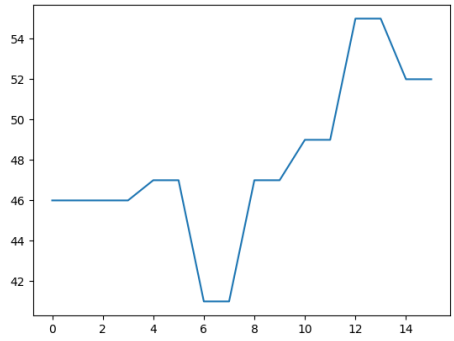
Figure 4.25: Test 3 measurement for path 1



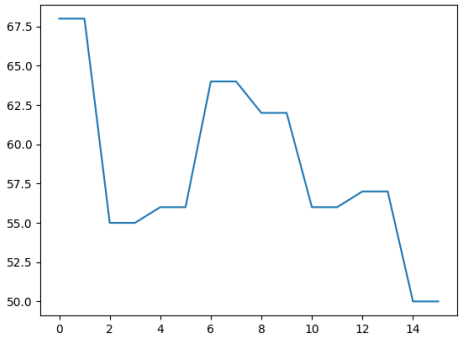
(a) Motorola RSSI A



(b) Motorola RSSI B

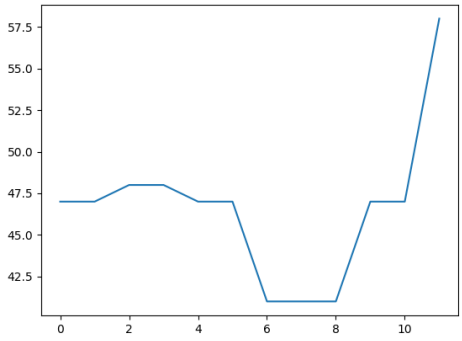


(c) Samsung tablet RSSI A

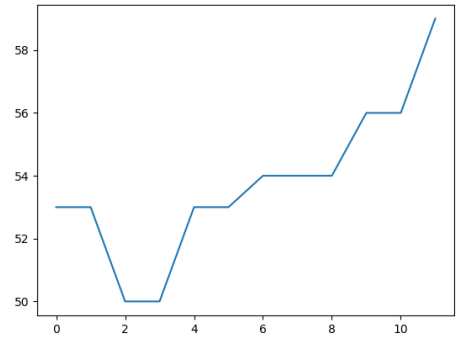


(d) Samsung tablet RSSI B

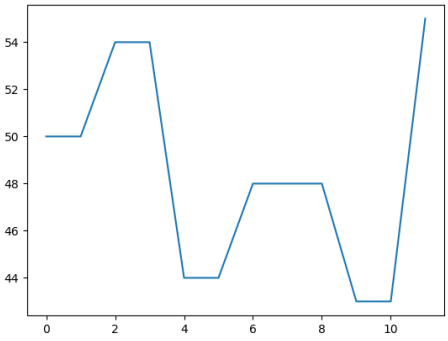
Figure 4.26: Test 3 measurement for path 2



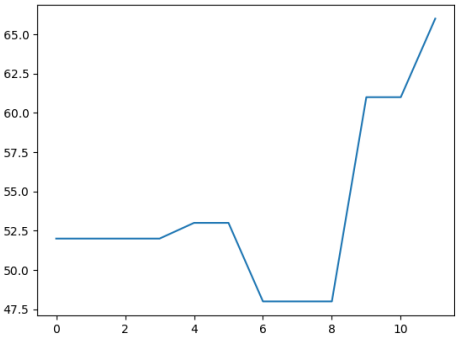
(a) Motorola RSSI A



(b) Motorola RSSI B

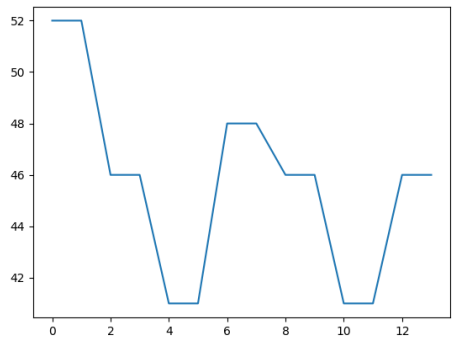


(c) Samsung tablet RSSI A

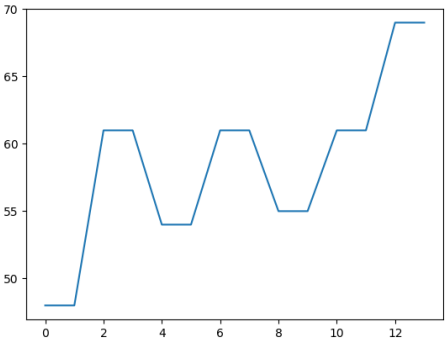


(d) Samsung tablet RSSI A

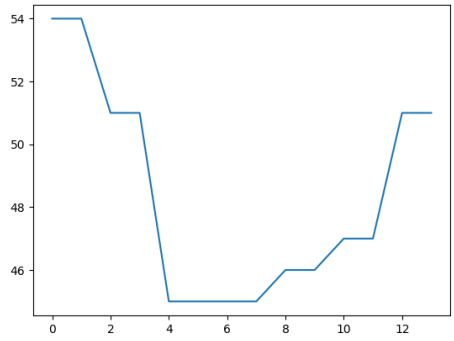
Figure 4.27: Test 3 measurement for path 3



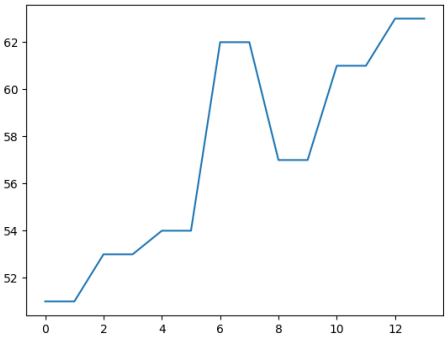
(a) Motorola RSSI A



(b) Motorola RSSI B



(c) Samsung tablet RSSI A



(d) Samsung tablet RSSI A

Figure 4.28: Test 3 measurement for path 4

4. **Android devices placed in the same position covered by a human body and with the scanning devices in position 1**:

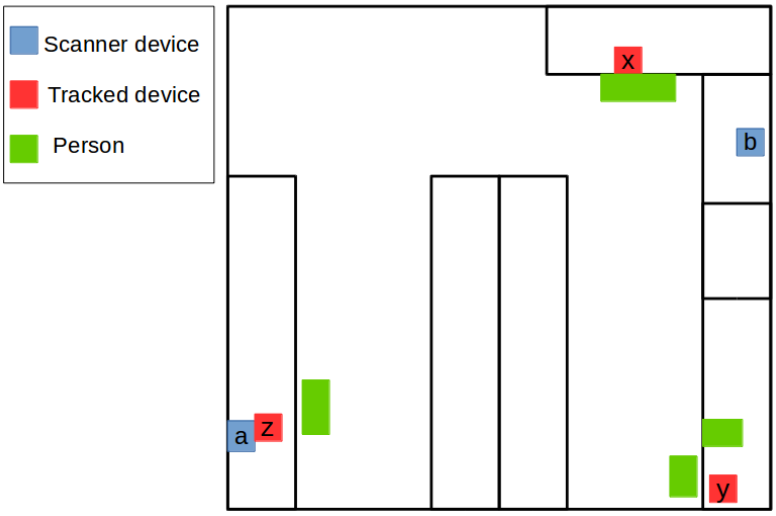


Figure 4.29: Test 4 scenario

In this test the signal will be measured for each place of the room in all the devices in two cases, covering one of the scanner devices and without covering the scanning device, this is done in order to measure the losses in the RSSI that are produced when the Wi-Fi waves go trough a person. This test has a huge importance in the current used case, engagement in a live event, as in that case will be many people moving being the difference on the measurements important as it can change the zone detected in methods with previous mapping.

The measurements has been taken with and without covering the device in the positions x, y and z of the figure 4.29, covering in the case of the position x the strongest signal, in the case of the position y the signal of one of the scanning devices, and then the other at a different time, which will be taken as y and y', and the position z in which the lower signal is the one covered.

As it can be seen in the table A.7 that the act of covering one of the scanning devices affect to the tracked devices, this change affect to both scanning devices not only the covered one, which is explained by the reflection that a water mass, as the human body, does to the Wi-Fi waves. It can be seen that the measurements while covered has a lower variability, by observing the data of the variance and maximum difference, although this can be because of the constant change of the characteristics of the room.

In the case of the table A.8, it can be seen a similar performance as the one described for the table A.7, changing the measurement although not being covered, and the variability of the data in this case is not as maintained as before, which would indicate that the previous supposition of the lower variability was only due to the characteristics of the room while the measurements were done.

In the general case divided by places, covered in the table 4.8, it can be seen the same as analysing the measurements device by device, both measurements are affected although the scanning device and the tracked one are on line of sight one from the other.

Raspberry Pi 2B	y	y'	z	x	y(cover)	y'(cover)	z(cover)	x(cover)
Mean	-45.33	-52.13	-55.64	-51.0	-47.53	-52.4	-54.88	-57.91
Variance	18.88	4.78	22.65	9.27	32.14	3.43	15.32	17.49
Maximum difference	9.0	6.0	13.0	9.0	17.0	5.0	12.0	12.0
Raspberry Pi 2B	y	y'	z	x	y(cover)	y'(cover)	z(cover)	x (cover)
Mean	-58.38	-62.46	-61.86	-44.95	-67.13	-68.46	-61.44	-43.33
Variance	5.23	16.24	7.97	1.41	12.24	1.71	30.47	12.63
Maximum difference	9.0	11.0	9.0	4.0	8.0	4.0	15.0	12.0

Table 4.8: Data from Test 2

5. **Android devices placed in the same position covered by a human hands as it will be being used and with the scanning devices in position 1**:

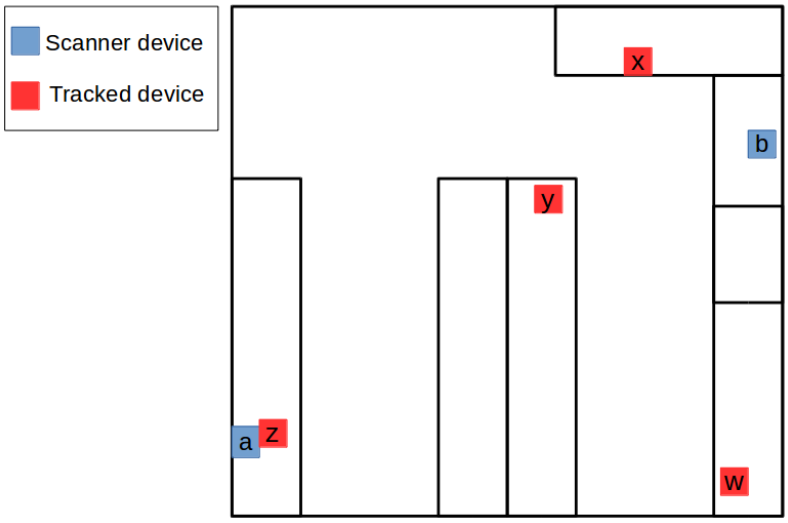


Figure 4.30: Test 5 scenario

In this test the signal will be measured for each place of the room in all the devices in two cases, one person using the tracked device normally, including the covering of some parts with the hands to hold it, this is done in order to measure the losses in the RSSI that happens when the Wi-Fi waves go trough a person while using thee phone.

The measurements has been taken with and without covering the device in the positions of the table 4.5, in the test the signal will be measured for each place of the room in all the devices in two cases, one person covering half of the devices as they were being used and then the rest, this is done in order to measure the losses in the RSSI that happens when the Wi-Fi waves go trough the hands. This test has a huge importance in the current use case as many people use their phones at some time during the live events to perform actions such as recording the live event or communicate with people with their phones.

As it can be seen in the tables A.9 and A.10 the act of using one of the scanning devices affect to the tracked devices, as the person is holding the device with the hand around most of the device, covering the antennas, which reduce the signal strength of the signal. The same happens when analysing the other scanning device, as it can be seen in the tables A.11 and A.12. In the general case divided by places, seen in the table 4.9, it can be seen the same as analysing the measurements device by device, both measurements are affected in a similar way.

Raspberry Pi 2B	y	z	y(cover)	z(cover)
Mean	-53.65	-50.14	-61.4	-62.29
Variance	12.30	11.17	25.44	307.76
Maximum difference	13.0	10.0	17.0	44.0
Raspberry Pi 2B	y	z	y(cover)	z(cover)
Mean	-56.34	-43.61	-61.4	-48.96
Variance	23.07	34.71	12.773	28.48
Maximum difference	16.0	16.0	13.0	15.0

Table 4.9: Data from Test 2

6. **Android devices placed in the different positions with the scanning devices in position 2**:

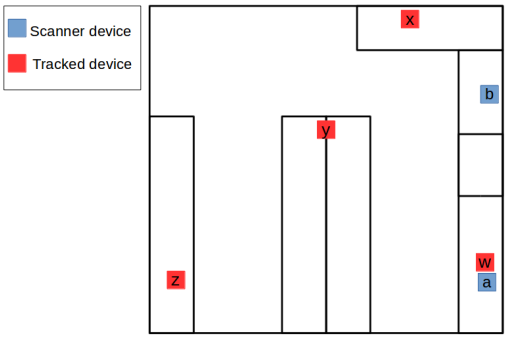


Figure 4.31: Test 6 scenario

In this test the Android devices were placed in several places across the room as in the case of the test 1, with the main difference of the position of the scanning devices. The order of the devices in each position can be seen in the table 4.5, each of the measurements is given by the download of at least ten RSSIs. The difference with the first one is the movement of one of the scanning devices, this movement is done to probe the difference on the precision with different placement of the scanning devices, in this case the position of the scanning device a is not optimal as both scanners are on the same side of the room and closer. Because of the change on the position the results are expected to be worse than in the test one, having lower difference between the points in the measurements.

Looking closely to the table A.13 and A.14 it can be seen that when the device is place near the scanning device has a higher RSSI as it was supposed, but in some measures the difference between the data is so high that some weights should be added, as it was done in the second zone differentiation method. With the information from the test it can be said that with two scanning devices there is a good precision to divide the room, but the division is more difficult than in the test 1. This behaviour is sum up by positions in the table 4.10, where it can be seen the difference between the regions not as clear as in the first test.

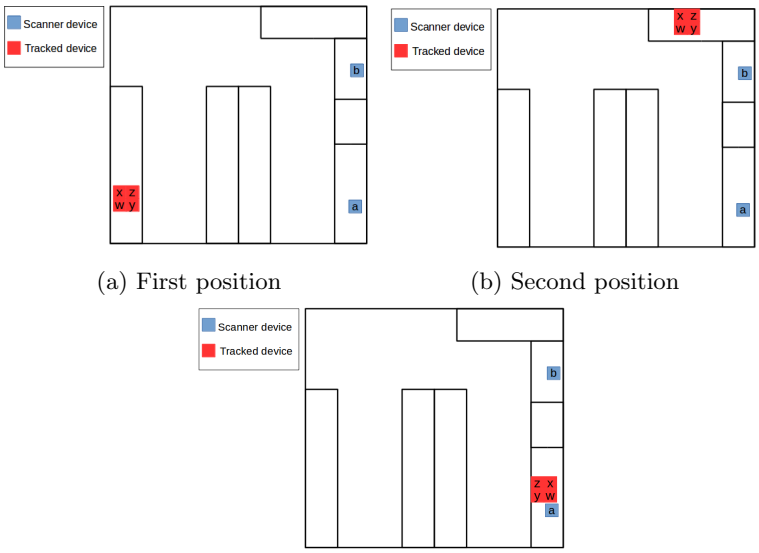
Comparing the tables 4.10 and 4.6 it can be seen the measurements are similar, but the deviation, both in variance and maximum difference, is higher this prove the difference on the precision due to the position of the scanning devices not being the optimal.

Raspberry Pi 2B	w	z	x	y
Mean	-39.44	-53.34	-51.37	-45.97
Variance	21.95	11.25	48.00	10.61
Maximum difference	14.0	10.0	19.0	9.0
Raspberry Pi 3B	w	z	x	y
Mean	-52.47	-30.45	-59.14	-50.09
Variance	2.66	375.96	12.23	14.67
Maximum difference	6.0	63.0	11.0	15.0

Table 4.10: Data from Test 1

7. **Android devices placed in the same position with the scanning devices in position 2**:

In this test, as it can be seen in the figure 4.32 all the devices are placed in the same part of the room at the same time, as in the case of the test 2, with same difference on the placement of the scanning devices as the one between tests 1 and 6.



(a) First position (b) Second position

(c) Third position

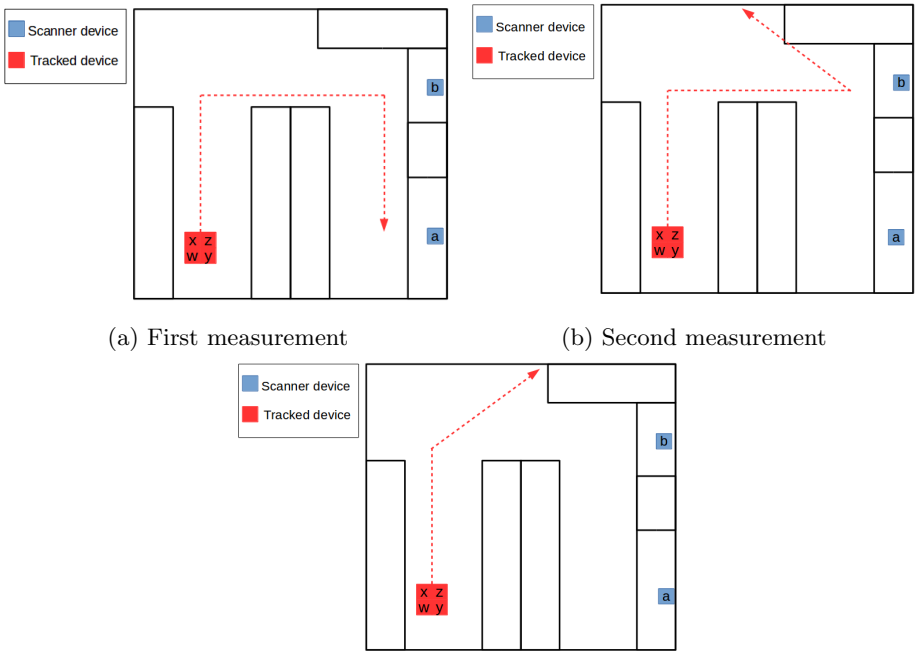
Figure 4.32: Test 7 scenario

As it can be seen in the tables A.15 and A.16 although all the devices were placed in the same part of the room the measurements were a bit different, so it can be said that the difference previously seen in the devices were not only caused by the room characteristics, it was caused in part by the device used, these are the same results as in the comparison between test 1 and 2. In the case of the table 4.11 it can be seen that the different zones can be differentiate using the mean of the last RSSIs of the devices using both scanning devices but not as accurate as with the optimal placement of the scanning devices, this is the same result as in the previous test.

Raspberry Pi 2B	w	z	x
Mean	-58.0	-35.52	-54.96
Variance	21.51	6.60	32.69
Maximum difference	15.0	7.0	15.0
Raspberry Pi 3B	w	z	x
Mean	-59.45	-55.97	-33.51
Variance	27.65	32.67	67.7
Maximum difference	20.0	19.0	23.0

Table 4.11: Data from Test 2

8. **Android devices moving together with the scanning devices in position 2**:



(a) First measurement (b) Second measurement

(c) Third measurement

Figure 4.33: Test 8 scenario

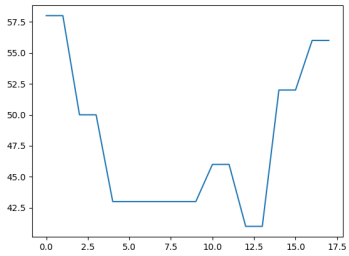
In this test the precision of the localisation of a moving device will be analysed, as in test 3, the paths chosen are the ones shown in the figure 4.33. This test will measure the behaviour of the measurements when a device is being moved. The paths were chosen to maximise the distance moved and to cover the following changes:

- **Exact path vs relative path**: a change in the measurements will indicate a good precision in the detection while the device is moving, allowing the detection of different paths that can be related with the engagement of the people.
- **Percentage of line of sight**: This will cover if the change on the percentage of direct line of sight affect drastically to the signal, this is cover in last movements of the second and third measurements, where a wall is placed covering the majority of the line of sight in last measure.

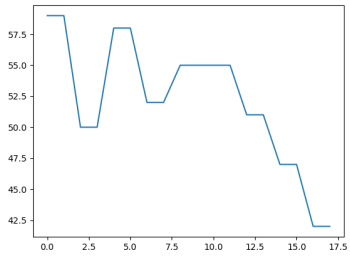
In the tables A.17 and A.18 can be seen that the measurement have very similar means from one path to the other and to the mean of the measurements of the room, and that the variance and maximum difference are very high, as in the case of the test 3, but as in previous comparissons with a lower precision. The different graphs of the raw data can be seen in the figures 4.34 to 4.36, in this figures it can be seen the exact raw data of the devices, from this data the following information can be extracted:

- The general behaviour is the same from the device Motorola to the device Samsung tablet when both are measured by the same scanning device. This could be used to perform the detection of the movement.

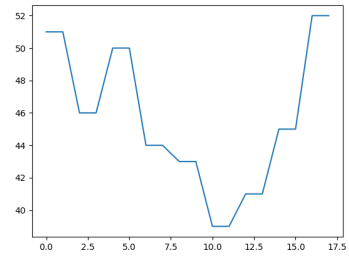
- The exact behaviour from the device Motorola to the device Samsung tablet when both are measured by the same scanning device suffer small changes, this make that the detection of movement can change from one device to another having some deviation and reducing the precision.



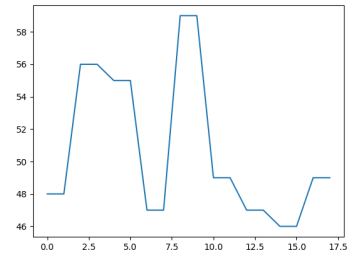
(a) Motorola RSSI A



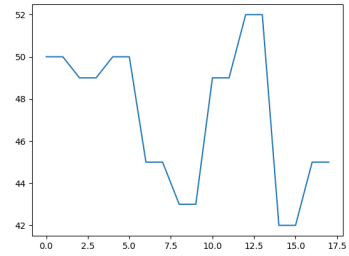
(b) Motorola RSSI B



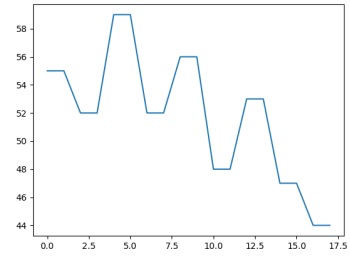
(c) Samsung tablet RSSI A



(d) Samsung tablet RSSI A

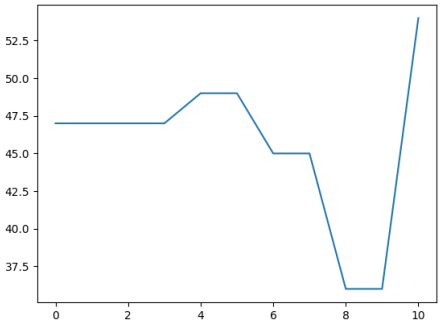


(e) Samsung tablet RSSI A

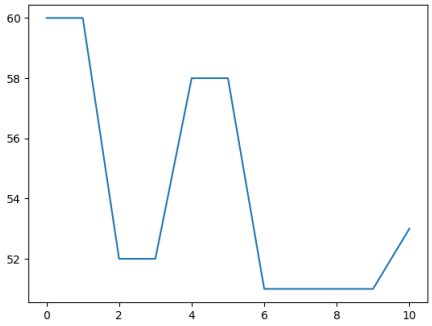


(f) Samsung tablet RSSI A

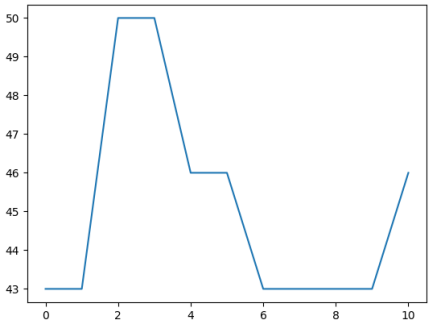
Figure 4.34: Test 3 measurement for path 1



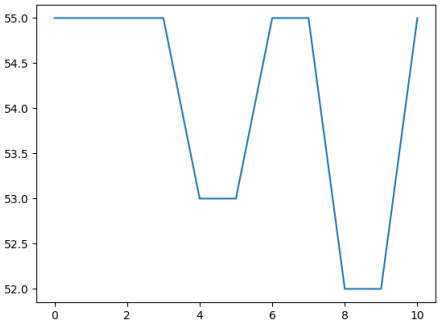
(a) Motorola RSSI A



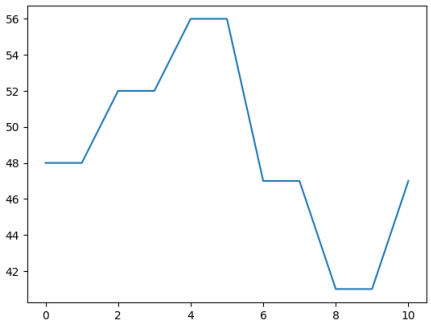
(b) Motorola RSSI B



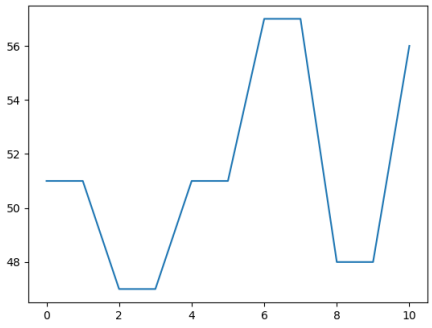
(c) Samsung tablet RSSI A



(d) Samsung tablet RSSI A

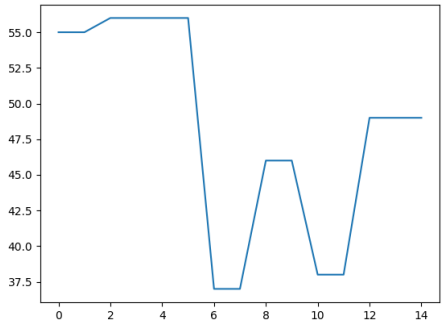


(e) Samsung tablet RSSI A

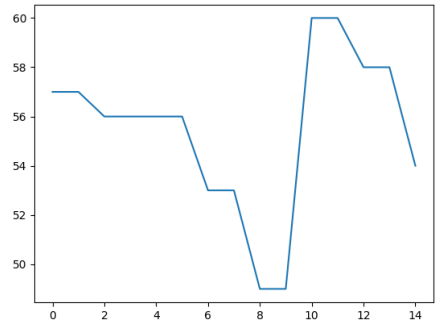


(f) Samsung tablet RSSI A

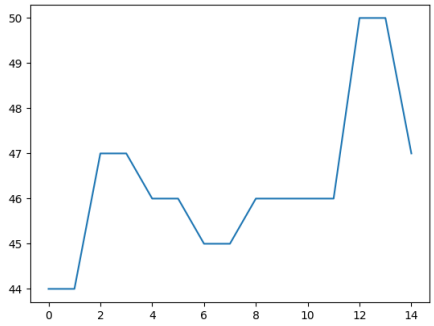
Figure 4.35: Test 3 measurement for path 3



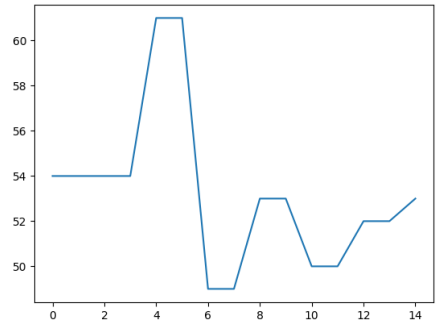
(a) Motorola RSSI A



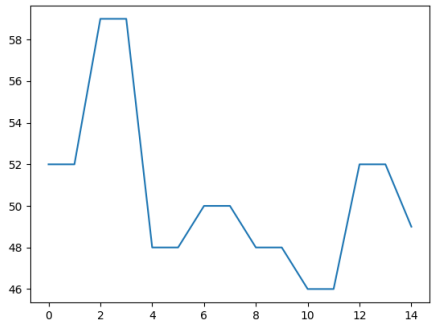
(b) Motorola RSSI B



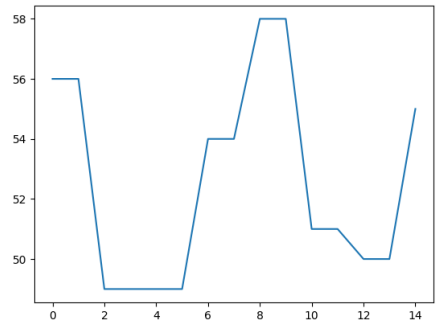
(c) Samsung tablet RSSI A



(d) Samsung tablet RSSI A



(e) Samsung tablet RSSI A



(f) Samsung tablet RSSI A

Figure 4.36: Test 3 measurement for path 4

9. **Zone differentiation** The last test done is the one that using the data from the previous tests can determine the zones where different devices are placed, the test compares three methods for the zone differentiation. In the table 4.12 it can be seen that as the method is has a bigger complexity it also has better results, although the computation time is also greater.

	Manual	Manual with weighted values	Automatic with reference devices
Precision with similar device	<10%	≈40%	>80%
Precision with different device	<5%	≈20%	>60%
Computation time	0.003234863	0.070149898529	0.71599507331

Table 4.12: Zone differentiation method comparison

4.2.2.3 Conclusions for radio waves

The conclusions for the testing of the radio waves method are:

- With a low number of scanning devices inside a normal size room several zones can be distinguish, taking into account the two methods for separating the zones, manual and automatic, it can be said that the precision of the detection is good, although it is dependable of the quantity of scanners and the size of the room partitions.
- The movement can be tracked when the device to be tracked is connected to the same net as the scanners. In the case that they are not connected to the same net the movement tracking is not possible but the detection of the number of devices can be done.
- The usage of the device and the people going through the middle of the line of sight from the scanner to the tracked devices affect to the precision of the detection. The precision can be improved by updating the values for dividing the zones.
- The hardware and the conditions of the different devices distorts the measurements, as it can be prove by the usage during the testing of different devices, and by the usage of the same device, Samsung tablet, with different conditions it can be proved the dependency characteristics such as battery or age of the components as well as the hardware.
- As each device has a different behaviour in with the RSSI and because of thee characteristics of the room it can have some changes, it is a good approach to obtain the position with a weighted value from the RSSI and the previous values.

Chapter 5

Design and implementation of a hybrid solution

Chapter 6

Results and discussion

Chapter 7

Budget and Gantt chart

Bibliography

[1] F. Chollet, *Deep Learning with Python*, 1st ed. Greenwich, CT, USA: Manning Publications Co., 2017.

[2] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," 2003.

[3] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," 2001.

[4] T. Malisiewicz, A. Gupta, and A. A. Efros, "Ensemble of exemplar-SVMs for object detection and beyond," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011.

[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, 2005.

[6] GoogleResearch, "TensorFlow: Large-scale machine learning on heterogeneous systems," *Google Research*, 2015.

[7] "TensorFlow." [Online]. Available: https://www.tensorflow.org/

[8] "Home - Keras Documentation." [Online]. Available: https://keras.io/

[9] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv*, 2018.

[10] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014.

[12] C. Szegedy, S. Reed, D. Erhan, D. Anguelov, and S. Ioffe, "Scalable, High-Quality Object Detection," 2014. [Online]. Available: http://arxiv.org/abs/1412.1441

[13] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet," *Advances in Neural Information Processing Systems 25 (NIPS2012)*, 2012.

[15] Z. Xu, L. Zhu, and Y. Yang, "Few-shot object recognition from machine-labeled web images," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.

[16] D. E. King, "dlib." [Online]. Available: https://github.com/davisking/dlib

[17] A. Ponnusamy, "cvlib." [Online]. Available: https://github.com/arunponnusamy/cvlib

[18] M. OLAFENWA, "ImageAI." [Online]. Available: https://github.com/OlafenwaMoses/ImageAI

[19] J. Redmon, "YOLO." [Online]. Available: https://pjreddie.com/darknet/yolo/

[20] G. Huang and E. Learned-Miller, "Labeled Faces in the Wild: Updates and New Reporting Procedures," *People.Cs.Umass.Edu*, 2014.

[21] V. Jain and E. Learned-Miller, "Fddb: A benchmark for face detection in unconstrained settings," University of Massachusetts, Amherst, Tech. Rep. UM-CS-2010-009, 2010.

[22] S. Yang, P. Luo, C. C. Loy, and X. Tang, "Wider face: A face detection benchmark," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[23] A. Geitgey, "face_recognition." [Online]. Available: https://github.com/ageitgey/face{_}recognition

[24] Z. A. Deng, G. Wang, D. Qin, Z. Na, Y. Cui, and J. Chen, "Continuous indoor positioning fusing WiFi, smartphone sensors and landmarks," *Sensors (Switzerland)*, vol. 16, no. 9, 2016.

[25] W. Bejuri and M. Mohamad, "Ubiquitous WLAN/Camera Positioning using Inverse Intensity Chromaticity Space-based Feature Detection and Matching: A Preliminary Result," *arXiv preprint arXiv:1204.2294*, 2012.

[26] Y. E. Dari, S. S. Suyoto, and P. P. Pranowo, "CAPTURE: A Mobile Based Indoor Positioning System using Wireless Indoor Positioning System," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 12, no. 1, p. 61, 2018.

[27] I. Cushman, D. B. Rawat, A. Bhimraj, and M. Fraser, "Experimental approach for seeing through walls using Wi-Fi enabled software defined radio technology," *Digital Communications and Networks*, 2016.

[28] F. Adib, Z. Kabelac, D. Katabi, R. C. Miller, and F. C. Adib Zachary Kabelac Dina Katabi Robert Miller, "3D Tracking via Body Radio Reflections," in *Proceedings of the 11th USENIX Symposium on Networked Systems Design and Implementation (NSDI '14)*, 2014.

[29] F. Adib, Z. Kabelac, and D. Katabi, "Multi-Person Localization via RF Body Reflections," in *Proceedings of the 12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*, 2015.

[30] G. K.Nanani and K. M V V Prasad, "A Study of WI-FI based System for Moving Object Detection through the Wall," *International Journal of Computer Applications*, vol. 79, no. 7, pp. 15–18, 2013.

[31] A. Bekkelien, "Bluetooth indoor positioning," *Master's thesis, University of Geneva*, no. March, p. 1, 2012.

[32] M. Altini, D. Brunelli, E. Farella, and L. Benini, "Bluetooth indoor localization with multiple neural networks," *ISWPC 2010 - IEEE 5th International Symposium on Wireless Pervasive Computing 2010*, no. June, pp. 295–300, 2010.

[33] B. Alsinglawi, T. Liu, Q. V. Nguyen, U. Gunawardana, A. Maeder, and S. Simoff, "Passive RFID localisation framework in smart homes healthcare settings," *Studies in Health Technology and Informatics*, vol. 231, pp. 1–8, 2016.

[34] K. Weekly, H. Zou, L. Xie, Q. S. Jia, and A. M. Bayen, "Indoor occupant positioning system using active rfid deployment and particle filters," *Proceedings - IEEE International Conference on Distributed Computing in Sensor Systems, DCOSS 2014*, pp. 35–42, 2014.

[35] E. Ceseracciu, Z. Sawacha, and C. Cobelli, "Comparison of markerless and marker-based motion capture technologies through simultaneous data collection during gait: Proof of concept," *PLoS ONE*, 2014.

[36] G. Cheung, T. Kanade, J.-Y. Bouguet, and M. Holler, "A real time system for robust 3D voxel reconstruction of human motions," 2002.

[37] H. H. Hsu, W. J. Peng, T. K. Shih, T. W. Pai, and K. L. Man, "Smartphone indoor localization with accelerometer and gyroscope," in *Proceedings - 2014 International Conference on Network-Based Information Systems, NBiS 2014*, 2014.

[38] C. H. Hsu and C. H. Yu, "An Accelerometer based approach for indoor localization," in *UIC-ATC 2009 - Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing in Conjunction with the UIC'09 and ATC'09 Conferences*, 2009.

[39] J. Scarlett, "AN-900 Enhancing the Performance of Pedometers Using a Single Accelerometer Application Note (Rev. 0)," pp. 1–16, 2007.

[40] G. Hasan, K. Hasan, R. Ahsan, T. Sultana, and R. C. Bhowmik, "Evaluation of a Low-Cost MEMS IMU for Indoor Positioning System," *International Journal of Emerging Science and Engineering*, vol. 1, no. 11, pp. 70–77, 2013.

[41] G. Papandreou, T. Zhu, L. C. Chen, S. Gidaris, J. Tompson, and K. Murphy, "Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018.

[42] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, and K. Murphy, "Towards accurate multi-person pose estimation in the wild," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.

[43] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2d human pose estimation: New benchmark and state of the art analysis," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[44] J. Charles, T. Pfister, D. Magee, D. Hogg, and A. Zisserman, "Personalizing human video pose estimation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[45] T. Pfister, K. Simonyan, J. Charles, and A. Zisserman, "Deep convolutional neural networks for efficient pose estimation in gesture videos," in *Asian Conference on Computer Vision*, 2014.

[46] "PoseTrack." [Online]. Available: https://posetrack.net/

[47] L. Xiao, "deep-high-resolution-net.pytorch." [Online]. Available: https://github.com/leoxiaobin/deep-high-resolution-net.pytorch

[48] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep High-Resolution Representation Learning for Human Pose Estimation," 2019. [Online]. Available: http://arxiv.org/abs/1902.09212

[49] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields," dec 2018. [Online]. Available: http://arxiv.org/abs/1812.08008

[50] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *CVPR*, 2017.

[51] Google, "posenet." [Online]. Available: https://github.com/tensorflow/tfjs-models/tree/master/posenet

[52] H. Wallon, "The emotions," *INT.J.MENT.HLTH*, 1972.

[53] W. B. Cannon, "The James-Lange Theory of Emotions: A Critical Examination and an Alternative Theory," *The American Journal of Psychology*, 2006.

[54] J. Kumari, R. Rajesh, and K. M. Pooja, "Facial Expression Recognition: A Survey," *Procedia Computer Science*, vol. 58, pp. 486–491, 2015.

[55] "Cohn-Kanade (CK and CK+) database Download Site." [Online]. Available: http://www.consortium.ri.cmu.edu/ckagree/

[56] T. Kanade, J. F. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis," in *Proceedings - 4th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2000*, 2000.

[57] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended Cohn-Kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, CVPRW 2010*, 2010.

[58] "MMI Facial Expression Database - Home." [Online]. Available: https://mmifacedb.eu/

[59] M. Valstar and M. Pantic, "Induced Disgust, Happiness and surprise: an addition to the MMI Facial Expression Database," in *Proc Intl Conf Language Resources and Evaluation*, 2010.

[60] J. Susskind, a. Anderson, and G. E. Hinton, "The Toronto face dataset," *U. Toronto, Tech. Rep. UTML TR*, 2010.

[61] T. Valderas, J. Bolea, P. Laguna, and S. M. Ieee, "Human Emotion Recognition Using Heart Rate Variability Analysis with Spectral Bands Based on Respiration," pp. 6134–6137, 2015.

[62] S. Wiens, E. S. Mezzacappa, and E. S. Katkin, "Heartbeat detection and the experience of emotions," *Cognition and Emotion*, 2000.

[63] M. Zhao, F. Adib, and D. Katabi, "Emotion recognition using wireless signals," *Communications of the ACM*, 2018.

[64] E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation in computer vision: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009.

[65] R. Valenti, N. Sebe, and T. Gevers, "Combining head pose and eye location information for gaze estimation," *IEEE Transactions on Image Processing*, 2012.

[66] J. G. Wang and E. Sung, "Study on eye gaze estimation," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 2002.

[67] Z. Schollz, "Find3." [Online]. Available: https://github.com/schollz/find3

Appendices

Appendix A

Tables of the Wi-Fi testing

Mean	Motorola	Tablet 1(MAC 7E)	Tablet 2(Mac 78)
First measure	-39.63	-57.0	-43.16
Second measure	-51.0	-53.0	-39.45
Third measure	-54.55	-48.45	-48.18
Fourth measure	-51.1	-36.18	-58.1
Variance	Motorola	Tablet 1(MAC 7E)	Tablet 2(Mac 78)
First measure	0.23	1.0	0.472
Second measure	0.36	15.63	0.2479
Third measure	3.15	6.066	0.158
Fourth measure	10.81	0.1487	1.355
Maximum difference	Motorola	Tablet 1(MAC 7E)	Tablet 2(Mac 78)
First measure	1.0	3.0	2.0
Second measure	2.0	11.0	1.0
Third measure	4.0	8.0	1.0
Fourth measure	9.0	1.0	4.0

Table A.1: Data from Raspberry A in test 1

Mean	Motorola	Tablet 1(MAC 7E)	Tablet 2(Mac 78)
First measure	-64.0	-67.5	-59.0
Second measure	-65.27	-44.27	-64.64
Third measure	-33.91	-56.55	-60.09
Fourth measure	-63.1	-67.73	-42.54
Variance	Motorola	Tablet 1(MAC 7E)	Tablet 2(Mac 78)
First measure	2.0	11.58	4.33
Second measure	3.47	3.29	1.32
Third measure	0.083	3.15	2.44
Fourth measure	5.72	2.74	0.61
Maximum difference	Motorola	Tablet 1(MAC 7E)	Tablet 2(Mac 78)
First measure	4.0	10.0	6.0
Second measure	5.0	5.0	3.0
Third measure	1.0	4.0	5.0
Fourth measure	7.0	1.0	2.0

Table A.2: Data from Raspberry B in test 1

Mean	Motorola	Tablet 1(MAC 7E)	Tablet 2(Mac 78)
First measure	-52.75	-53.0	-49.91
Second measure	-42.72	-36.0	-37.54
Third measure	-59.54	-52.81	-58.9
Variance	Motorola	Tablet 1(MAC 7E)	Tablet 2(Mac 78)
First measure	2.52	7.0	0.409
Second measure	2.92	0.364	0.43
Third measure	15.52	2.33	7.53
Maximum difference	Motorola	Tablet 1(MAC 7E)	Tablet 2(Mac 78)
First measure	5.0	10.0	2.0
Second measure	4.0	2.0	2.0
Third measure	11.0	4.0	9

Table A.3: Data from Raspberry A in test 2

Mean	Motorola	Tablet 1(MAC 7E)	Tablet 2(Mac 78)
First measure	-63.91	-61.75	-58.0
Second measure	-63.91	-66.36	-65.0
Third measure	-49.0	-43.55	-41.18
Variance	Motorola	Tablet 1(MAC 7E)	Tablet 2(Mac 78)
First measure	4.07	8.021	0.83
Second measure	1.17	7.32	8.36
Third measure	54.72	0.24	2.51
Maximum difference	Motorola	Tablet 1(MAC 7E)	Tablet 2(Mac 78)
First measure	6.0	8.0	2.0
Second measure	3.0	8.0	9.0
Third measure	19.0	1.0	4.0

Table A.4: Data from Raspberry B in test 2

Mean	Motorola	Tablet 2(Mac 78)
First measure	-49.76	-46.0
Second measure	-45.63	-47.875
Third measure	-46.58	-48.41
Fourth measure	-45.71	-48.43
Variance	Motorola	Tablet 2(Mac 78)
First measure	22.06	29.75
Second measure	6.73	15.609
Third measure	19.07	17.409
Fourth measure	12.77	10.81
Maximum difference	Motorola	Tablet 2(Mac 78)
First measure	12.0	18.0
Second measure	7.0	14.0
Third measure	17.0	12.0
Fourth measure	11	9

Table A.5: Data from Raspberry A in test 3

Mean	Motorola	Tablet 2(Mac 78)
First measure	-58.23	-58.11
Second measure	-58.13	-58.5
Third measure	-53.75	-53.83
Fourth measure	-58.42	-57.28
Variance	Motorola	Tablet 2(Mac 78)
First measure	20.88	14.57
Second measure	67.11	29.0
Third measure	5.68	30.64
Fourth measure	38.81	19.63
Maximum difference	Motorola	Tablet 2(Mac 78)
First measure	14	12
Second measure	25	18
Third measure	9	18
Fourth measure	21	12

Table A.6: Data from Raspberry Bin test 3

Mean	Motorola	Motorola (covered)	Tablet 2(Mac 78)	Tablet 2(Mac 78) (covered)
First measure	-54.83	-48.55	-61.0	-53.45
Second measure	-51.66	-51.43	-58.11	-59.85
Third measure	-53.25	-50.42	-53.625	-51.48
Fourth measure	-42.75	-49.66	-53.0	-41.0
Variance	Motorola	Motorola (covered)	Tablet 2(Mac 78)	Tablet 2(Mac 78) (covered)
First measure	13.97	2.066	2.0	4.43
Second measure	4.88	5.67	4.98	4.12
Third measure	3.43	1.67	2.73	1.67
Fourth measure	2.18	0.22	10.28	0.0
Maximum difference	Motorola	Motorola (covered)	Tablet 2(Mac 78)	Tablet 2(Mac 78) (covered)
First measure	11.0	3.0	4.0	6.0
Second measure	8.0	6.0	6.0	5.0
Third measure	4.0	3.0	5.0	3.0
Fourth measure	4.0	1.0	11	0.0

Table A.7: Data from Raspberry A in test 4

Mean	Motorola	Motorola (covered)	Tablet 2(Mac 78)	Tablet 2(Mac 78) (covered)
First measure	-40.66	-45.63	-46.0	-44.27
Second measure	-56.22	-60.85	-66.66	-62.85
Third measure	-68.5	-58.71	-65.75	-68.42
Fourth measure	-64.75	-59.88	-69.85	-56.88
Variance	Motorola	Motorola (covered)	Tablet 2(Mac 78)	Tablet 2(Mac 78) (covered)
First measure	2.38	1.32	8.66	0.56
Second measure	2.39	2.97	4.0	10.97
Third measure	1.75	0.21	7.18	1.67
Fourth measure	10.68	5.65	0.12	0.32
Maximum difference	Motorola	Motorola (covered)	Tablet 2(Mac 78)	Tablet 2(Mac 78) (covered)
First measure	4.0	3.0	8.0	2.0
Second measure	4.0	5.0	6.0	9.0
Third measure	4.0	1.0	6.0	3.0
Fourth measure	8.0	7.0	1.0	2.0

Table A.8: Data from Raspberry B in test 4

Mean	Motorola (covered)	Tablet 1 (Mac 7E) (covered)	Tablet 2 (Mac 78) (covered)
First measure	-60.0	-67.0	-57.2
Second measure	-50.55	-49.33	-87
Variance	Motorola (covered)	Tablet 1 (Mac 7E) (covered)	Tablet 2 (Mac 78) (covered)
First measure	9.6	15.2	0.559
Second measure	0.25	6.88	0.0
Maximum difference	Motorola (covered)	Tablet 1 (Mac 7E) (covered)	Tablet 2 (Mac 78) (covered)
First measure	8.0	10.0	2.0
Second measure	1.0	9.0	0.0

Table A.9: Data from Raspberry A in test 5

Mean	Motorola	Tablet 1 (Mac 7E)	Tablet 2 (Mac 78)
First measure	-56.1	-51.5	-52.75
Second measure	-53.71	-51.85	-50.85
Variance	Motorola	Tablet 1 (Mac 7E)	Tablet 2 (Mac 78)
First measure	17.49	4.25	0.9375
Second measure	7.91	17.55	3.83
Maximum difference	Motorola	Tablet 1 (Mac 7E)	Tablet 2 (Mac 78)
First measure	12.0	5.0	3.0
Second measure	7.0	10.0	5.0

Table A.10: Data from Raspberry A in test 5

Mean	Motorola (covered)	Tablet 1 (Mac 7E) (covered)	Tablet 2 (Mac 78) (covered)
First measure	-60.4	-21.0	-56.63
Second measure	-42.57	-38.14	-50.14
Variance	Motorola (covered)	Tablet 1 (Mac 7E) (covered)	Tablet 2 (Mac 78) (covered)
First measure	8.64	7.5	7.48
Second measure	27.67	2.40	0.41
Maximum difference	Motorola (covered)	Tablet 1 (Mac 7E) (covered)	Tablet 2 (Mac 78) (covered)
First measure	8.0	7.0	7.0
Second measure	13.0	5.0	2.0

Table A.11: Data from Raspberry B in test 5

Mean	Motorola	Tablet 1 (Mac 7E)	Tablet 2 (Mac 78)
First measure	-59.4	-64.0	-60.8
Second measure	-45.66	-47.22	-54.0
Variance	Motorola	Tablet 1 (Mac 7E)	Tablet 2 (Mac 78)
First measure	12.64	5.2	9.36
Second measure	24.66	21.50	0
Maximum difference	Motorola	Tablet 1 (Mac 7E)	Tablet 2 (Mac 78)
First measure	10.0	6.0	7.0
Second measure	13	14	0

Table A.12: Data from Raspberry B in test 5

Mean	Motorola	Tablet 1(MAC 7E)	Tablet 2(Mac 78)
First measure	-46.0	-52.27	-44.1
Second measure	-50.16	-44.83	-36.5
Third measure	-50.0	-43.5	-57.5
Fourth measure	-50.55	-36.1	-60.0
Variance	Motorola	Tablet 1(MAC 7E)	Tablet 2(Mac 78)
First measure	3.1	1.83	0.45
Second measure	1.47	0.47	0.25
Third measure	21.66	0.583	1.25
Fourth measure	0.61	0.81	5.45
Maximum difference	Motorola	Tablet 1(MAC 7E)	Tablet 2(Mac 78)
First measure	5.0	4.0	2.0
Second measure	4.0	2.0	1.0
Third measure	14.0	2.0	3.0
Fourth measure	2.0	2.0	7.0

Table A.13: Data from Raspberry A in test 6

Mean	Motorola	Tablet 1(MAC 7E)	Tablet 2(Mac 78)
First measure	-52.63	-36.54	-49.64
Second measure	-46.33	-59.5	-53.33
Third measure	-61.0	-53.66	-9.0
Fourth measure	-46.64	-51.36	-56.72
Variance	Motorola	Tablet 1(MAC 7E)	Tablet 2(Mac 78)
First measure	3.51	279.52	0.95
Second measure	93.88	7.58	1.55
Third measure	14.0	16.55	0
Fourth measure	0.23	0.95	5.65
Maximum difference	Motorola	Tablet 1(MAC 7E)	Tablet 2(Mac 78)
First measure	6.0	44.0	3.0
Second measure	29.0	8.0	3.0
Third measure	11.0	11.0	0
Fourth measure	1.0	3.0	6.0

Table A.14: Data from Raspberry B in test 6

Mean	Motorola	Tablet 1(MAC 7E)	Tablet 2(Mac 78)
First measure	-57.0	-58.27	-58.72
Second measure	-32.66	-35.36	-38.81
Third measure	-59.45	-47.0	-58.45
Variance	Motorola	Tablet 1(MAC 7E)	Tablet 2(Mac 78)
First measure	10.72	48.74	3.47
Second measure	0.22	0.23	0.15
Third measure	1.70	0.36	0.24
Maximum difference	Motorola	Tablet 1(MAC 7E)	Tablet 2(Mac 78)
First measure	10.0	15.0	7.0
Second measure	1.0	1.0	1.0
Third measure	4.0	2.0	1.0

Table A.15: Data from Raspberry A in test 7

Mean	Motorola	Tablet 1(MAC 7E)	Tablet 2(Mac 78)
First measure	-61.72	-57.0	-59.63
Second measure	-60.66	-58.09	-48.72
Third measure	-29.81	-44.36	-26.36
Variance	Motorola	Tablet 1(MAC 7E)	Tablet 2(Mac 78)
First measure	3.65	65.09	2.95
Second measure	15.38	2.44	0.74
Third measure	15.23	5.14	0.23
Maximum difference	Motorola	Tablet 1(MAC 7E)	Tablet 2(Mac 78)
First measure	5.0	20.0	5.0
Second measure	11.0	5.0	3.0
Third measure	11.0	7.0	1.0

Table A.16: Data from Raspberry B in test 7

Mean	Motorola	Tablet 1 (Mac 7E)	Tablet 2(Mac 78)
First measure	-48.0	-45.66	-47.22
Second measure	-45.63	-45.09	-48.63
Third measure	-48.2	-46.33	-50.6
Variance	Motorola	Tablet 1 (Mac 7E)	Tablet 2(Mac 78)
First measure	34.66	18.22	11.061
Second measure	26.05	6.99	23.32
Third measure	53.77	2.88	14.77
Maximum difference	Motorola	Tablet 1 (Mac 7E)	Tablet 2(Mac 78)
First measure	17.0	13.0	10.0
Second measure	18.0	7.0	15.0
Third measure	19.0	6.0	13.0

Table A.17: Data from Raspberry A in test 8

Mean	Motorola	Tablet 1 (Mac 7E)	Tablet 2(Mac 78)
First measure	-52.11	-50.66	-51.77
Second measure	-54.27	-54.09	-51.27
Third measure	-55.46	-53.26	-52.70
Variance	Motorola	Tablet 1 (Mac 7E)	Tablet 2(Mac 78)
First measure	25.87	19.77	19.95
Second measure	13.47	1.537	13.29
Third measure	10.51	12.195	10.77
Maximum difference	Motorola	Tablet 1 (Mac 7E)	Tablet 2(Mac 78)
First measure	17.0	13.0	15.0
Second measure	9.0	3.0	10.0
Third measure	11.0	12.0	9.0

Table A.18: Data from Raspberry B in test 8