







**AALBORG UNIVERSITY**  
STUDENT REPORT

4<sup>th</sup> Semester

Vision, Graphic and Interactive Systems

Fredrik Bajers vej 7C

9220 Aalborg Ø

<http://www.sict.aau.dk>

**Title:**

Sports Tracking by Correlating  
Thermal Video with Inertia Data

**Project Theme:**

Computer Vision

**Project Period:**

February 2017–June 2017

**Project Group:**

1001

**Members:**

Mike Røntved

Niels Dyremose Bodin

**Supervisors:**

Rikke Gade

Thomas B. Moeslund

**No. Printed Copys: 5**

**No. of Pages: 94**

**Completed: 2017-06-08**

**Abstract:**

The aim of this project is to correlate thermal video with an accelerometer and be able to connect tracklets made from a video tracker to trajectories for each person.

The project resulted in developing a pipeline capable of detecting players, tracking them, calculate acceleration in the video to match with the acceleration from the accelerometers and from this be able to match accelerometer to tracklets.

The report covers all the blocks in this process: Detection, tracking and matching, where each block have been designed, implemented and tested.

The proof-of-concept system has been tested on a dataset made of half a field in Gigantium sports arena, where the participants played football. The system shows promising results and that it is possible to correlate thermal video with an accelerometer, but the tracking system requires more work to know if it can be implemented in an automatic system.



# Preface

This project is developed by group 1001 and it ends the master of Vision, Graphics and Interactive Systems.

We would like to thank Jesper Franch from department of Health Science and Technology, for providing the group with the accelerometers used in the project.

In the report figures will be referred to as Figure 1.1, where the first number indicates the chapter in which the figure can be found. The second number refers the the figure number in this chapter.

Tables will be referred to as Table 2.1, where first number indicates the chapter in which the table can be found and second number refers to the table number

Chapters and sections are referred to as section 3.2.1, where first number indicates the chapter, second number indicates the section and third number indicates the subsection.

Citing will be referred to as [1] where the number indicates the number in the bibliography.



# Projekt Opsummering (Danish)

I kapitel 1, blev projektforslaget introduceret, hvilket har ledt til en initierende problemstilling.

Den initerende problemstilling var grundlag for den tekniske analyse i kapitel 2, hvor tidlige arbejde indenfor projektets tema er undersøgt, med fokus på detektion og featureless tracking metoder. Derefter blev udstyr, som er anvendt i projektet undersøgt og testet, der bliver bla. beskrevet det data udstyret giver og hvordan det anvendes. Undersøgelser viser at accelerometerene ikke er synkroniseret med hinanden, hvilket resulterede i et behov for at løse dette problem. Det data, som er optaget og brugt i projektet er vist og forklaret, samt en beskrivelse af, hvordan ground truth labels er lavet. Den tekniske analyse slutter til sidst af med en problemstilling, samt specifikationer og krav.

I kapitel 3, er et tracking system udviklet, som er i stand til at detektere og følge individuelle spillere i et indørs scenarie. Tracking algoritmen kan detektere når der forekommer en occlusion (når to personer står oveni hinanden, så der ikke kan skelnes forskel). Systemet er derudover i stand til at detektere når en occlusion bliver separeret til individuelle spillere igen. Trackeren giver informationer for hver tracklet, såsom: verdenskoordinater, mængden af personer i tracket, samt hvor langtid personen er tracket. En metode til at konvertere positioner om til acceleration er blevet implementeret, således det er muligt at sammenligne accelerationen der er estimeret ud fra det termiske kameras video sammen med accelerometers målte data. For at kunne synkronisere accelerometer med video, er der blevet anvendt en dot product maximisation. Når dataen er synkroniseret kan hvert tracklet blive matched med hvert accelerometer og derved kan det bedste match findes. I projektet er der anvendt flere forskellige metoder for test og evaluering af matching, heriblandt: dot product maximisation, Russell-Rao og Hungarian algoritme.

I kapitel 4, er systemet blevet testet på et kaos scenarie, hvor en snippet fra dataen optaget i Gigantium er blevet anvendt. Først blev tracking systemet testet, hvor CLEAR MOT metrics findes, hvilket fortæller hvor robust trackeren er. Tracklets, fundet af trackeren, blev brugt til at teste systems evne til at matche tracklets med accelerometerne. Dette var testet med alle metoder præsenteret i design og implementation, kapitel 3. Senere blev der matched mellem tracklet og accelerometer testet, for at finde ud af hvor meget af et tracklet er nødvendig for at foretage et match. Til sidst blev der testet om systemet er i stand til at forbinde tracklets efter en occlusion.

I kapitel 5, er konklusionen af projektet, samt en diskussion på hvad der skal være fokus i en fremtidig udvikling og hvad der skal gøres anderledes. Konklusionen på projektet er at det er muligt at korrelere accelerometers data med video data og derved forbinde tracklets til trajectories. Dog er resultaterne ikke så solide ved kaos scenariet, hvilket kan skyldes den tracking metode der er blevet anvendt og synkroniserings problemer.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Project Proposal . . . . .	2
1.2	Initial Problem Statement . . . . .	2
<b>2</b>	<b>Technical Analysis</b>	<b>5</b>
2.1	Previous Work . . . . .	5
2.1.1	Feature-less Tracking Methods . . . . .	5
2.2	Devices Used in the Project . . . . .	6
2.2.1	Accelerometer . . . . .	7
2.2.2	Thermal Camera . . . . .	9
2.2.3	From Position to Acceleration . . . . .	10
2.2.4	Time-Sync Between Devices . . . . .	12
2.3	Data Gathering . . . . .	14
2.3.1	Ground Truth Creation . . . . .	15
2.3.2	Data Definition . . . . .	16
2.4	Summary of Technical Analysis . . . . .	17
2.5	Problem Statement . . . . .	18
2.6	Specifications and Requirements . . . . .	18
<b>3</b>	<b>Design and Implementation</b>	<b>19</b>
3.1	Description of the System . . . . .	19
3.2	Detection and Tracking . . . . .	21
3.2.1	Detection . . . . .	21
3.2.2	Tracking . . . . .	23
3.2.3	ID Class . . . . .	26
3.2.4	Example of Tracking . . . . .	26
3.3	From Pixel to World Coordinates . . . . .	28
3.3.1	Perspective Transformation . . . . .	30
3.4	Synchronising Accelerometers and Video . . . . .	32
3.4.1	Matching 9 Trajectories With 9 Accelerometers . . . . .	36
3.5	42 Trajectories Matched to 9 Accelerometers . . . . .	39
3.5.1	Conclusion . . . . .	43
3.6	Certainty Threshold . . . . .	44
<b>4</b>	<b>Acceptance Test</b>	<b>47</b>
4.1	Test of the Tracking System . . . . .	47
4.1.1	CLEAR MOT Metrics . . . . .	48
4.2	Matching Tracking Results to Accelerometers . . . . .	52
4.2.1	Conclusion of the Test . . . . .	53
4.3	Test of Minimum Samples Required for Matching . . . . .	53

---

4.4	Connecting Tracklets After an Occlusion . . . . .	54
4.5	Conclusions of the Acceptance Test . . . . .	58
<b>5</b>	<b>Conclusion and Discussion</b>	<b>59</b>
5.1	Project Summary . . . . .	59
5.2	Assessment . . . . .	60
5.3	Discussion . . . . .	60
	<b>Bibliography</b>	<b>63</b>
<b>A</b>	<b>Appendix</b>	<b>65</b>
A.1	Matching Between Devices - Data . . . . .	65
A.2	42 To 9 Trajectories . . . . .	79
A.3	Matching Tracking Results to Accelerometers . . . . .	83



# Introduction

---

Over the past years, self-tracking has become an increasing trend in society, which people perform every day. Self-trackers measure a vast amount of information about themselves. The data gathered spans from steps taken, sleeping patterns to amount of water consumed each day.

The trend of self-tracking has led to various companies and wearable gadgets making self-tracking an integrated part of peoples lives. Wearables such as running shoes with sensors to watches, capable of tracking your activity. Another big contributor helping with self-tracking are the vast amount of applications developed for smartphones. An app named "Sleep Cycle" is used for tracking your sleep by measuring how a person is moving during sleep. From these measurements, the app is able to present the user with information about when and how long they have been in different sleeping cycles. By tracking their daily behaviour, people feel that they improve various aspects of their life.

People are becoming more interested in their health and the measuring of blood pressure on daily basis has increased as well[2]. The desire to track health, activity and consumption has lead to the medical sector taking an interest in their self obtained data. The data can help the medical staff propose the best treatment based on this information. A research group based in Vejle, Denmark is attempting to implement people's own data into the decision taking in terms of treatment[3].

The desire to obtain information in form of numbers about their activity has spanned into the professional sports scene. In a higher degree, professional trainers are using information about athletes to improve the training schedule. Companies like Trackman A/S[4] is a technology company focused on radar equipment for golf and baseball. By presenting golfers with information after each swing it enables the players to both visually see their mistakes and what effect it has on their swing. Another company Sports Performance Tracking[5] has developed a wearable GPS module which can be used to capture statistics of sport players during training and games. The Sport Performance Tracking GPS only works outdoor, which propose an issue for indoor sports activities.

At Aalborg University research has gone into tracking sports activities based on thermal cameras. During the research of tracking sports players using thermal imagery, several problem statements have been found. In research conducted it was found that automatic tracking of sports athletes resulted in too many fragments of a trajectory (called tracklet in the remainder of the report). This research have led to the project proposal in finding a way to connect these to reduce the amount of tracklets.

## 1.1 Project Proposal

From thermal cameras a limited amount of information can be extracted compared to colour cameras, which leads to issues when tracking people. This is especially the case when it is attempted to detect and track individual players in sports. Due to the vast amount of occlusions occurring in sport. Occlusion are the moments where a person is walking in front of another person seen from the camera. This action leads to the two people being detected as a single object. The issue with these occlusions is that once the persons are no longer blocking each other, thereby separating from the occlusion (this action is described by a split in the remainder of the report), it is hard to re-identify each person, especially regarding thermal images. This issue results in a vast amount tracklets instead of trajectories. The motivation behind this project is to develop a method to improve the tracking of the individual players, by connecting the tracklets into trajectories. The project will focus on a correlation between inertia sensors, mounted on players and thermal camera recordings to connect the tracklets. In Figure 1.1 an example can be seen which explains the problem and what the project achieve to solve.

## 1.2 Initial Problem Statement

An initial problem statement is made which provides basis for the Technical Analysis, Chapter 2.

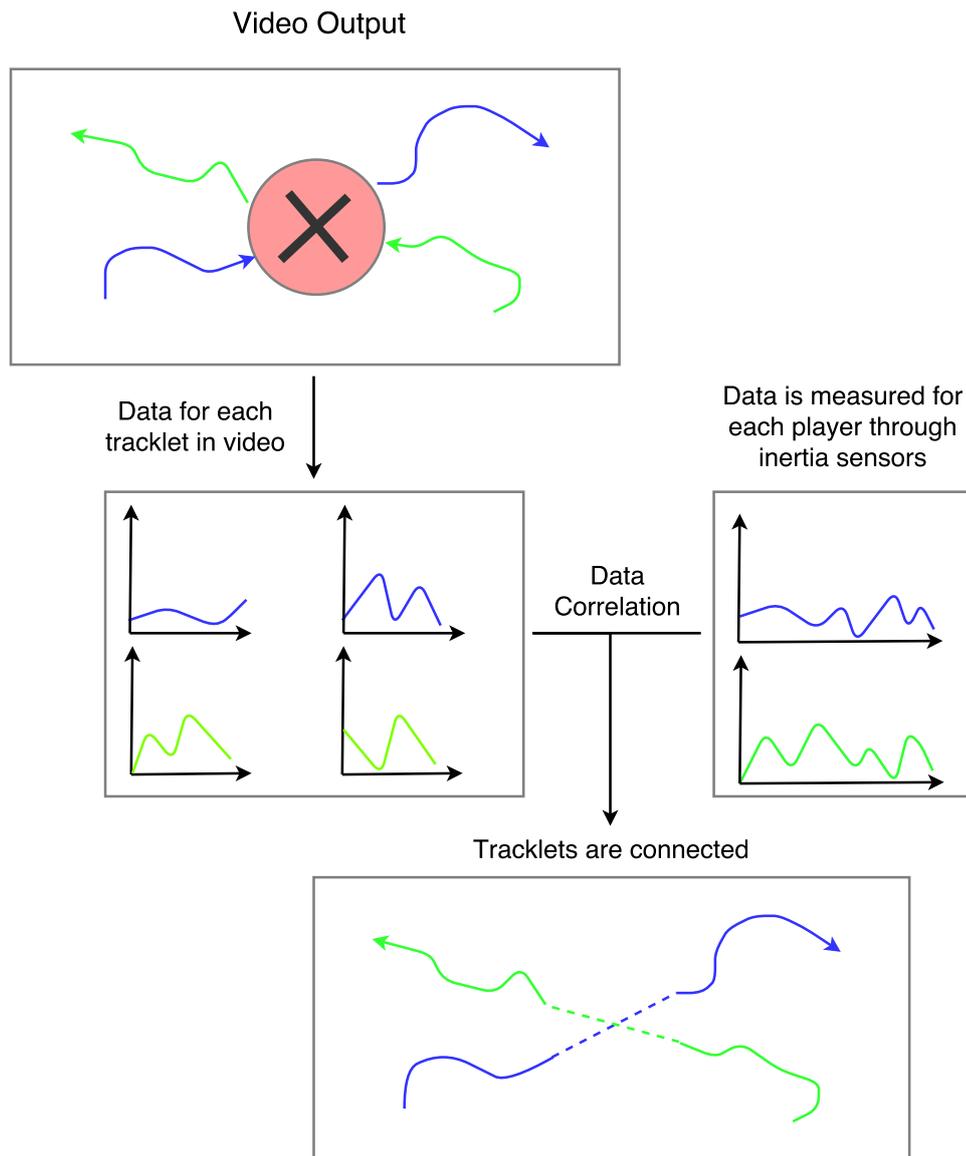


Figure 1.1: The figure shows the motivation for the project: Which is to take the broken tracklets from the video and compare these to the data from the inertia sensors and then be able to connect the broken tracklets into trajectories. The blue and green lines shows information from two different persons. The circle is where the two persons occlude and their individual tracking information is lost. The purpose is then to reconnect the two tracklets as seen in the end result.

*Is it possible to correlate data from thermal imaging with data from inertia sensor?*



In this chapter the previous work conducted within this field of research at Aalborg University will be looked upon along with some of the work within Feature-less Tracking Methods. The devices used in the project will be presented and explained, furthermore how to go from video position to an acceleration is explained. This will further lead into preliminary tests of the devices and data gathering. The chapter will result in a problem statement, specification and requirements.

## 2.1 Previous Work

This research topic has been an ongoing project at Aalborg University, which means previous work has been conducted. In 2012, a paper was released which focused on improving how sports arenas are utilised, which lead to a system observing the occupancy through the use of thermal imaging. Through thermal imaging a lot of visible features such as colours are lost, but it provides a certain anonymity for the people being recorded. The report ended in a automatic detection of people, which could distinguish between empty, few or many people in the sports arena.[6]

In 2013 the system was further developed to recognise specific sports types based on signature heat maps. This system was based on detecting the persons in the sports arena and how the people moved in the arena. The classification was done by applying Fischer's Linear Discriminant to the heat maps and calculating the mean coordinate to each class in the training heat maps. The overall recognition rate achieved in this work was 90.76%.[7]

In 2014, tracking of the individual sport players was implemented. This was done by using a Kalman Filter (KF) and Continuous Energy Minimization (CEM). From the results of both methods it was concluded that the KF provided more tracks, but they are closer to the ground truth. However the KF proved higher identity swaps than CEM, which leads to more broken trajectories[8]. It is wished to have a high precision and therefore the KF is more suitable for sports tracking. The increased amount of broken trajectories has however lead to the research topic for this report.

### 2.1.1 Feature-less Tracking Methods

As explained in chapter 1 using thermal imagery propose some issues when detecting and tracking people in a sport scenario. The lack of colour features and vast amount of occlusion occurring in sport scenarios makes general tracking method less effective. In general purpose surveillance, the first step is often to detect objects wanted. Next step is to track the detected object through a series of frames. In order to have a basis for

developing a such system, previous work conducted on tracking using thermal imagery is investigated.

In 2004, Davis et al.[9] proposed detection based on a Gaussian background subtraction method in order to define region of interest(ROI). In case of thermal halos surrounding the people a Contour Saliency Map was used to extract the contours of people. A watershed analysis was then applied to complete the contours in case of contour fragments. Their method proved the ability to separate people within a ROI. However they also mention issues if the thermal intensity of the people is similar to the background. Also the method used had tendencies to result in creating false contours.

In 2007, Zin et al.[10] proposed a robust person detection with a far infrared camera. The method proposed involved an initial extraction of head regions based on threshold and morphology operations. From extracted head regions the upper and lower body was estimated based on ratios.

In 2007 Dai et al.[11] proposed a detection and tracking method, for pedestrian tracking, which involved several steps. First step was creating a layered representation where the background was static objects and foreground was moving objects. In order to separate the two layers a light version of Generalized Expectation Maximization was used. Once the background and foreground is extracted a shape-based classification was used on the foreground in order to remove non-pedestrian objects. This was done by two perimeters being the compactness,  $r_1 = \frac{p^2}{A}$ , where p is the perimeter and A is the area of the moving object, and leanness,  $r_2 = \frac{l_v}{l_h}$ , where l is the length of the contour horizontal and vertical. Once the moving objects has been verified as pedestrians. Their tracking method is based on position of feet, head and waist. The Hausdorff distance between the positions to the previous positions was used to determine whether the pedestrian corresponds to the same pedestrian in the previous frame.

In 2015, Jeon et al.[12] proposed an alternative background extraction method, where a median filter was applied to a sequence of input images. Pixels in same position over the processed sequence was then averaged. By using an average threshold for each row the human objects in the background image is located. Once the human objects are located they are erased by using a linear interpolation by the non human objects in the horizontal line. By extracting the background and applying a Sobel kernel as edge detector to the input image, the person objects was detected.

With this introduction of previous work within detection and tracking of persons using thermal cameras, the following section will look at the devices which has been used in this project for cross correlation between thermal image information and inertia data.

## 2.2 Devices Used in the Project

In this section the devices used for the project will be presented and explained, furthermore the theory of how they work and their output will be described. Lastly which steps have been used to synchronise the devices will be explained.

### 2.2.1 Accelerometer

For this project the inertia sensor chosen is the accelerometer *Axivity AX3*[1]. The accelerometer is used in order to record the acceleration of people. *AX3* is a 3-axis logging accelerometer, which outputs the acceleration in g [ $9.82 \text{ m/s}^2$ ][13] on the sensors X-, Y- and Z-axis. The device specifications can be seen in Table 2.1.

Specification for Axivity AX3	
Parameter	Value
<b>Typical Capabilities</b>	
Memory	512 MB NAND flash non-volatile
Logging Frequencies	Configurable 12.5 Hz - 3200 Hz
Maximum Logging Periods	30 days at 12.5 Hz or 14 days at 100 Hz
<b>Real Time Clock</b>	
Type	Quartz real time clock
Frequency	32.768 KHz
Precision	$\pm 50$ ppm (typical)
<b>Accelerometer</b>	
Sensor Type	MEMS
Range	$\pm 2/4/8/16$ g
Resolution	upto 13-bit

Table 2.1: Specifications for the accelerometer sensor Axivity AX3

Since this project is based on a correlation between several accelerometer and video data it is important that the devices used in this project is synchronised. To test whether the accelerometers is synchronised, it is chosen to start seven devices, using the same settings; 25 Hz, 8 g range and start time based on a date and clock, which is taken from the computer, using the software from Axivity. All accelerometers are strapped to a stable surface and then moved around an area. It is expected the data collected from the accelerometers are the same. The raw data are extracted from the accelerometers and contains the acceleration, g, in X-, Y- and Z-axis. The data is then processed using standard vector magnitude equation (2.1).

$$g_{xyz} = (\sqrt{g_x^2 + g_y^2 + g_z^2}) - 1g \quad (2.1)$$

As accelerometers measure the "true acceleration" there is always an added acceleration due to gravity[14]. By subtracting 1 g, as seen in equation (2.1), this added acceleration is removed. In order to understand the concept behind the added acceleration, when in rest; Consider an accelerometer, consisting of 3 springs along X-, Y- and Z-axis, positioned on a plane surface. The X- and Y-axis is parallel to the plane, while the Z-axis is orthogonal from the plane. The accelerometer will in this case measure +1 g along the Z-axis, due to the plane adding 1 g acceleration in order to keep the accelerometer in steady position. If the accelerometer is free falling with no rotation around axis, all axis on the accelerometer will measure 0 g, although the accelerometer is visually moving.

If the accelerometer were static mounted, only capable of moving along the 3-axes, the Z-axis could be excluded, thereby removing the acceleration added by gravity, as the

acceleration in the plane is the interesting part in this project. However since the accelerometers are mounted on people, with unpredictable rotation around the X-, Y- and Z-axis. Since this rotation is present the added acceleration due to gravity, is affecting the acceleration in all 3 axis. In order to take the added acceleration into consideration the  $1g$  is subtracted when calculating the standard vector magnitude Equation (2.1).

The results of the test can be seen in Figure 2.1 and as the results show[15] the accelerometers are not synchronised with each other. They start at different time and therefore end at different time, but the number of samples from each sensor is the same. This shows the accelerometers clocks are not synchronised with each other. In the documentation for the accelerometers and software, it is said to synchronise the time with the PC, from which they are started. However since the accelerometers are unsynchronised it is required to shift the data in order to match the starting times for all accelerometers. Another issue found from this test is that the sensors are sensitive or have low tolerance. Only one sensor, *Accelerometer3* is resting at  $0g$ , see Figure 2.2 for a better view, when no movement is applied and the remaining accelerometers are resting around  $0.1g$ .

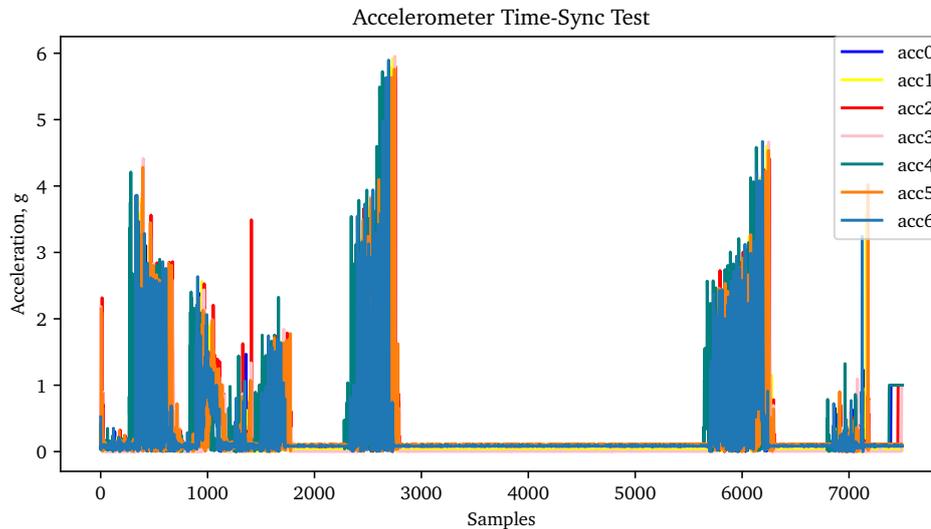


Figure 2.1: Time sync test for the accelerometers, the figure shows results for 7 devices started at the same time.

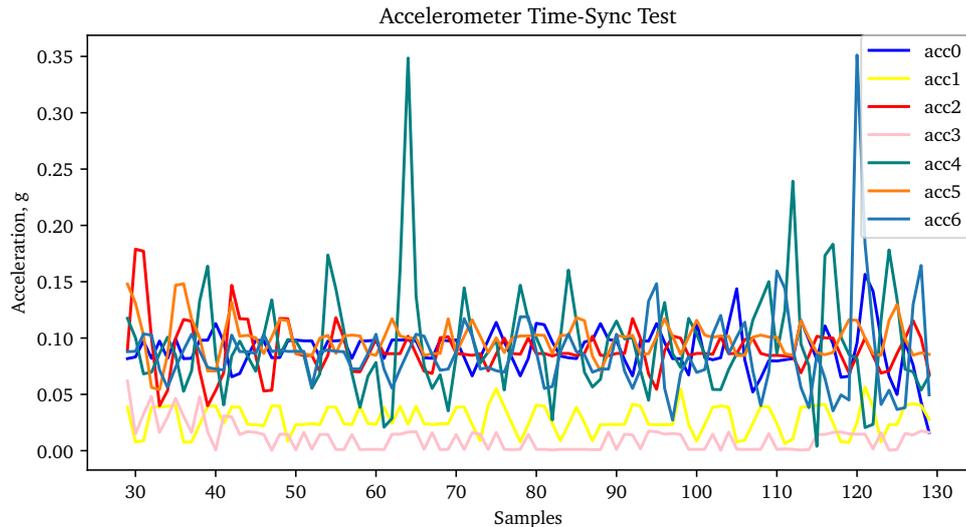


Figure 2.2: A snippet of the no movement, sample 30-130 from Figure 2.1.

## 2.2.2 Thermal Camera

This section will give an explanation on how the thermal cameras work and the theory behind, which is based on Gade et al.[16] unless otherwise specified. Furthermore the camera which have been used in the project is described.

Traditional colour cameras (RGB), measure electromagnetic radiation, or photons, in the visible light spectrum which have wavelengths between 390 and 700 nm. This range is where the human vision can perceive photons and is also where the RGB camera work by capturing the photons reflected or emitted by objects. [17].

Thermal cameras also measure electromagnetic radiation, but at a much higher wavelength, between 0.7 and 1000  $\mu\text{m}$ . Objects above 0 K emits radiation relative to its temperature within this spectrum. Within the infrared spectrum there are five sub-bands, see Table 2.2. Observe on Figure 2.3 that the emission from the body in NIR and SWIR sub-bands is nearly zero, which requires illumination at the appropriate wavelengths. Due to these spectrums different thermal sensors exist that focus on different sub-bands. Night-vision cameras usually operate in the NIR or SWIR-range with an active illumination source, where what is referred as thermal infrared (TIR) cameras operate in the MWIR and LWIR range, where the black body peaks in this range, see Figure 2.3. Within this range the cameras are able to measure direct emission from objects with temperatures between 190 – 1000 K which emits radiation in the TIR spectral range.

How much thermal radiation an object emits is decided by two factors: its temperature and emissivity. The perfect emitters are called black bodies, where their emissivity value is 1. Most materials are called grey bodies which have an emissivity between 0 and 1. Both human skin and cloth has an emissivity very close to 1, which means a person's appearance in the thermal image is highly correlated to its temperature. Other high emissivity objects can therefore cause troubles with detection of persons, but since the focus for the project is inside a sports arena it is not expected to be any issue regarding objects with temperatures

and emissivity in the same level as persons.

Table 2.2: Infrared sub-division

Name	Abbreviation	Wavelength [ $\mu\text{m}$ ]
Near-infrared	NIR	0.7-1.4
Short-wavelength infrared	SWIR	1.4-3
Mid-wavelength infrared	MWIR	3-8
Long-wavelength infrared	LWIR	8-15
Far-infrared	FIR	15-1000

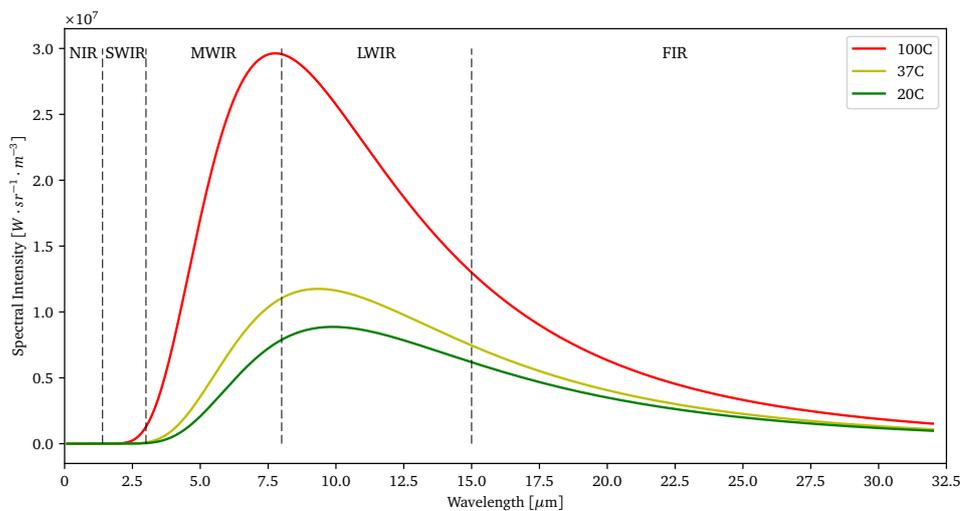


Figure 2.3: Spectral radiance emitted for black body objects at different temperatures

The camera used for the project is *AXIS Q1922 10 mm*[18], thermal camera which have a resolution of 640x480 pixels and can record up to 30 fps. The camera creates a webserver, where it is possible to connect and access to properties of the camera. It is possible to make the camera run on a pre-defined date and time, sync with the computer or with a Network Time Protocol server. It is possible to create a customised recording profile, which gives the opportunity to change the fps and the codec of the recordings, it have been chosen to record with 25 fps, due to the accelerometers being incapable of having a sample rate of 30, but it is possible to pick 25 Hz. So the camera and accelerometer can have the same sample rate, which makes it easier to compare data. The camera view can be seen at Figure 2.4.

### 2.2.3 From Position to Acceleration

Since the project involves correlate video data with accelerometer data it is necessary to make the two data types similar.

It is assumed that the initial positions of the players and the orientation of the accelerometers are unknown it is impossible to create a trajectory from the accelerometer data. Because of this it is chosen to estimate the acceleration from the tracklets.



Figure 2.4: An example of the AXIS Q1922 camera view

Since the tracklets presents a position of the person on the field for each frame it is possible to calculate the distance which the tracked person has moved. This distance is calculated using the Euclidean distance Equation (2.2).

$$dist = \sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2} \quad (2.2)$$

Where  $dist$  is the distance between the two points the person have travelled in timestep,  $t$ . The two points,  $p_t$  and  $p_{t-1}$ , consist each of an X- and Y-value.

With the distance from one frame to the next calculated, the speed which the player has moved with can be calculated as seen in Equation (2.3).

$$v = \frac{dist}{\Delta t} \quad (2.3)$$

Where  $v$  is the velocity ( $\frac{m}{s}$ ) and  $\Delta t$  is the time in seconds between frames. Since the video is recorded using 25 fps, the  $\Delta t$  is in this case  $1/25$  seconds.

After the velocity have been calculated it is now possible to calculate the acceleration for the person.

$$a = \frac{\Delta v}{\Delta t} \quad (2.4)$$

Where  $a$  is the acceleration ( $\frac{m}{s^2}$ ),  $\Delta v$  is the difference in velocity and  $\Delta t$  is the time, between the time steps.

The acceleration is given by  $m/s^2$  where the accelerometer measures acceleration in g, which as explained in section 2.2.1 where 1 g corresponds to  $9.82 \frac{m}{s^2}$ . In order to have the same unit the estimated acceleration is divided by 9.82.

### 2.2.4 Time-Sync Between Devices

As mentioned section 2.2.1, the accelerometer is not synchronised with each other and therefore it is not synchronised with the video either. It is necessary to make sure both are in synchronised else it will be impossible to correlate any data.

There are different methods to synchronise the devices. One method is to make the players perform a distinct movement on the camera, such as a jump and then evaluate the acceleration of all the accelerometers to find the jump. After analysing the output of the accelerometer when jumping, it is found difficult to locate when the person jumps and when the person is landing. Another method is to calculate the acceleration for each person's movement in the video feed and match with the acceleration from the accelerometers. In Figure 2.5 the acceleration for the video have been estimated and the acceleration measured by the accelerometer is seen.

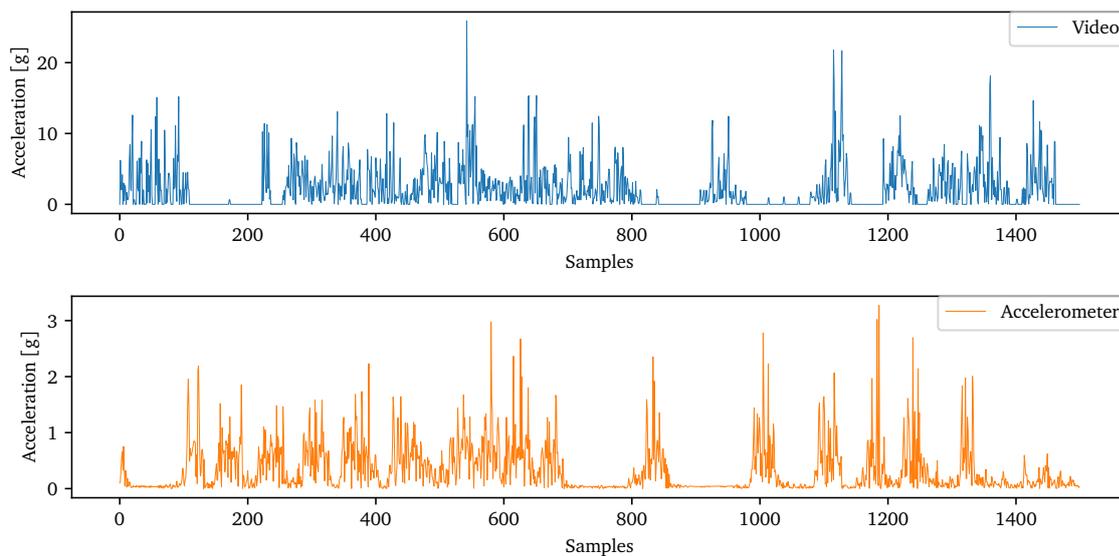


Figure 2.5: In the top graph the estimated acceleration from the video is seen and in the bottom graph the measured acceleration from the accelerometer is seen. The camera and accelerometer is set to start at same time, but it can be seen the two graphs resembles each other, but camera has started recording before the accelerometer.

In order to overcome the asynchronous time issue an mathematical method is tested from Koutra et al.[19], the paper states methods such as Dot Product or Russell-Rao can be used to match. To calculate at which data sample the measured acceleration fits the estimated acceleration best, it have been chosen to use the dot product to calculate the similarity of the two vectors.

Consider vector  $a$  being the estimated acceleration, vector  $b$  being the measured acceleration data and  $\theta$  being the angle between the vectors in Equation (2.5). The

calculation of dot product for these two vectors is here shown.

$$a \bullet b = \sum_{i=1}^n \|a_i\| \cdot \|b_i\| \cdot \cos(\theta) \quad (2.5)$$

Consider these two vectors as time series. The correlation is then done by applying a time coefficient  $\tau$  to one of the time series as seen in Equation (2.6).

$$r_i = \sum_{i=1}^n \|a_i\| \cdot \|b_{i+\tau}\| \cdot \cos(\theta) \quad (2.6)$$

Where  $\tau = 0, 1, 2, 3, ..n$  and  $r_i$  is resulting vector of the correlation. The results of the correlation can be seen in Figure 2.6

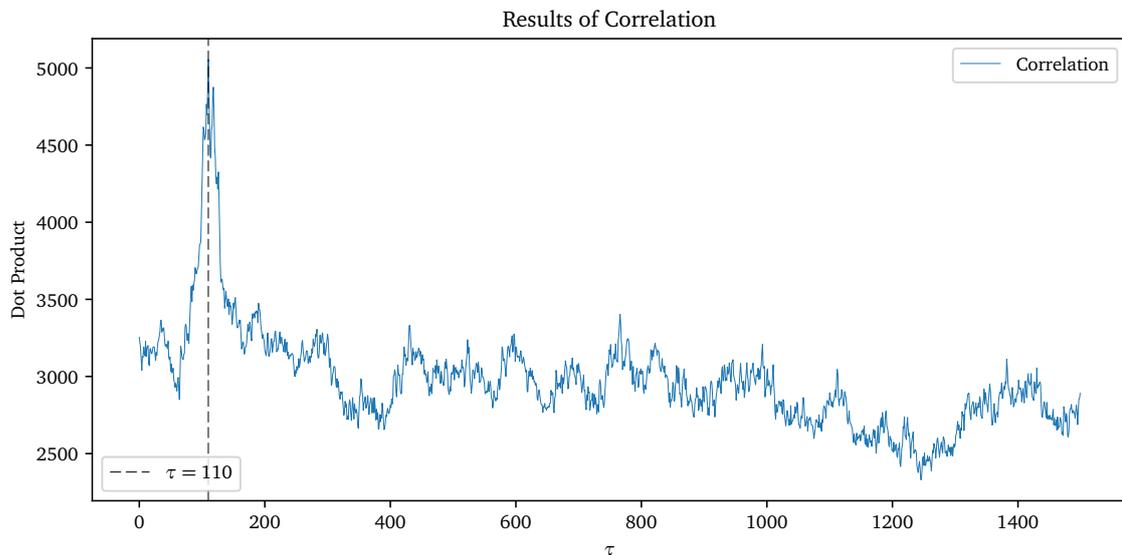


Figure 2.6: This graph represent the correlation result over the two vectors. The dashed black line represent the maximum, which is at  $\tau = 110$ . The maximum represent the  $\tau$  where the two vectors resemble each other the most.

By calculating a dot product maximisation between the two vectors it is possible to synchronise the two time series. As it is seen in Figure 2.6, the two vectors resembles best by shifting the video acceleration vector with 110 samples. This can be seen in Figure 2.7.

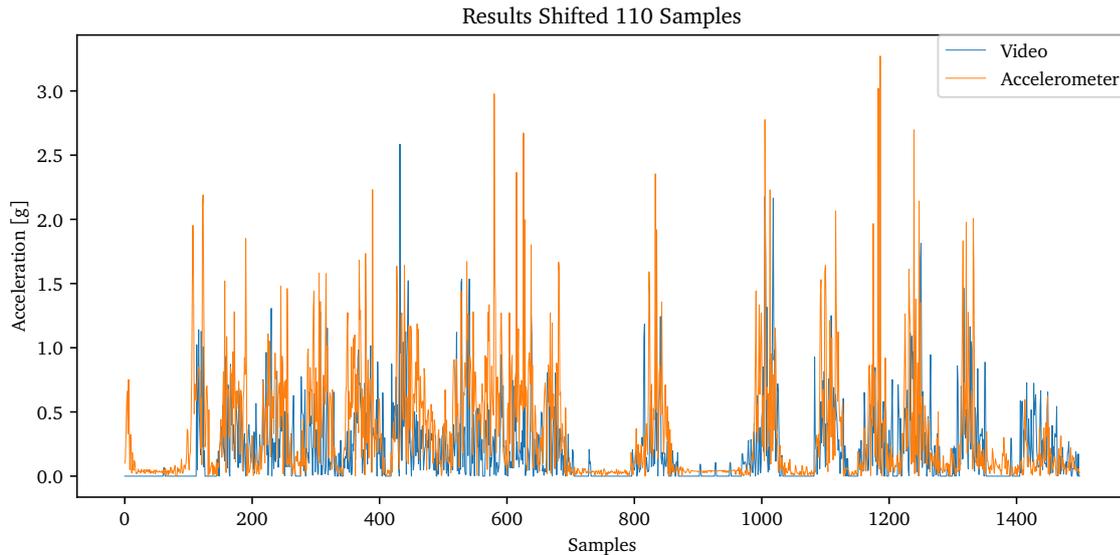


Figure 2.7: In this graph the two data vectors are shown. The estimated acceleration has in this case been scaled for visual purposes. The first 110 data points from the estimated acceleration has been removed to synchronise the two vectors. As it can be seen the two vectors now fit each other in the periods where no movement is happening.

## 2.3 Data Gathering

Since previous work and data gathering conducted by Aalborg University on the sports arenas has consisted on data without inertia measuring there was currently no recorded data, which could be used for this project. Data used in this project was therefore recorded by the authors. The first data recording conducted was used in the preliminary test seen in section 2.2.4. This data was recorded using a single person, with an accelerometer mounted on the waist, in an indoor environment at Aalborg University. The preliminary test was created in order to investigate the possibility of correlating the two data types. The recording was performed in a 3.5x3.5m area and lasted 5 min. The corners was marked during the recording, in order know the pixel position of each corner.

As the preliminary test showed positive results on a single player it was decided record a larger dataset with several players. The recording was conducted in Gigantium sport arena. A single thermal camera was placed 8.75 meters above the field positioned from the side. From this position the camera was able to capture half of an indoor handball field in which the test was conducted. A frame from the camera can be seen in Figure 2.8.



Figure 2.8: Viewpoint from camera positioned 8.75 m above the ground.

Before the test players entered the field a calibration was made. This was done by dividing the field into  $5 \times 5$  m. During the recording a person would then stand at each corner of these squares, the world coordinate and pixel position is then saved. The divided field can be seen in Figure 3.12. The total size of the field was  $15.5 \times 20$  m.

After measuring the corners, 9 people were equipped with an accelerometer each. The test then proceeded by having 5 players on the field at the time. One player would remain through the entire recording as this player is goalkeeping. The remaining 4 players were then split into two teams. Every 2 minutes two players leave the field and two new players enter to take their place. This was done in order to simulate a match where substitutions would occur. The recording lasted 12 min with a frame rate of 25 per second.

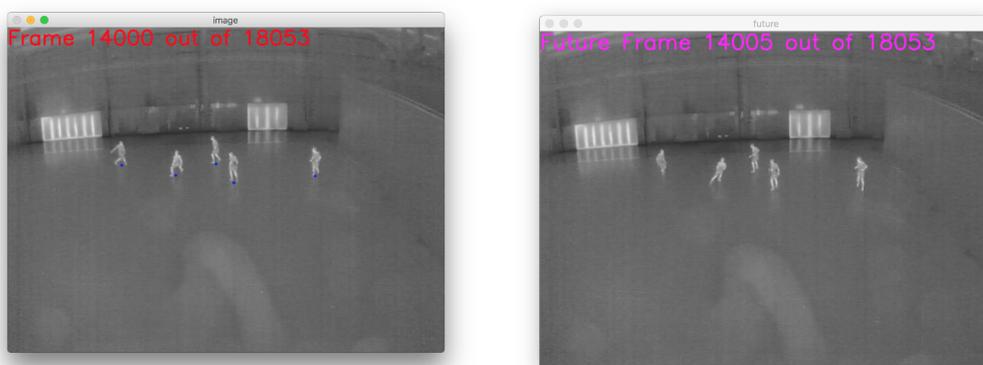
### 2.3.1 Ground Truth Creation

In order to correlate the accelerometer data and acceleration from the video and to later verify tests a ground truth is required. To obtain the ground truth information from the video a program has been developed. The program consists of presenting each frame of the video in chronological order. Each time a player is entering the scene it is then possible to mark their position by clicking on the pixel in the frame. Once the mark has been placed it is possible to move the mark using the arrow keys, in the case the mark is slightly miss placed.

When clicking "space"-button the mark is saved for the specific frame. Once saved the

program will automatically proceed to the next frame. The program can only take one mark per frame, so once a person has been marked he will be manually tracked until he leaves the field. If no mark is set, when all players in the frame is already marked, space can be used to continue to the next frame without saving any information.

By using this program to track each person entering the scene till they leave the scene, it has been possible to create a ground truth set for each player. Additionally to add the feature of foresee if a still standing player will begin to move soon an additional image is shown which can be manually adjusted to show x amount of frames in the future or past of the current frame, this is done to make the ground truth data more accurate. The program also presents the marks which has already been placed for the other players which are in the scene. An example of the ground truth program can be seen in Figure 2.9. The ground truth collection resulted in 42 trajectories of the players over 12 minutes of recording. These images also displays how the position of the players has been done. By placing the mark at between the feet of the players in an attempt to mark the position of the players.



(a) Current frame.

(b) Future frame.

Figure 2.9: **(a)** Displays the current image to be marked. **(b)** Displays the future image 5 frames ahead. As seen in (a), all players have been marked, seen at the blue dots at their feet. If the image were clicked a green dot would appear in that position, representing a new player.

### 2.3.2 Data Definition

In this section the different data recorded through this project will be described and named for further references. All trajectories described in this section is the ground truth data, created as seen in previous section 2.3.1.

**Dataset 1** *Contains the recording conducted at Aalborg University, where a single person wearing a single accelerometer, was running around for 5 minutes in a square of 3.5x3.5m.*

*(a) Contains a single trajectory for the person.*

**Dataset 2** *This dataset contain recordings conducted in Gigantium sports arena. The video contain 12 minutes and 3 seconds video recorded in 25 fps (a total of 18075 frames).*

*In the dataset people play football, normally five people are on the field where one of them is the goalkeeper. Approximate every 2 minute two people gets substituted with two new people. A total of 9 different people enter or leave the field during the recording. Since people are not staying on the field the entire time, there have been created two different sub-datasets.*

- (a) This sub-dataset contains 9 trajectories corresponding to the full trajectories of each person recorded during the test.*
- (b) This sub-dataset contains 42 trajectories. Each trajectory presents the position of a player from the frame where he enters the field and until he leaves the field.*

## 2.4 Summary of Technical Analysis

In the previous work, section 2.1, conducted at Aalborg University it was found that tracking in a sport environment resulted in many tracklets, due to the vast amount of occlusion occurring in sports. It has been found that identifying the individual players after an occlusion is hard when using thermal cameras, where colour features are lost. In section 2.2, the devices used in this project have been explained. Research conducted on the accelerometers have shown that the accelerometers clocks are unsynchronised, despite being started at same time. In section 2.2.3 the conversion from video information to acceleration have been presented, which in section 2.2.4 led to a preliminary test of matching the video information with the data from the accelerometer. The test showed that the accelerometer and camera was not synchronised either. A method based on dot product maximisation was presented for synchronising the accelerometer data with the video. In section 2.3 the data gathered and labelled in this project is presented.

## 2.5 Problem Statement

Based on the analysis of the initial problem statement section 1.2 and the summary of the Technical Analysis, chapter 2 the following problem statement is formulated:

*How can a system be developed to connect broken tracklets into trajectories, using thermal video- and accelerometer-data?*

## 2.6 Specifications and Requirements

In the following Table 2.3, the specification and requirements will be listed for the project.

Table 2.3: Specification and requirements

Req. #	Description	Specification	Reference
1)	Detection	Must be able to segment persons from the background.	section 2.1.1
2)	Detection	Must be able to detect all persons on the field.	section 2.1.1
3)	Tracking	Must be able to track a single person.	section 2.1
4)	Tracking	Must be able to detect when an occlusion occurs.	section 1.2
5)	Tracking	Must be able to detect when an occlusion splits into individuals.	section 1.2
6)	Tracking	Must be able to detect when an occlusion splits into a new occlusion.	section 1.2
7)	Dataprocessing	Must be able to calculate the world position of each person based on the pixel position of the detected person.	section 2.2.3
8)	Dataprocessing	Must be able to calculate the acceleration based on world positions.	section 2.2.3
9)	Dataprocessing	Must be able to synchronise accelerometer data with video recording.	section 2.2.4
10)	Dataprocessing	Must be able to match trackless/trajectories with accelerometers	section 2.2.4

# Design and Implementation

---

# 3

In this chapter the different parts of the systems are described and designed. First there will be an introduction to the system illustrated with a flowchart.

## 3.1 Description of the System

The overall system Figure 3.1 is a setup of a thermal camera and accelerometers for each person who are on a field. The accelerometers output the acceleration in each axis and the camera output a thermal image, where the persons on the field stands out to its background. From this the pixel positions can be extracted from each person and these can be converted to world coordinates of the field. The world coordinates are then used to estimate the acceleration from one image to another. It is then possible to match the estimated acceleration from the video with the measured acceleration from the accelerometers and then find the best match between the devices.

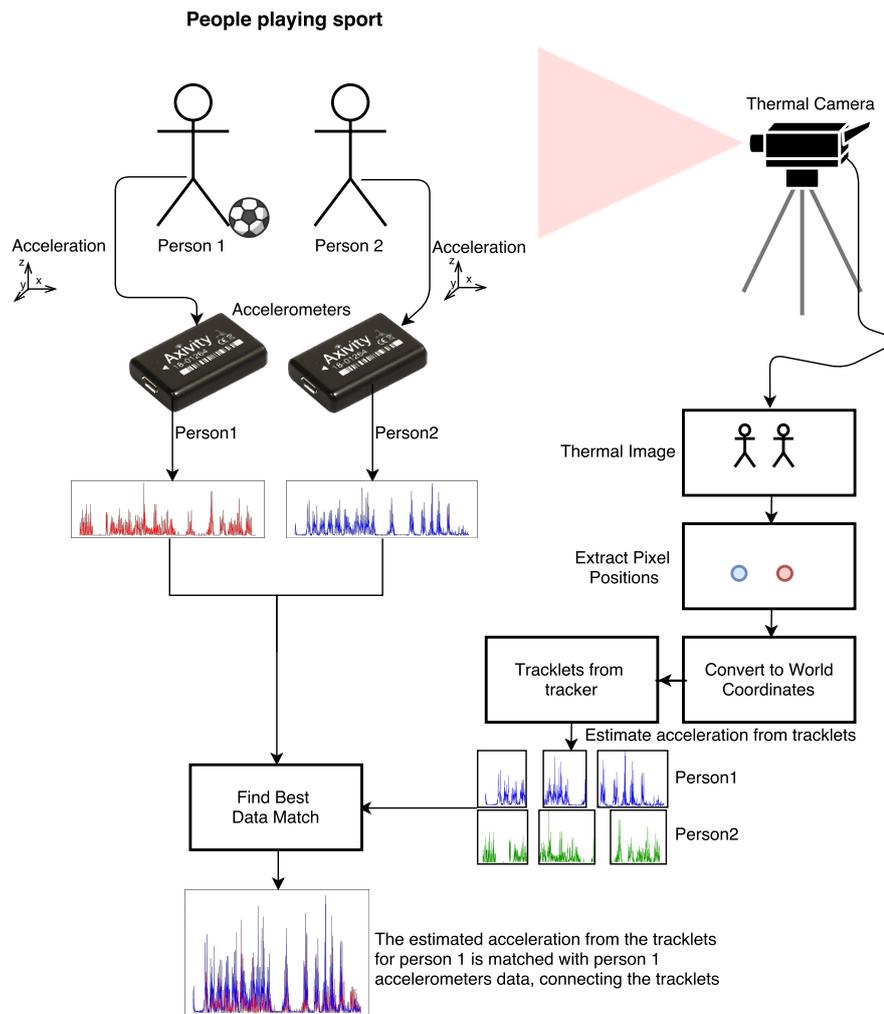


Figure 3.1: In this flowchart it is seen how the full pipeline is designed. The chart is read from the left where the acceleration measurements are seen and to the right the information provided by the camera is seen.

## 3.2 Detection and Tracking

In section 2.3.1 it was shown how ground truth player trajectories could be created using a simple program. However the process of creating trajectories for each player occurring during the video is a time consuming task, and a natural step is therefore to implement a simple detection and tracking method. By having a automatic tracking system removes the need of manually track each player in a potential finished product.

### 3.2.1 Detection

First step of the tracking is to detect the players in each frame. The detection step can be seen in Figure 3.2. Before the detection can be applied it is necessary to extract the foreground, to do several different background subtraction techniques exist. Due to previous experience from the group, Mixture of Gaussians[20], (MOG) algorithm have been applied from OpenCV's library. A series of 300 frames of the background, without any players on the field has been used to train the background subtractor. MOG has been used to accommodate potential changes, due to noise in the thermal camera sensor or changes in temperatures for the background.

A median filter is applied to each frame to reduce potential noise. Once trained, the background image is applied to each frame in the video in order to determine the foreground. Once the foreground is extracted the image is thresholded. The threshold in the grey scaled image has been found experimentally and is set to 117. After the threshold have been applied the contours of each player stands out.

To remove potential noise, the minimum size of a contour belonging to a person is found. If a contour is not equal or greater to the minimum size it is either discarded or saved. The minimum size threshold for a contour has likewise been experimentally found. Since the contour size of a player varies based on their position in the field the size is set such that the players furthest from the camera is minimum requirement to be accepted.

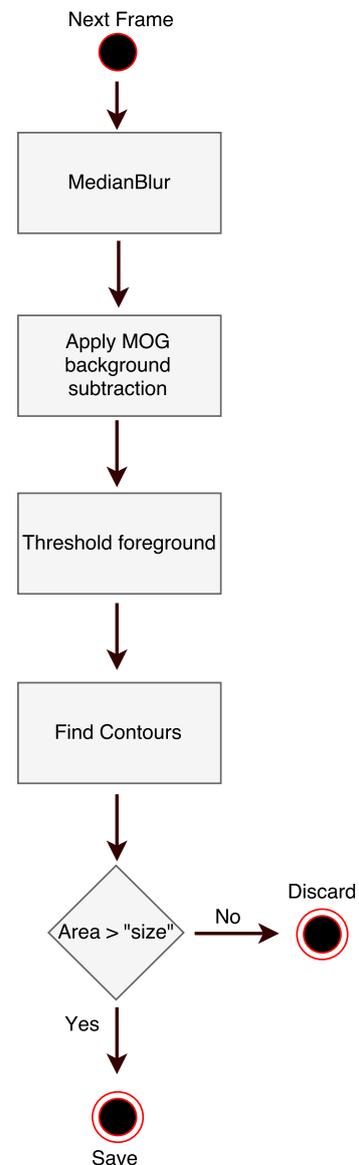


Figure 3.2: Flow chart of detection method used to detect the players on the field. Each contour found is verified by the size of the contour.

### Mixture of Gaussian

As mentioned for the background subtraction a mixture of Gaussian (MOG) have been used, the method is made by Zivkovic [20]. The MOG method is modelling each pixel as a mixture of Gaussians and it use online approximation to update the model. For the method to work it is assumed that every pixel's intensity in the video can be modelled using a Gaussian mixture model. The special about Zoran's MOG version is that it is able to select the appropriate number of Gaussian distributions for each pixel compared to the traditional which have a predefined  $K$  amount of Gaussian distributions. This provides better adaptability for varying scenes such as noise hitting the thermal camera sensor which in an image is illustrated similar to illumination changes.

### Contour Detection

For this project the contour detection method is based on Suzuki [21]. Suzuki proposed a method to find the boundaries by using a kernel seen in Figure 3.3. In this case  $i$  corresponds to the row number and  $j$  is the column number.

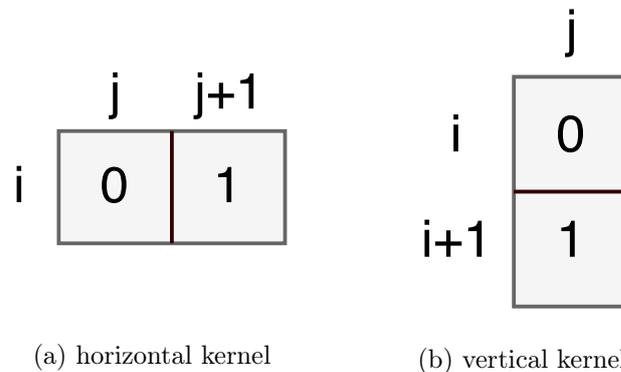
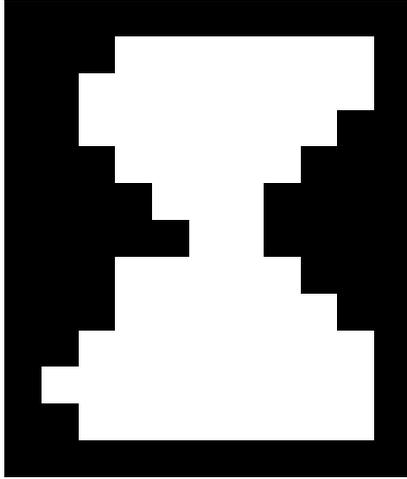


Figure 3.3: By applying the horizontal kernel in each row of the image the outer borders are found, when both conditions in the kernel are fulfilled, a leftmost border is found. If only  $j + 1$  condition is fulfilled, the kernel is currently within a contour. When none of the conditions are fulfilled the kernel has reached the rightmost border in that row. The vertical kernel is likewise applied to the binary image.

In Figure 3.4 an example of how a contour is found by applying the kernels from Figure 3.3 can be seen. In this figure each value corresponds to a pixel in a binary image.



(a) Input to contour kernels.

0	0	0	0	0	0	0	0	0	0	0
0	0	0	2	2	2	2	2	2	:	0
0	0	2	1	1	1	1	1	1	:	0
0	0	2	1	1	1	1	1	:	0	0
0	0	0	2	1	1	1	:	0	0	0
0	0	0	0	2	1	:	0	0	0	0
0	0	0	0	0	2	:	0	0	0	0
0	0	0	2	2	1	1	2	0	0	0
0	0	0	2	1	1	1	1	2	0	0
0	0	2	1	1	1	1	1	1	:	0
0	2	1	1	1	1	1	1	1	:	0
0	0	2	:	:	:	:	:	:	:	0
0	0	0	0	0	0	0	0	0	0	0

(b) Output of contour kernels.

Figure 3.4: The pixel values with 0 corresponds to the points where only the  $i,j$  element is fulfilled. The pixel values with 2 corresponds to where both condition of the kernel is fulfilled and indicates the top and left borders of the contour. The pixel values 1 indicates the filling of the contours. The pixels with : corresponds to the right and bottom most contour borders, meaning that 0 of the conditions for the kernels are fulfilled.

As it can be seen in Figure 3.4 the contours has been found.

### 3.2.2 Tracking

When all contours has been found by the detector in each frame, a tracking method based on contours from previous frame to next frame has been implemented. This tracking method keeps the trajectories of all players detected and tracked throughout the video. A few terminology expressions is explained in order to improve the understanding. An occlusion is defined as a contour containing more than a single person. A split is when an occlusion is separates into two or more contours. A flowchart of the tracking method can be seen in Figure 3.5.

The input to the tracking algorithm are a list of *NewContours* for the current frame and a list of *TrackedContours* from the previous frames. It is then first determined if any persons is present in the current frame by checking the amount of *NewContours*. If any new contour is present, it is matched with the *TrackedContours* to check if the new contour is corresponding to a tracked contour. If the new contour is not corresponding to any tracked contours, a new ID is created. If the new contour is corresponding to a tracked contour it is checked if multiple tracked contours are part of the new contour. If this is the case the new contours is added to the *OcclusionList* along with the tracked contours.

If this is not the case it is checked if multiple of the new contours are matched with a tracked contour. If this is the case the new contours is added to the *SplitList* along with the tracked contour.

If the new contour is not in the *SplitList* or *OcclusionList* it is added to the *SingleList* containing the new contours matched with the corresponding tracked contours. From the

*SingleList* the tracked contours are updated with new information contained in the new contours, such as position and number of persons.

To handle the occlusions and splits occurring in a frame it is first checked if any elements in the *SplitList* are matching with the *OcclusionList*. In the case no elements are matching new IDs are created for the splits and occlusions. If elements are found in both the *OcclusionList* and *SplitList*, the *SplitToOcclusion* function will be used, see Code snippet 3.1 for pseudo code, explaining the function.

```

1   for split in splitlist:
2       if split in occlusionList:
3           createNewPlayer from occlusion
4           remove split from splitlist
5           remove occlusion from occlusionlist
6   if splitList:
7       createNewPlayer from splitList
8   if occlusionList:
9       createNewPlayer from occlusionList

```

Code snippet 3.1: Pseudo code for splitToOcclusion-function

By generating a binary contour image of each tracked contour and each new contour it is possible to find the similarities between two contours by multiplying the images together.

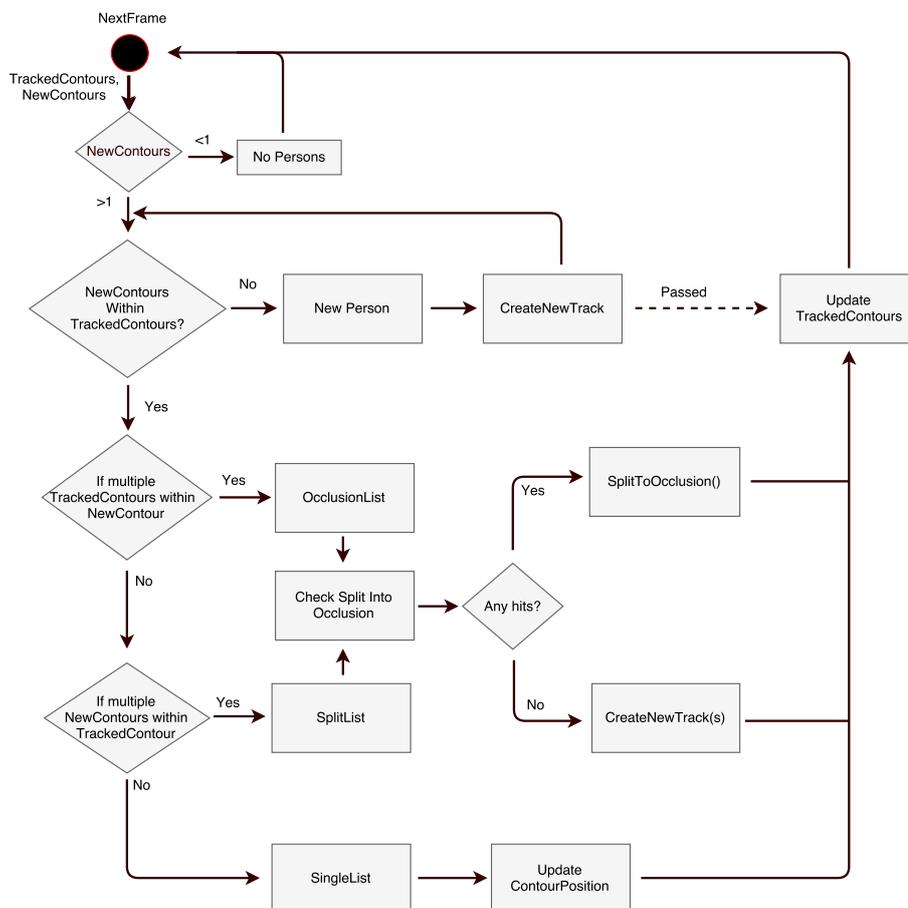


Figure 3.5: Presents the process of tracking each player found by the detection algorithm.

The number of white pixel presented in the resulting images correspond to how many pixels in the previous contour is also present in the new contour. Since we are working with 25 fps it is considered impossible to move outside your contour in the previous frame. An example of this can be seen in Figure 3.6.

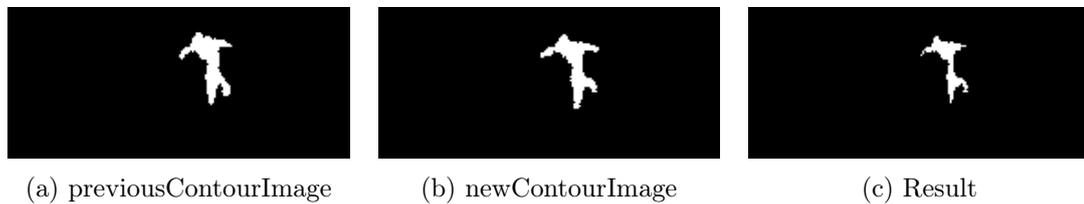


Figure 3.6: **(a)** Image from tracked contour. **(b)** A matching contour from the new contours. **(c)** The result of multiplying the images together. As it can be seen the result display the contour pixels which are equal from the previous to the next frame.

In Figure 3.6 an example of single previous contour tracked to the next contour of same person. In Figure 3.7 and example of a person going from being a single person into an occlusion. Notice how two previous contour images share the same new contour image and therefore share the same position.

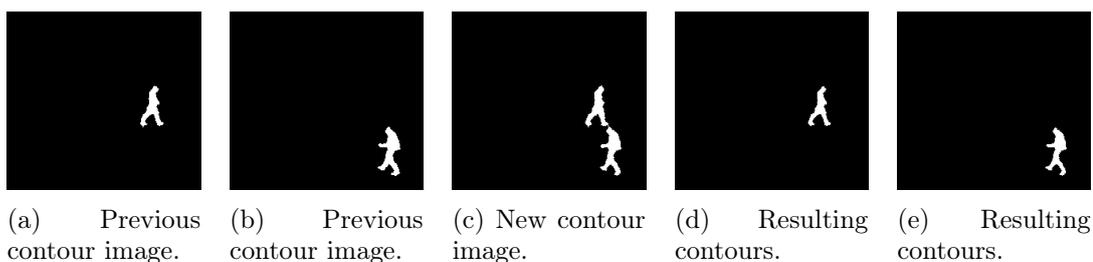


Figure 3.7: **(a)** and **(b)** previous tracked contours. **(c)** contour in new frame. **(d)** and **(e)** resulting images after multiplying each of the previous contours with the new contour. As it can be seen from this example it can be determined that the two previous contours is entering an occlusion with each other.

As seen in Figure 3.7 it is possible to determine when two individuals are entering an occlusion with each other. Likewise will Figure 3.8 show how it is determined if an occlusion is splitting into two contours.

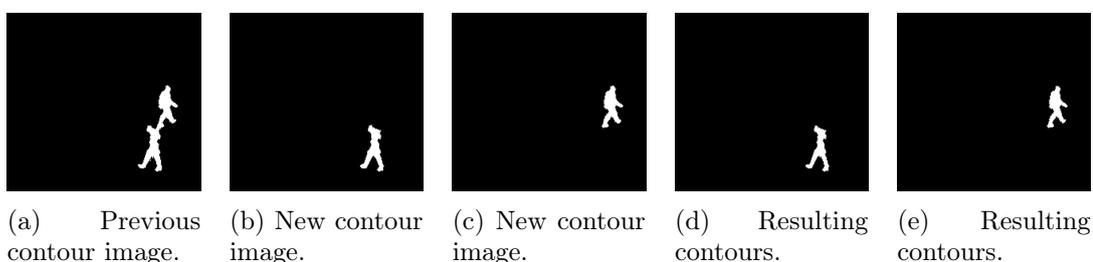


Figure 3.8: **(a)** previous tracked contour. **(b)** and **(c)** contours in new frame. **(d)** and **(e)** resulting images after multiplying each of the previous contours with the new contour. As it can be seen from this example it can be determined that the previous contours is splitting into two individual players.

With Figure 3.8 explaining how the splitting of an occlusion is handled it concludes how the tracking method is working. Next section will explain how the tracking information obtained is saved.

### 3.2.3 ID Class

In order to further use the information found from the detection and tracking method, a Python class has been created. This class was created to save various information found by the tracking algorithm. In Table 3.1 the information saved for each tracked contour can be found.

Table 3.1: The python class created to control the data obtained from the tracking method.

ID Class and description of indices	
IDname	This saves the number of a tracked contour, corresponding to when it entered the frame.
contour	Contains the last updated contour.
WorldX	Contains the last updated world coordinate X.
WorldY	Contains the last updated world coordinate Y.
PixelX	Contains the last updated pixel coordinate X.
PixelY	Contains the last updated pixel coordinate Y.
WorldHistory	Contains a history of all world positions where the tracked contour has been.
PixelHistory	Contains a history of all pixel positions where the tracked contour has been.
PeopleCount	Contains the number of players within this contour.
Parents	Contains a list of previous ContoursID which the tracked contour have been part of This is mainly used during occlusions and splits.
Frames	Contains a list of frames in which the ID has been in the frame.

### 3.2.4 Example of Tracking

To follow up on how the tracker works, Figure 3.5, and how it handles splits and occlusion, Figures 3.6 to 3.8 shows examples from the images in Dataset 2. In these example images the contours containing a single person is coloured green and contours containing more than one person, also known as an occlusion, are coloured red. First example, see Figure 3.9, shows how the system handles when two persons walk close enough for their contours to collide which the system interpret as an occlusion. It stops the tracking of the two colliding persons (*ID52* and *ID53*), Figure 3.9a and starts tracking of the occlusion (*ID54*), Figure 3.9b. Later in Figure 3.9c the occlusion splits again and it stops tracking of *ID54* and begins to track two new IDs, one for each person (*ID55* and *ID56*), Figure 3.9c. The reason it dedicate new ID's is the that it is a challenging task to distinguish which of the persons belonged to which ID pre-occlusion.

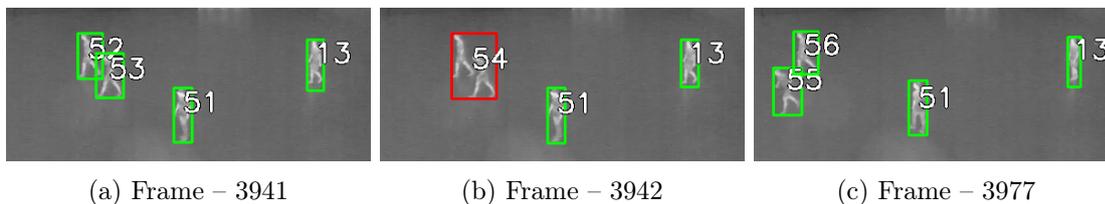


Figure 3.9: Each image is cropped and shows a bounding box around each contour and the text indicates the contours corresponding ID. **(a)** 4 persons, 4 contours. **(b)** after an occlusion happened between *ID52* and *ID53*, now called *ID54*. **(c)** after 34 frames *ID54*, splits into two new contours which create *ID55* and *ID56*.

In Figure 3.10 a more chaotic scenario is happening, where two occlusions ( $ID8$  and  $ID9$ ), Figure 3.10a are present and in next frame both occlusions splits at the same time. The figure shows that the system can handle these scenarios and give a new ID to each contour ( $10$ ,  $11$ ,  $12$  and  $13$ ), Figure 3.10b.

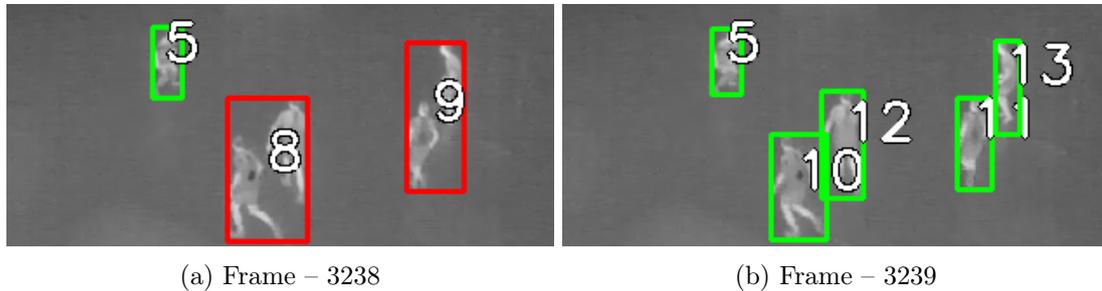


Figure 3.10: Each image is cropped and shows a bounding box around each contour and the text indicates the contours corresponding ID. **(a)** consist of two occlusions  $ID8$  and  $ID9$  with two people in each. **(b)** next frame where both occlusions splits into four persons occlusion  $ID8$  becomes  $ID10$  and  $ID12$  and occlusion  $ID9$  becomes  $ID11$  and  $ID13$ .

At Figure 3.11 is another special scenario, where an occlusion between two persons exist ( $ID49$ ), Figure 3.11a with another person ( $ID46$ ) close to the occlusion. In the next frame the occlusion splits, and  $ID46$  collide with the upper person in the occlusion resulting in a new occlusion. The program is still able to detect this scenario, and the information saved for the new occlusion ( $ID50$ ) is that it consist of two persons and  $ID51$ , Figure 3.11b, consist of one person.

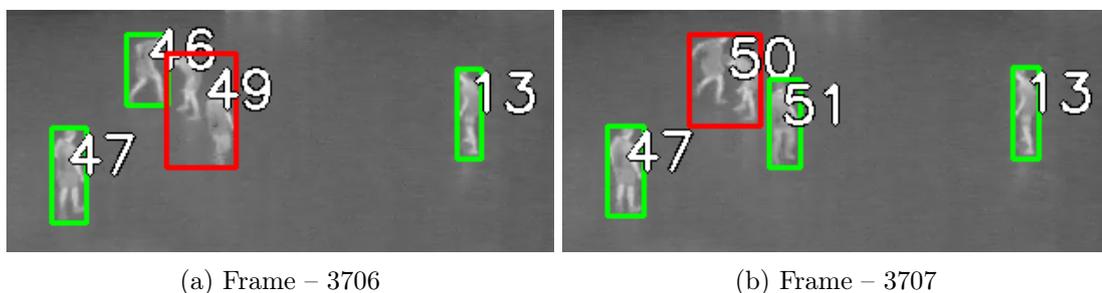


Figure 3.11: Each image is cropped and shows a bounding box around each contour and the text indicates the contours corresponding ID. **(a)** consist of one occlusions  $ID49$  with one person  $ID46$  close to the occlusion. **(b)** next frame the lower person in the previous occlusion splits and becomes  $ID51$  but the upper person in the previous occlusions gets close enough to  $ID46$ , and occlude into  $ID50$

### 3.3 From Pixel to World Coordinates

In this section the procedure which is required to go from the camera's pixel coordinates to real world coordinates will be described.

First it is necessary to calibrate the field, this is mentioned in section 2.3, but to summarise it is done by using measuring tape and then at every 5 meter from top left a person will stand. This makes it possible to extract pixel coordinates for the same place. The process is done for the entire field for Dataset 2 on 15.5m x 20m with a total of 25 measurement points. As mentioned every 5 meter in both x- and y-axis have been measured which makes it possible to generate tiles as mentioned by Gade et al.[22]. The idea with the tiles is when a person enters a tile, the calibration for this tile will be used and further when a person exits and enter a new tile, the calibration will be changes which makes the pixel to world process more precise.

The field is measured and the corresponding pixel locations is found, see Figure 3.12. With four known points it is possible to make a transformation matrix for each tile in the field, this can be done by using OpenCV's `getPerspectiveTransform` function[23], which takes four source and destination points as input and then it returns the transformation matrix for these points. It is also possible use `findHomography` from OpenCV, where the `getPerspectiveTransform` is the base for `findHomography`, the great thing with `findHomography` is the fact that it can take more than four points and it use a method called RANSAC which is able to reject outliers if at least  $50\% + 1$  of the data points is okay.

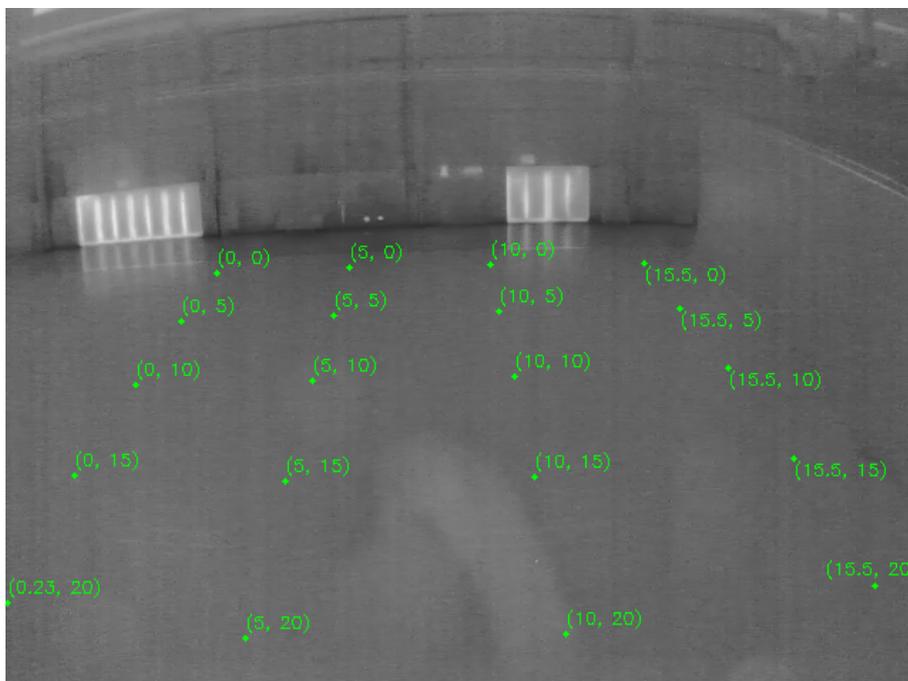


Figure 3.12: The image is showing the calibrated points on the field and their corresponding real world coordinates, one tile is consisting of a square of four points.

Since only four points have been measured for each tile and the correct pixel locations for the same points are extracted it is chosen to use the base-method `getPerspectiveTransform`.

The code for generating a list containing each tile in the field can be seen in Figure 3.13.

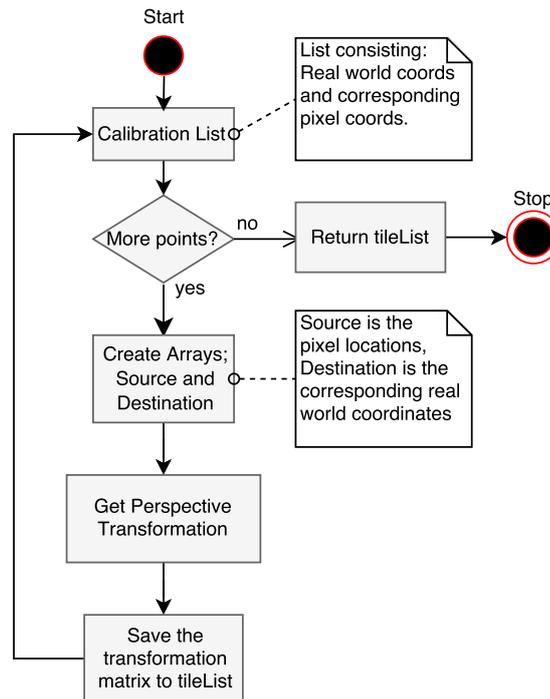


Figure 3.13: Flowchart is explaining how the system calibration is done in the system.

With the tileList created it is now possible to go from pixel to world coordinate, but first it is necessary to know which tile the pixel belongs to, see Figure 3.14. The figure explains how this is done by creating a polygon from each tile, by using their four corner coordinates in the image. Then you can control if the pixel is within the polygons boundary or outside, if the pixel is within, the correct tile is found and then the transformation matrix for this tile can be used to calculate the pixel to world conversion. If it is not within, the program creates the next tile polygon, until there are no more tiles left to be checked. If the pixel is not within the tile yet, the program will find the tile with smallest distance to the pixel which will be used to calculate from pixel to world coordinates. Another option is to discard if the person is outside the tiles, but it have been decided not to and instead take it into consideration.

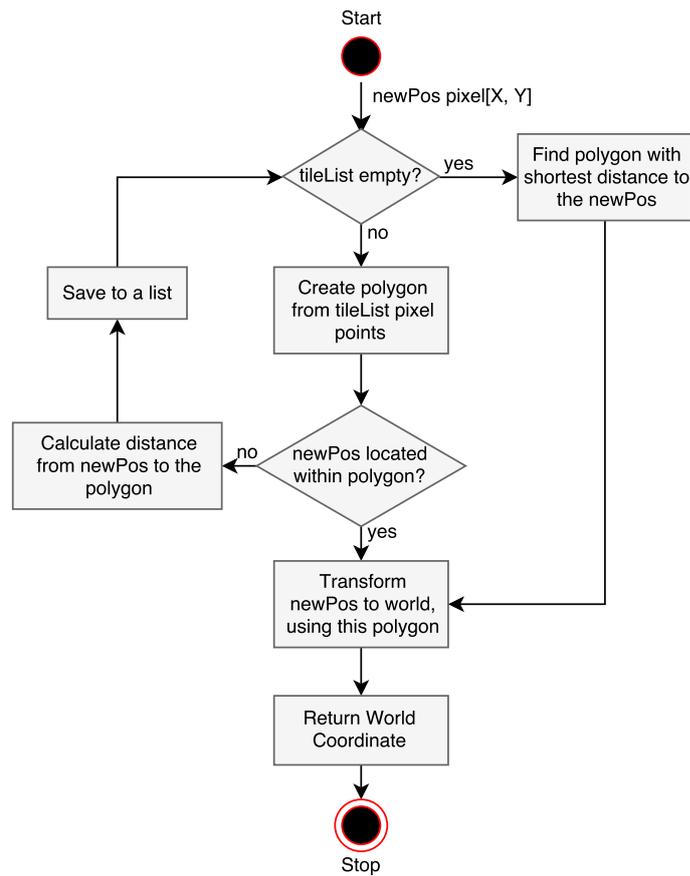


Figure 3.14: Flowchart telling how the system decide which of the calibrated tiles from Figure 3.13 it needs to use to calculate the corresponding world coordinate.

### 3.3.1 Perspective Transformation

Once the tile have been located the following explains the process in going from pixel- to world-coordinates.

To transform four points  $(x_i, y_i)$  to the four points  $(u_i, v_i)$  for  $i = 0, 1, 2, 3$  we use a perspective transform of the form seen at Equations (3.1) and (3.2), the method is based on [17].

$$u_i = \frac{a_0x_i + a_1y_i + a_2}{c_0x_i + c_1y_i + 1} \quad (3.1)$$

$$v_i = \frac{b_0x_i + b_1y_i + b_2}{c_0x_i + c_1y_i + 1} \quad (3.2)$$

This gives eight unknown coefficients  $a_0, a_1, a_2, b_0, b_1, b_2, c_0, c_1$  which can be calculated by

solving the linear system, Equation (3.3).

$$\begin{bmatrix} x_0 & y_0 & 1 & 0 & 0 & 0 & -x_0u_0 & -y_0u_0 \\ x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1u_1 & -y_1u_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2u_2 & -y_2u_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3u_3 & -y_3u_3 \\ 0 & 0 & 0 & x_0 & y_0 & 1 & -x_0v_0 & -y_0v_0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1v_1 & -y_1v_1 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2v_2 & -y_2v_2 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -x_3v_3 & -y_3v_3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ b_0 \\ b_1 \\ b_2 \\ c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ v_0 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} \quad (3.3)$$

To test whether this is correct, an example from the upper left tile will be used, see Equations (3.4) to (3.11).

$$(x_0, y_0) = (148, 187) \quad (3.4)$$

$$(x_1, y_1) = (241, 183) \quad (3.5)$$

$$(x_2, y_2) = (230, 217) \quad (3.6)$$

$$(x_3, y_3) = (123, 221) \quad (3.7)$$

$$(u_0, v_0) = (0, 0) \quad (3.8)$$

$$(u_1, v_1) = (5, 0) \quad (3.9)$$

$$(u_2, v_2) = (5, 5) \quad (3.10)$$

$$(u_3, v_3) = (0, 5) \quad (3.11)$$

The system of equations will then be, see Equations (3.12) to (3.19).

$$148a_0 + 187a_1 + a_2 = 0 \quad (3.12)$$

$$241a_0 + 183a_1 + a_2 - 1205c_0 - 915c_1 = 5 \quad (3.13)$$

$$230a_0 + 221a_1 + a_2 - 1150c_0 - 1085c_1 = 5 \quad (3.14)$$

$$123a_0 + 221a_1 + a_2 = 0 \quad (3.15)$$

$$148b_0 + 187b_1 + b_2 = 0 \quad (3.16)$$

$$241b_0 + 183b_1 + b_2 = 0 \quad (3.17)$$

$$230b_0 + 217b_1 + b_2 - 1150c_0 - 1085c_1 = 5 \quad (3.18)$$

$$123b_0 + 221b_1 + b_2 - 615c_0 - 1105c_1 = 5 \quad (3.19)$$

When solving this system the results for the coefficients can be seen at Equations (3.20) to (3.22).

$$a_0 = -0.40841, \quad a_1 = -0.30030, \quad a_2 = 116.60 \quad (3.20)$$

$$b_0 = -0.56061, \quad b_1 = -1.3034, \quad b_2 = 252.04 \quad (3.21)$$

$$c_0 = -0.0030251, \quad c_1 = -0.041678 \quad (3.22)$$

The required transformation from Equations (3.1) and (3.2) can be seen at Equations (3.23) and (3.24)

$$u = \frac{-0.40841x_i + -0.30030y_i + 116.60}{-0.0030251x_i + -0.041678y_i + 1} \quad (3.23)$$

$$v = \frac{-0.56061x_i + -1.3034y_i + 252.04}{-0.0030251x_i + -0.041678y_i + 1} \quad (3.24)$$

When inserting the pixel coordinates from the corners it can be confirmed that the transformation work correctly.

### 3.4 Synchronising Accelerometers and Video

In this section a test have been conducted on Dataset 2a, from section 2.3.2, recorded in Gigantium sports arena. The test is an extended version of the preliminary test which have been conducted in section 2.2.4. Where the goal was to test if dot product maximisation is a viable method for matching accelerometer data with estimated acceleration. This section presents the second preliminary test conducted in order to both test the dot product maximisation, but will also be used to synchronise the accelerometer and video data.

The position history extracted from the video of the players is used to estimate the acceleration for each trajectory as seen in section 2.2.3. For all players in the video, the acceleration will be calculated and a dot product will be found between the estimated acceleration and each measured acceleration from the accelerometers. As previous experience have shown in section 2.2.1 there is an issue with time synchronisation between the accelerometers, where the clock within the accelerometer is not synchronised and therefore start at separate real world times. Furthermore accelerometers are not synchronised with the video as experienced in section 2.2.4. Because of the synchronisation issue it has been decided to shift the video in negative and positive direction by inserting zeros as previously done in section 2.2.4. By performing dot product maximisation in both directions, it is possible to determine whether the accelerometers started recording before or after the camera and how many samples is needed to synchronise the data.

As explained in Dataset 2a the dataset consist 9 trajectories being the full trajectories of the 9 players recorded. The 9 players are through the video leaving the camera view when a substitution is happening or when the ball is leaving the field of view. The test will involve matching the 9 ground truth trajectories to the 9 accelerometers using dot product maximisation. Each trajectory is matched with all 9 accelerometers, both without shifting and shifting in negative and positive direction.

In Figures 3.15 and 3.16 estimated video acceleration for *person0* have been plotted, matched with *accelerometer0* and *accelerometer1*'s data. As seen *person0* enters the field three times during the entire video, seen by three spikes with nearly no activity between. When comparing Figures 3.15 and 3.16 it is noticeable that *accelerometer0*'s data contains the third spike, where *accelerometer1* is resting. By visually judging the two matches, it is seen that *accelerometer0*'s data matches the trajectory for *person0*. Comparing the calculated dot product, above each graph, it is seen that *accelerometer0* have a better match to the ground truth data than *accelerometer1*.

All results of the dot product maximisation for the 9 persons matched to all accelerometers can be seen in appendix A.1. The top matches for each person can be seen in Table 3.2. As seen five out of nine players are matched correctly, when no shift and positive shift is applied, but when negative shift is applied it matches six out of nine correctly. Notice that *person7* is matched to *accelerometer6*, when no shift and positive shift is applied and with its *accelerometer7*, when negative shift is applied. Also take note that the dot product

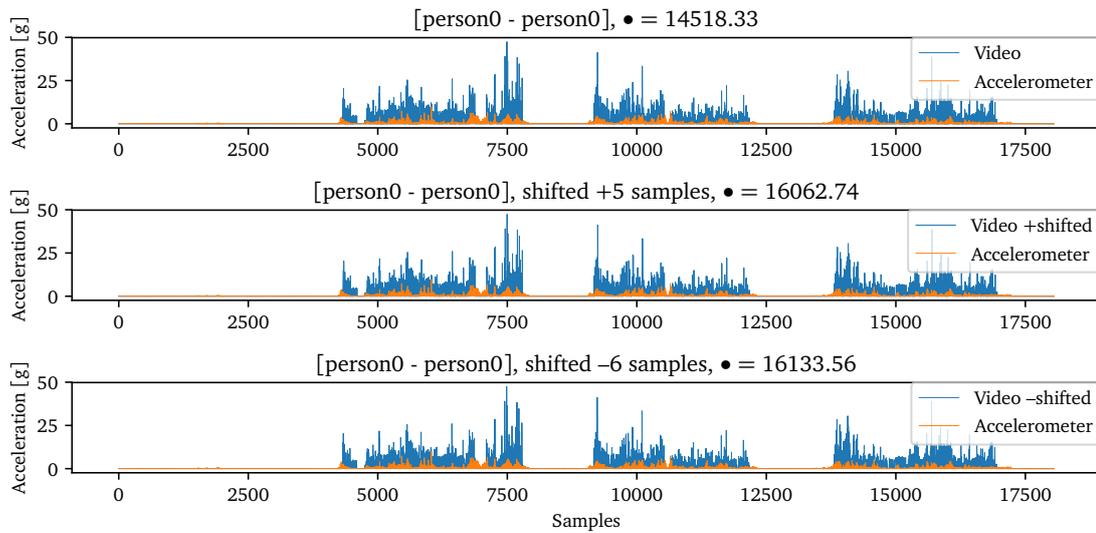


Figure 3.15: *person0* matched with *accelerometer0*, from top the bottom it shows; no shift, shift in positive direction and shift in negative direction. In the title for each graph the number e.g. middle graph tells  $+5$  which indicates how much the video is shifted in positive direction. As seen on the figure it is shifted with very small samples compared to Figure 3.16

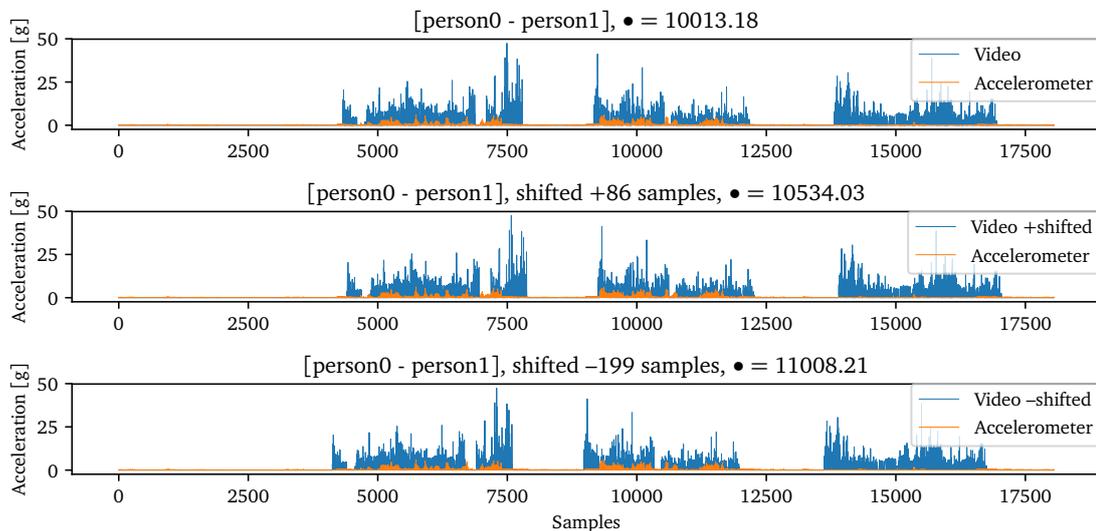


Figure 3.16: *person0* matched with *accelerometer1*, from top the bottom it shows; no shift, shift in positive direction and shift in negative direction. As seen here video is shifted considerably more in order to fit the accelerometer, with negative shift it is shifted with  $199$  samples to fit best.

during negative is significantly higher. When looking at *person2* and *person3*'s results it is seen that both are matched with *accelerometer3*, but the dot product between *person3* and *accelerometer3* is significantly higher and with less shifting required than *person2* matched with *accelerometer3*. Since a person can only be matched to one accelerometer, it makes sense to look at second best match in cases such as *person2*. As seen in Table A.3 the second best match for *person2* is *accelerometer6*, where no shift applied. However when both positive and negative shift is applied, *person2* matches with *accelerometer2*.

Table 3.2: Table showing the result of each video trajectory matched to all accelerometers, there are three columns, no shift, positive shift and negative shift representing if the video have been shifted or not. Then it shows columns with which accelerometer it have been matched and the resulting dot product. Lastly if its shifted there is a column showing how much for the best match.

Video	Results							
	No Shift		Positive Shift			Negative Shift		
	Acc.	Dot	Acc.	Shift+	Dot+	Acc.	Shift-	Dot-
<b>person0</b>	acc0	14518.33	acc0	5	16062.74	acc0	6	16133.56
<b>person1</b>	acc1	10193.76	acc1	2	10633.86	acc1	109	12642.14
<b>person2</b>	acc3	12976.63	acc3	10	13916.21	acc3	62	14393.84
<b>person3</b>	acc3	18088.68	acc3	2	18659.21	acc3	73	24559.12
<b>person4</b>	acc6	5380.56	acc6	11	5783.74	acc6	366	6422.89
<b>person5</b>	acc8	3413.56	acc8	26	3700.06	acc8	191	4038.75
<b>person6</b>	acc6	13642.12	acc6	5	14606.98	acc6	72	16654.60
<b>person7</b>	acc6	12530.91	acc6	407	13600.24	acc7	117	13766.39
<b>person8</b>	acc8	8758.56	acc8	25	9225.99	acc8	135	11654.01

The results shows that to some extent it is possible to match the estimated acceleration from video information with a players corresponding accelerometer. Some persons in the dataset spend little time on the field, which makes it harder to separate them from the others on the field at the same time. This is specifically seen for *person5*, whose estimated acceleration is matched better with other players accelerometer, in appendix Table A.6. This is caused due to *person5* only entering the field once during the recording as see in Figure 3.17. What is also seen from this accelerometer is that even when *person5* is on the field very little movement is measured by the accelerometer.

The resting acceleration for each accelerometer is variant, as seen in Figure 2.2, since the video and accelerometer is rarely at full rest in the test, it might be relevant to floor the acceleration if it is below a certain threshold. For the accelerometers the threshold is set to 0.2 g, based from the results in Figure 2.2. If the acceleration is less than the threshold it is set to 0, the thought behind this is to make the accelerometers rest at the same value and therefore the matches might become stronger. The results of this, see appendix A.1.2, shows no changes in the matches, but *person7*, see appendix Table A.8, with shift in negative direction, the necessary amount of samples to be shifted is 121 compared to 117 without threshold. As a natural cause of flooring under the threshold the length of the graph is reduced and the dot product is therefore likewise reduced.

As seen the data recording involve both time synchronisation issues and accelerometer measurements outside the field. Both of these can cause issues when matching

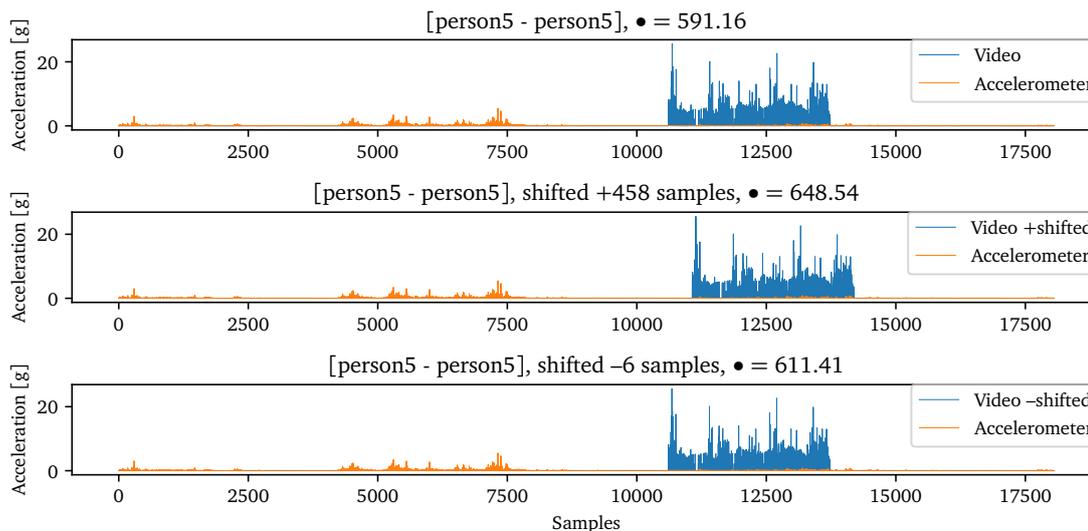


Figure 3.17: As it is seen in these graphs *person5* only enters the field once during recording (blue data). It is however also seen that *person5* is very stationary when it is on the field. Due to this low activity other accelerometers data might fit better. In positive direction it needs to be shifted 458 samples, where in negative direction it needs to be shifted 6 samples.

the trajectories with the accelerometers. It is attempted to improve the results by time synchronising the accelerometers with the video and remove the accelerometer measurements performed outside the camera view.

This is done by determine how much shift each accelerometer needs to be in sync with the video. The shift have been calculated by moving the video in both positive and negative direction over each accelerometer and the result with highest dot product (best match) have been saved. By looking at the dot results and shift in both positive and negative direction from Table 3.2, it can be concluded that the negative direction shift have most impact and overall gives a greater dot than positive shift and without shift.

The only person where negative shifting is not superior is *person5*, where the negative shift produces a dot value of 611.41 and the positive shift, 648.54. It is not significant greater and when comparing the amount of samples to be shifted, it needs to be shifted 6 samples in with negative shift and 458 during positive shift. As 458 samples corresponds to 18.32 seconds. Since the other persons will all be shifted in the negative direction it will not make sense to shift *person5*, so significantly in the opposite direction of the other accelerometers.

To shift the accelerometer so it is synchronised with the video it is necessary to move all accelerometers in positive direction by the amount of samples shown in negative shift at Table 3.2. The reasoning behind this is that the shift is calculated by shifting the video, while keeping the accelerometers steady, and an opposite movement is therefore required when video is kept static.

Last step is to remove the noise measured by the accelerometers when a person is not

present in the camera view. This is done checking for each time@£ step if the ground truth data contains information. If the the ground truth does not contain any data for the specific time step, the accelerometer data is set to zero for the corresponding person.

An example from the results after the sync can be seen at Figure 3.18, the important thing to notice is the upper figure shows spikes in the acceleration even though there are no video acceleration of the person, this can give some noise when comparing another person who have activity in this area since they match better with some small activity than with no activity.

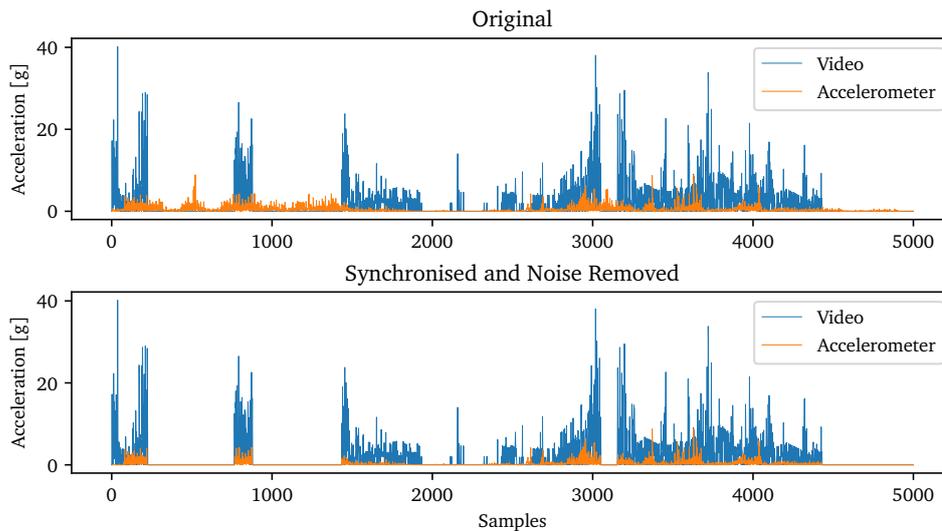


Figure 3.18: Figure showing the video track for *person3* and *accelerometer3* with no changes in the top figure and in the bottom figure it shows after they have been synchronised and their noise when not appearing in the video have been removed.

### 3.4.1 Matching 9 Trajectories With 9 Accelerometers

In section 3.4 the raw accelerometer data was held up against the video acceleration data created for each person. As seen the system was not capable matching all accelerometers with their corresponding players. The results from this section however helped synchronising the individual accelerometer data with the video. By having the accelerometers synchronised with the video it was further possible to remove noise and movement happening when the players was not occurring in the frames. By using the negative shift calculated in section 3.4, because of their general higher dot product.

Table 3.3: Here is the dot product results for each person’s trajectory matched to all accelerometers, the table shows the normalized dot product results in each row. The green boxes mark a correct match and the red boxes mark an incorrect match.

Traj. / Acc	acc0	acc1	acc2	acc3	acc4	acc5	acc6	acc7	acc8
person0	1.00	0.59	0.42	0.43	0.16	0.01	0.48	0.54	0.24
person1	0.82	1.00	0.40	0.23	0.26	0.03	0.26	0.62	0.35
person2	0.67	0.50	1.00	0.93	0.24	0.03	0.72	0.76	0.55
person3	0.41	0.14	0.42	1.00	0.01	0.02	0.69	0.48	0.24
person4	1.00	0.65	0.84	0.90	0.55	0.00	0.82	0.66	0.07
person5	0.35	0.45	0.47	0.81	0.10	0.18	0.00	0.66	1.00
person6	0.55	0.24	0.44	0.87	0.01	0.00	1.00	0.54	0.16
person7	0.78	0.64	0.68	0.84	0.15	0.05	0.60	1.00	0.53
person8	0.36	0.44	0.60	0.76	0.08	0.09	0.32	0.69	1.00

In Table 3.3 it can be seen that synchronisation and noise removal has improved the results found in section 3.4. In the negative shift results shown in Table 3.2 matched 6 out of 9 persons to their corresponding accelerometers. In Table 3.3 it is seen that 7 out of 9 persons are now matched to their corresponding accelerometers. Specifically *person4* is now correctly matched with its corresponding accelerometer. In order to improve the results further the optimisation algorithm known as Hungarian algorithm or Munkres-Kuhn algorithm is applied.

### Hungarian Algorithm

The Hungarian algorithm is an optimisation algorithm developed to solve assignment problems based on the cost of each task for each worker. Hungarian algorithm can assign task either based on minimising the cost or maximising the effectiveness. In our case the Hungarian algorithm will be used to maximise the effectiveness of each player with the accelerometers, by using the dot product.

The Hungarian algorithm for maximisation consist of the following steps[24, 25]:

- Step 1:** *Negate all elements in  $n \times n$  matrix and subtract the lowest value from all elements in matrix.*
- Step 2:** *Subtract the minimum row value from all elements in row.*
- Step 3:** *Subtract the minimum column value from all elements in column.*
- Step 4:** *Cover all zeroes in matrix with minimum number of lines. If number of lines  $< n$  proceed to step 5. If number of lines  $= n$  proceed to step 6.*
- Step 5:** *Subtract lowest uncovered value from all uncovered value and add lowest uncovered value to all values covered by twice. Return to step 4.*
- Step 6:** *For each row and column containing a single zero mark that zero as accepted. Exclude zeroes occurring in rows or columns with accepted zeroes as false and repeat step 6 if all workers are not assigned.*

For the purpose of showing how the algorithm works with all steps in affect an example is presented step-by-step in appendix A.1.4 to explain how the Hungarian algorithm works when maximising the effectiveness. In Table 3.4 shows the results of the Hungarian algorithm.

Table 3.4: Here is the results of Table 3.3, but with Hungarian algorithm applied. The green boxes mark a correct match. Each row shows the row normalized dot product.

Traj. / Acc	acc0	acc1	acc2	acc3	acc4	acc5	acc6	acc7	acc8
person0	<b>1.00</b>	0.59	0.42	0.43	0.16	0.01	0.48	0.54	0.24
person1	0.82	<b>1.00</b>	0.40	0.23	0.26	0.03	0.26	0.62	0.35
person2	0.67	0.50	<b>1.00</b>	0.93	0.24	0.03	0.72	0.76	0.55
person3	0.41	0.14	0.42	<b>1.00</b>	0.01	0.02	0.69	0.48	0.24
person4	1.00	0.65	0.84	0.90	<b>0.55</b>	0.00	0.82	0.66	0.07
person5	0.35	0.45	0.47	0.81	0.10	<b>0.18</b>	0.00	0.66	1.00
person6	0.55	0.24	0.44	0.87	0.01	0.00	<b>1.00</b>	0.54	0.16
person7	0.78	0.64	0.68	0.84	0.15	0.05	0.60	<b>1.00</b>	0.53
person8	0.36	0.44	0.60	0.76	0.08	0.09	0.32	0.69	<b>1.00</b>

As seen in Table 3.4 the Hungarian algorithm is matching the acceleration for each person, estimated from the video with its corresponding accelerometer data.

### 3.5 42 Trajectories Matched to 9 Accelerometers

With the synchronised accelerometers from section 3.4, it is possible to investigate how robust the dot product together with the Hungarian algorithm is, by matching the trajectories to the 9 accelerometers. The 42 trajectories is presented in Dataset 2b, section 2.3.2. As explained the trajectories starts when a player enters the field and ends when the player leaves the field.

There will be conducted seven different methods to investigate the impact they have on the results in this section:

**Method 1:** Will show how strong the dot product is to get correct matches.

**Method 2:** Will show how strong the dot product is, using Hungarian algorithm with no learn.

**Method 3:** Will show how strong row normalized dot product is, using Hungarian algorithm with no learn.

**Method 4:** Will show how strong dot product is, using Hungarian algorithm with learn.

**Method 5:** Will show how strong row normalized dot product is, using Hungarian algorithm with learn, to get correct matches.

**Method 6:** Is using a different similarity method proposed by Koutra et al.[19], as mentioned in section 2.2.4. This method is using binarized data to calculate the Russell-Rao dissimilarity using Hungarian algorithm with no learn.

**Method 7:** This method is using binarized data to calculate the Russell-Rao dissimilarity using Hungarian algorithm with learn.

The results from all the methods can be found in appendix A.2, where the dot product, row normalized dot product and binarized data for each trajectory is matched with all accelerometers. In Table 3.6 the correct and false predictions is shown. a minimised version of this table can be seen below in Table 3.5. A more thorough explanation of each method and the results is presented in the remaining part of this section.

Table 3.5: Shows a minimised table of the results in Table 3.6

	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7
<b>Total</b>	<b>18</b> (42.86%)	<b>22</b> (52.38%)	<b>20</b> (47.62%)	<b>31</b> (73.81%)	<b>35</b> (83.33%)	<b>39</b> (92.86%)	<b>42</b> (100.00%)

#### Method 1 – Dot Product

In this method the 42 trajectories are matched with the 9 accelerometers. The highest dot product calculated for a single trajectory to the 9 accelerometers are taken as the match.

As seen in Table 3.6 this results 35.71% of the trajectories being correctly matched with its corresponding accelerometer.

## Method 2 – Dot Product, Hungarian No Learn

The input to the Hungarian algorithm, presented in section 3.4.1, is found using two different methods. In this section the no learn method is explained where the learn method is going to be explained in *Method 4*. The trajectories presented in Figure 3.19 represent the time samples in which each trajectory is present in the video. The input matrix is found as following: Seen in Figure 3.19 *trajectory 2* is the first trajectory that ends, which is affected by other trajectories. As seen *trajectory 1* is present with *trajectory 2*. The first input matrix to the Hungarian algorithm is therefore *trajectory 1* and *trajectory 2*. Since *trajectory 1* have an impact on future trajectories only the results for *trajectory 2* can be accepted from the first calculation. As seen that the next trajectory that ends is *trajectory 3*. Since *trajectory 3* is affected by *trajectory 4-6* and *trajectory 1*. The input to the next Hungarian calculation is therefore the above mentioned trajectories. Since *trajectory 3* ends first, only the result of this trajectory is accepted. This process continues until all trajectories have been matched with an accelerometer. The process can be explained by the following steps:

**Step 1:** Find the next trajectory which stops, call it  $traj_{stop}$ .

**Step 2:** Take all the trajectories in the same timespan as  $traj_{stop}$ .

**Step 3:** Run Hungarian on trajectories.

**Step 4:** Save match result for  $traj_{stop}$ .

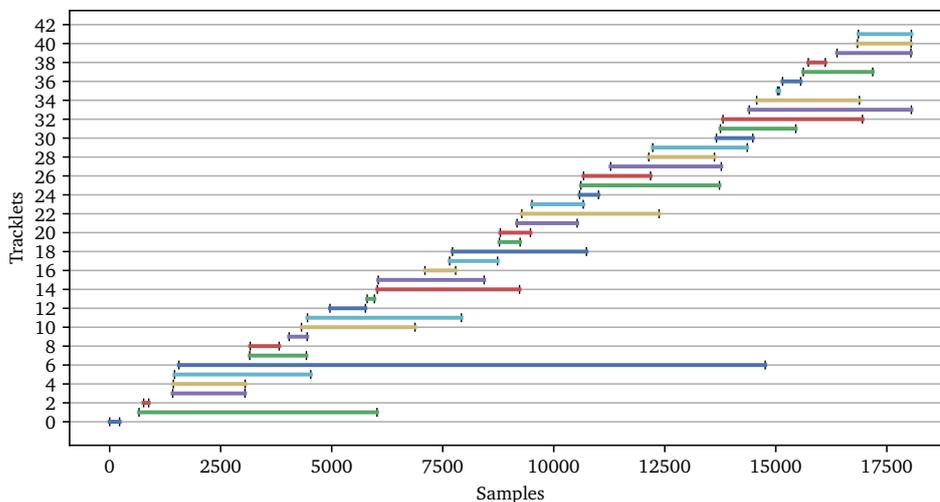


Figure 3.19: Showing each trajectory and how many samples they appear in the video, the only two trajectories which stops at the same time is *trajectory 33* and *41*.

As seen in table 3.6 this results in 52.38% trajectories being matched to the correct accelerometers.

### Method 3 – Row Normalized Dot Product, Hungarian No Learn

This method use similar method as explained in *Method 2*, where Hungarian no learn is applied. The dot product results is however row normalized, meaning that for each trajectory the dot products for each accelerometer is divided by the highest dot product for this trajectory.

As seen in Table 3.6 the method resulted in 42.38% trajectories being matched to the correct accelerometer, which compared to the results of *Method 2* is severely worse results.

### Method 4 – Dot Product, Hungarian Learn

In this method a Hungarian learn is introduced. In *Method 2* the method for no learn method was explained and the learn is an addition to this method. Hungarian learn is based on the fact that two trajectories present at same time can not belong to the same accelerometer. In first Hungarian calculation seen in *Method 2* states that *trajectory 2* is classified. As an example lets state that *trajectory 2* is classified to *accelerometer1*. Since *trajectory 1* is present at the same time as *trajectory 2*, it is given that *trajectory 1* cannot be matched with *accelerometer1*. By setting the dot product for *accelerometer1* to the value 0 in *trajectory 1* further Hungarian calculations using *trajectory 1*, will remove the possibility of *trajectory 1* being matched to *accelerometer1*. The process seen in *Method 2* gets a fifth step added:

**Step 1:** Find the next trajectory which stops, call it  $traj_{stop}$ .

**Step 2:** Take all the trajectories in the same timespan as  $traj_{stop}$ .

**Step 3:** Run Hungarian on trajectories.

**Step 4:** Save match result for  $traj_{stop}$ .

**Step 5:** Remove the result from the trajectories in the same timespan as  $traj_{stop}$ .

As seen in Table 3.6 the method resulted in 73.81% of the trajectories being matched to the correct accelerometer. The results of the learning method exceeds the results presented for *Method 2*.

### Method 5 – Row Normalized Dot Product, Hungarian Learn

In this method the row normalized dot product is used as input to Hungarian algorithm with learn. As seen in Table 3.6 the method resulted in 83.33% trajectories being matched to the correct accelerometer. As seen the results exceeds *Method 4*'s results.

### Method 6 – Russell-Rao, Binarized Data, Hungarian No Learn

As mentioned in section 2.2.4, other graph matching methods such as Russell-Rao can be used. In order to use this method the acceleration data for the 42 trajectories and accelerometer data needs to be translated into binary data[26]. This is done as seen in Code snippet 3.2.

```

1   for i in range(len(data)):
2       if videoData[i] > 0:
3           videoData[i] = 1
4       if accData[i] > 0:
5           accData[i] = 1

```

Code snippet 3.2: Pseudo code showing how Dataset 2b have been binarized

On the binarized data the Russell-Rao dissimilarity algorithm will be applied. Russell-Rao method computes the dissimilarity between two boolean vectors ( $u$  and  $v$ ). The Russell-Rao is then calculated as seen in equation (3.25).

$$\frac{n - c_{ij}}{n} \quad (3.25)$$

where  $c_{ij}$  is the number of occurrences where  $u[k] = i$  and  $v[k] = j$  for  $k < n$ , which mean that  $u[k] = v[k] = 1$ .  $n$  is the total number of samples. The reason for investigating Russell-Rao is that this algorithm does not take the magnitude of the vectors into account, as dot product does [19, 26].

Once the Russell-Rao is calculated between the trajectories and accelerometers the Hungarian algorithm with no learn is applied. As seen in Table 3.6 the method results in 92.86% of the trajectories being matched to the correct accelerometer.

### Method 7 – Russell-Rao, Hungarian Learn

In this method the same binarized data as seen in *Method 6* is used to calculate the Russell-Rao dissimilarity using Hungarian algorithm with learn is applied. As seen in Table 3.6 this method resulted in 100.00% of the trajectories being matched to the correct accelerometer.

Table 3.6: Table showing each tracklet in dataset 2b and their ground truth person match. **Method 1** is the best match with dot product. **Method 2** is after an Hungarian with no learn have been applied. **Method 3** is Hungarian with no learn on normalized results. **Method 4** is where Hungarian with learn have been used. **Method 5** is Hungarian with learn on row normalized dot product, **Method 6** is Russell-Rao using Hungarian with no learn. **Method 7** is Russell-Rao using Hungarian with learn

Trac.	GT.	Method 1	Method 2	Method 3	Method 4	Method 5	Method 6	Method 7
0	3	3 ✓	3 ✓	3 ✓	3 ✓	3 ✓	3 ✓	3 ✓
1	2	8 ✗	8 ✗	8 ✗	8 ✗	2 ✓	2 ✓	2 ✓
2	3	3 ✓	3 ✓	3 ✓	3 ✓	3 ✓	3 ✓	3 ✓
3	6	3 ✗	6 ✓	6 ✓	6 ✓	6 ✓	3 ✗	6 ✓
4	3	3 ✓	6 ✗	3 ✓	2 ✗	3 ✓	3 ✓	3 ✓
5	8	3 ✗	3 ✗	3 ✗	3 ✗	8 ✓	8 ✓	8 ✓
6	7	7 ✓	7 ✓	7 ✓	7 ✓	7 ✓	7 ✓	7 ✓
7	3	3 ✓	6 ✗	6 ✗	2 ✗	3 ✓	3 ✓	3 ✓
8	6	6 ✓	6 ✓	6 ✓	6 ✓	6 ✓	6 ✓	6 ✓
9	6	8 ✗	6 ✓	8 ✗	6 ✓	6 ✓	6 ✓	6 ✓
10	0	0 ✓	0 ✓	0 ✓	0 ✓	1 ✗	0 ✓	0 ✓
11	1	0 ✗	0 ✗	0 ✗	1 ✓	8 ✗	1 ✓	1 ✓
12	8	1 ✗	8 ✓	1 ✗	2 ✗	8 ✓	8 ✓	8 ✓
13	8	0 ✗	2 ✗	0 ✗	2 ✗	0 ✗	8 ✓	8 ✓
14	3	6 ✗	6 ✗	6 ✗	3 ✓	3 ✓	3 ✓	3 ✓
15	6	6 ✓	6 ✓	6 ✓	6 ✓	6 ✓	6 ✓	6 ✓
16	0	0 ✓	0 ✓	0 ✓	0 ✓	0 ✓	0 ✓	0 ✓
17	2	3 ✗	3 ✗	3 ✗	2 ✓	2 ✓	2 ✓	2 ✓
18	4	2 ✗	2 ✗	2 ✗	4 ✓	4 ✓	4 ✓	4 ✓
19	6	7 ✗	6 ✓	7 ✗	6 ✓	6 ✓	6 ✓	6 ✓
20	2	2 ✓	4 ✗	2 ✓	2 ✓	2 ✓	2 ✓	2 ✓
21	0	1 ✗	0 ✓	0 ✓	0 ✓	0 ✓	0 ✓	0 ✓
22	1	1 ✓	1 ✓	1 ✓	1 ✓	1 ✓	1 ✓	1 ✓
23	2	1 ✗	0 ✗	1 ✗	2 ✓	2 ✓	2 ✓	2 ✓
24	8	8 ✓	8 ✓	8 ✓	8 ✓	8 ✓	8 ✓	8 ✓
25	5	8 ✗	8 ✗	8 ✗	5 ✓	5 ✓	5 ✓	5 ✓
26	0	1 ✗	0 ✓	0 ✓	0 ✓	0 ✓	0 ✓	0 ✓
27	8	3 ✗	8 ✓	8 ✓	8 ✓	8 ✓	5 ✗	8 ✓
28	3	3 ✓	3 ✓	3 ✓	3 ✓	3 ✓	3 ✓	3 ✓
29	2	3 ✗	3 ✗	3 ✗	2 ✓	2 ✓	2 ✓	2 ✓
30	3	0 ✗	2 ✗	2 ✗	8 ✗	0 ✗	3 ✓	3 ✓
31	6	6 ✓	2 ✗	2 ✗	6 ✓	6 ✓	6 ✓	6 ✓
32	0	0 ✓	0 ✓	0 ✓	0 ✓	5 ✗	0 ✓	0 ✓
33	2	6 ✗	6 ✗	3 ✗	2 ✓	2 ✓	2 ✓	2 ✓
34	3	3 ✓	3 ✓	3 ✓	3 ✓	3 ✓	3 ✓	3 ✓
35	7	0 ✗	7 ✓	7 ✓	7 ✓	7 ✓	7 ✓	7 ✓
36	4	6 ✗	2 ✗	6 ✗	4 ✓	0 ✗	4 ✓	4 ✓
37	4	3 ✗	3 ✗	3 ✗	6 ✗	0 ✗	4 ✓	4 ✓
38	6	6 ✓	4 ✗	6 ✓	4 ✗	6 ✓	6 ✓	6 ✓
39	6	6 ✓	6 ✓	3 ✗	8 ✗	6 ✓	6 ✓	6 ✓
40	8	6 ✗	6 ✗	6 ✗	4 ✗	8 ✓	1 ✗	8 ✓
41	1	6 ✗	1 ✓	6 ✗	1 ✓	1 ✓	1 ✓	1 ✓
<b>Total</b>		<b>18</b> 42.86%	<b>22</b> 52.38%	<b>20</b> 47.62%	<b>31</b> 73.81%	<b>35</b> 83.33%	<b>39</b> 92.86%	<b>42</b> 100.00%

### 3.5.1 Conclusion

The results shows that a Hungarian with learn (*Method 4-5*) is significantly better than dot product only and Hungarian with no learn (*Method 2-3*). An issue which can occur with Hungarian learn method, is that if misclassification happens, see at *trajectory4*, *Method 4*, the error results in other trajectories being classified incorrectly. In *Method 6* and *7* it is

seen that Russell-Rao out performs the dot product and row normalized dot product. In *Method 7* the results ended in 100.00 %, which makes this method the best for this specific dataset. What is important to note is that the trajectories used in this dataset are the full trajectories when the players are present in the scene. This means that Russell-Rao might have issues when these trajectories are broken due to occlusions resulting in even smaller trajectories, where multiple trajectory might fit equally well to the same accelerometer. In this case the magnitude which is present in the dot product might end up being better to match smaller trajectories.

### 3.6 Certainty Threshold

In this section it will be attempted to use a threshold in order to verify the certainty of tracklet choice. The row normalized dot product in combination with Dataset 2b is used in order to visualise the effect of the certainty threshold. The True Positive, False Positive, True Negative and False Negative is defined.

**True Positive (TP):** Maximum value corresponds to the ground truth accelerometer and remaining results are below the threshold.

**False Positive (FP):** Maximum value does not correspond to ground truth accelerometer and remaining results are below threshold.

**False Negative (FN):** Maximum value corresponds to ground truth accelerometer and remaining results are not below threshold.

**True Negative (TN):** Maximum value does not correspond to ground truth accelerometer and remaining results are not below threshold.

To find the best certainty threshold a FN vs. FP graph has been created, which visualise how the threshold effect the results. Be cause the row normalized dot product is used, the maximum value for each tracklet is 1. Since all other values for each tracklet is below the maximum, the threshold will vary from 0 to 1. In Figure 3.20 the graph showing the FP against FN.

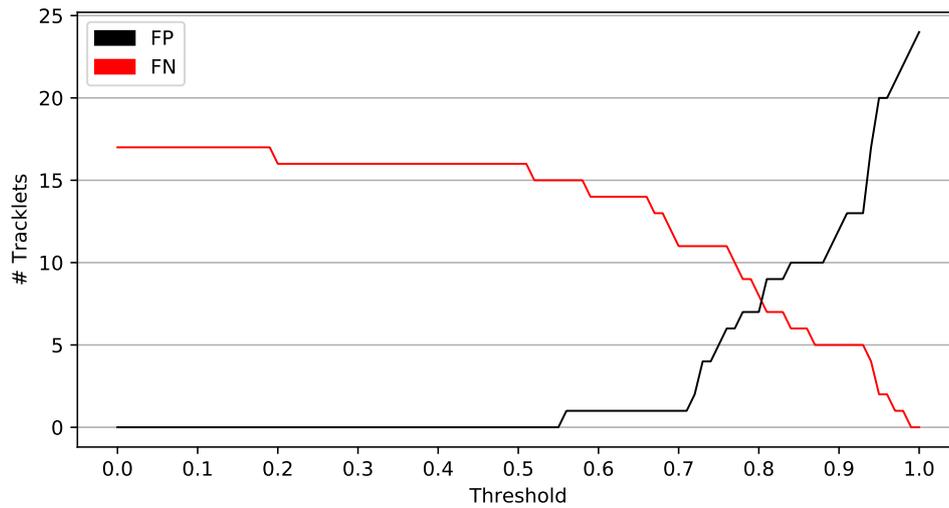


Figure 3.20: In this graph the FP and FN rates for a given threshold is displayed.

In this system it is preferred to not have any FP as these are not matching the ground truth, but still below the threshold. By looking at Figure 3.20 it is seen that if the FP is to be kept at a minimum the threshold needs to be 0.57. If the TP and TN is included in the graph as seen on Figure 3.21.

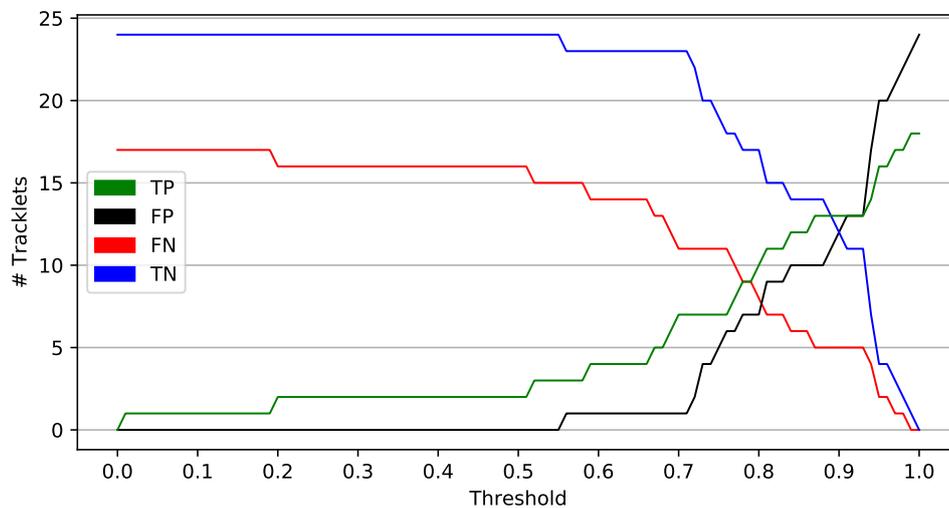


Figure 3.21: In this graph the previous FN and FP is displayed along with the TP and TN rates for a given threshold.

As seen here if the threshold is set to 0.57 the number of TP will only be 4. However if the threshold is 0.72 the number of TP will be 7 at the cost of FP being 1. If the threshold is increased further it is seen that the FP increase very quickly from this point.

When applying the certainty threshold it is seen that 8 tracklets of 42 are matched with an accelerometer. This leads to 34 tracklets unmatched with an accelerometer.

Another approach which has been attempted using the certainty threshold is to remove all values below the set threshold for each tracklet. After these have been removed it is then attempted to match the accelerometers using the Hungarian learn and no learn. The test results is seen in Table 3.7, where it is also seen that the thresholds 0.57 and 0.72 as these are discussed above.

Table 3.7: This table presents the trajectories matched to the correct accelerometer. The top bar corresponds to which Hungarian Method.

Hungarian	No learn	Learn	No learn	Learn
Threshold	0.57		0.72	
True (✓)	17	14	16	17
False (✗)	25	28	26	25
Correct	40.48%	33.33%	38.10%	40.48%

When comparing the results from Table 3.7 with the results from Table 3.6 it is seen that by applying a certainty threshold the results of the Hungarian becomes significantly worse.

As a conclusion to this investigation it can be seen that by applying a threshold to the results, the system is only able to predict 8 out of 42 tracklets to a accelerometers, when Hungarian is not used. If a threshold larger than 0.72 is chosen the number of FP increase substantially. It is also seen that when the row normalized dot product values below the certainty threshold is set to 0 in the input matrix. The Hungarian algorithm is performing significantly worse.

# Acceptance Test

---

In this chapter the system designed will be tested. A set of tests have been conducted on the system in order to evaluate the sub-modules. First the tracker will be tested on a snippet of the video being the first 3.5 minutes of the video recorded in Gigantium. A CLEAR MOT evaluation will be performed in order to see how the tracker is performing. The tracklets produced from the tracker is then used in order to test the ability of matching the tracklets with the accelerometer data.

In order to test how long tracklets are required in order for the matching method to perform as intended and when misclassifications might occur.

Lastly the systems ability to reconnect tracklets after an occlusion has separated is tested by using a snippet where an occlusion occurs and separates.

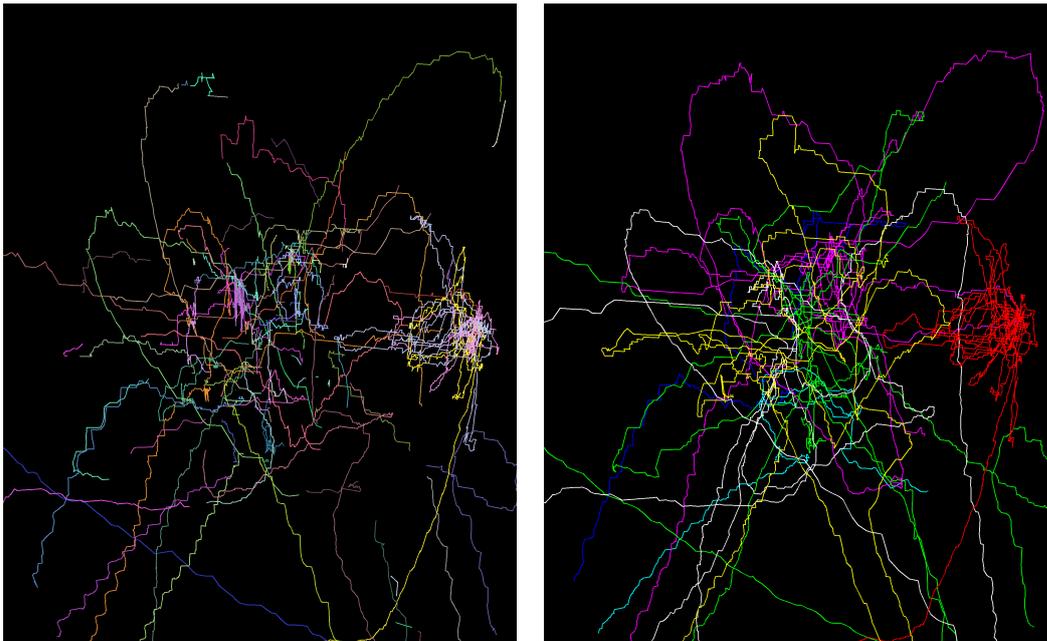
## 4.1 Test of the Tracking System

In this section the implemented tracker will be tested in two ways; One on how well it tracks persons. Second is a full-system test.

The test have been conducted on the Gigantium sports arena dataset, mentioned in Dataset 2. The first 5250 frames (3 min and 30 sec) of the 18075 frames big dataset have been used for the test. The tracker produced a total of 200 tracklets. Out of the 200 tracklets 123 of them contain a single person which means the remaining 77 tracklets consist of occluded persons.

As explained in Table 3.1, each tracklet is indicated by an ID which contains the number of people within this ID. Which is how the data have been filtered to only save the tracklets containing a single person. Furthermore out of the 123 tracklets with one person, 18 of these contain less than 3 data samples. Since an acceleration cannot be calculated from 1 or 2 samples, these have likewise been excluded. Therefore a total of 105 tracklets containing a single person and have more span over more than 2 samples, have been found, which is used to conduct the two tests for the tracking system.

Since the purpose of the project is to reconnect tracklets from individual players after an occlusion, the single player tracklets are the focus of this project. In Figure 4.1a the single player tracklets found by the tracker for the first 5250 frames are shown. In order to visually compare the tracker and ground truth, the ground truth trajectories for the first 5250 frames is shown in Figure 4.1b.



(a) Tracklets of single persons found by the tracking algorithm.

(b) Ground-truth trajectories from first 5250 frames

Figure 4.1: Both figures shows data for the first 5250 frames from the Gigantium, Dataset 2. **(a)** Shows the tracklets found by the automatic tracking system. **(b)** shows the ground truth trajectories. The colours corresponds to the individual players, where in Figure 4.1a the colours corresponds to a tracklet, but does not indict which player the tracklet corresponds to.

What is important to notice on Figure 4.1, is that the tracklets of all the single persons match the ground-truth tracklets visually quite well.

To test how strong the tracker is compared to ground truth the CLEAR MOT Metrics will be calculated.

#### 4.1.1 CLEAR MOT Metrics

CLEAR MOT is a "Multiple Object Tracking Performance"-method for testing tracking techniques, used to compare results between tracking techniques.

The procedure which is proposed by Bernardin et al.[27] will now be explained: Assuming that for every frame,  $t$ , a multi object tracker outputs a set of hypotheses  $h_1, \dots, h_m$  for a set of visible objects  $o_1, \dots, o_n$ . Then the evaluation for the procedure can be explained by the following steps:

For each time frame,  $t$ :

**Step 1:** Create the best possible matches between hypotheses  $h_j$  and objects  $o_i$ .

**Step 2:** For every found match compute the error in the object's position estimation.

**Step 3:** Accumulate all errors:

- a) Count all objects which does not have a match to a hypotheses, call these: misses.
- b) Count all hypotheses which have no match to any real objects, call these: false positives.
- c) Count all occurrences where the hypothesis for an object changed compared to the previous frames, call these: mismatch errors.

When following these steps it is now possible to calculate the performance metrics, which is defined by Bernardin et al.[27]:

Multiple object tracking precision (MOPT), Equation (4.1).

$$MOTP = \frac{\sum_{i,t} d_t^i}{\sum_t c_t} \quad (4.1)$$

$d_t^i$  is the distance between an object  $o_i$  and its corresponding hypothesis  $h_j$ , which is the total error in the estimated position for the matched pairs over all frames. It is then averaged over the total number of matched made,  $c_t$ . The MOTP evaluate the trackers ability to precisely estimate an object positions.

$$MOTA = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t} \quad (4.2)$$

$m_t$  is the number of misses,  $fp_t$  is the number of false positives and  $mme_t$  is the number of mismatches,  $g_t$  is the total number of objects present in all frames. The MOTA accounts for all errors made by the tracker over all frames, the result will tell the tracker's performance at detecting objects and keeping their trajectories, independent of the precision for the object locations. MOTA is derived from three error ratios shown in Equations (4.3) to (4.5).

$$\bar{m} = \frac{\sum_t m_t}{\sum_t g_t} \quad (4.3) \quad \bar{fp} = \frac{\sum_t fp_t}{\sum_t g_t} \quad (4.4) \quad \bar{mme} = \frac{\sum_t mme_t}{\sum_t g_t} \quad (4.5)$$

Lastly Bernardin et al. propose a threshold  $T$ , which means that all the match between an object  $o_i$  and hypothesis  $h_j$  should not be made if their distance exceed a certain threshold  $T$ . The threshold exist since it is not valid to speak about an error in position estimation but instead should argue that the tracker has missed the object and is tracking something else, when exceeding  $T$ .

When using these definitions and equations it is possible to calculate the CLEAR MOT metrics, the threshold have been set to  $T = 50$  cm which is the same threshold used from Bernardin et al.[27][28]. The results can be in Table 4.1.

Table 4.1: Results for the CLEAR MOT metrics on the first 5250 frames from Gigantium Dataset 2.

MOTP (cm)	$\bar{m}$	$\bar{fp}$	$\bar{mme}$	MOTA
20.29	32.06%	6.69%	0.15%	61.10%

The MOTP shows a result of 20.29 cm is the average position error, which is not bad, considered the position is based on the bottom centre of the bounding box, so a variance to the ground truth is expected.

The results shows a high ratio of misses  $\bar{m}$  on approximate 32 %, this is mainly due to the ground truth data have the data from occlusions, but in the tested data for the tracker, all tracklets with an occlusion have been removed, because the system does not know, which object  $o_i$  an occlusion hypothesis  $h_j$  belongs to, since both persons have the same position when they are occluded, as explained section 3.2.2.

There is a false positive rate,  $\bar{fp}$  on nearly 7 % which is primarily matches, who are above the set threshold.

Also it is expected to have a low ratio of mismatched,  $\bar{mme}$  since the occlusions have been removed from the tracking data and that is mainly where mismatches can occur.

Lastly the MOTA gives a score of 61 % which tells how many matches are correct.

The threshold can be adjusted and the following Figure 4.2 shows what effect the threshold have on the MOTA score and MOTP value.

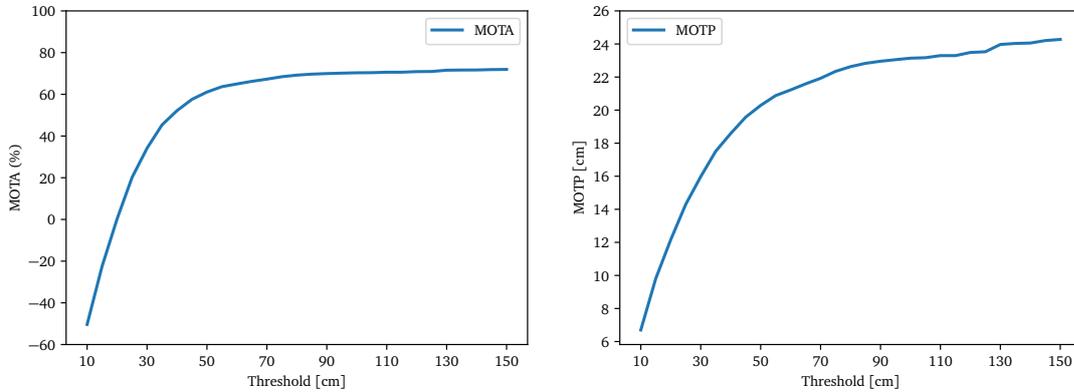


Figure 4.2: Shows what happens to the MOTA score and MOTP value when adjusting the threshold over a range from 10 to 150 cm.

As it can be seen in Figure 4.2 when the threshold hits 50 cm, it gains nearly nothing in MOTA score, but the MOTP value still increase a bit afterwards.

Since there is a lack of occlusion information in data from the tracking system this will increase the misses in the system, it can then be relevant to see how well the tracker performs if the segments where the occlusions are present is removed from the MOT results.

### CLEAR MOT Metrics – Without Any Occlusions

It is only possible to get the results without occlusion by using the ground truth data where the tracker have not detected any occlusions on the field. So by using the information which the tracker developed have, when an occlusion occur on the field. It is then possible to skip all samples where an occlusion occur. The results for this can be seen in Table 4.2.

Table 4.2: Results for the CLEAR MOT metrics on the first 5250 frames from Gigantium Dataset 2., where the MOT metrics have only been calculated if no occlusion appeared on the field, at the frame  $t$ .

MOTP (cm)	$\bar{m}$	$\bar{fp}$	$\bar{mme}$	MOTA
19.51	6.77%	9.42%	0.01%	83.80%

The MOTP value is a bit better than previous Table 4.1, furthermore the miss ratio  $\bar{m}$  have been reduced significantly, because segments where occlusions occur have been removed. The  $\bar{fp}$  have increased a bit.

Mismatches,  $\bar{mme}$  is nearly 0, which makes sense since the occlusions have been removed and therefore it is near impossible for the system to make mismatches since it is mostly due to the occlusion that a mismatch can occur.

Lastly the MOTA score have gone up by approximate 22%, showing that the system makes less error, when no occlusion occurs.

Lastly how the threshold impact MOTA score and MOTP value can be seen in Figure 4.3.

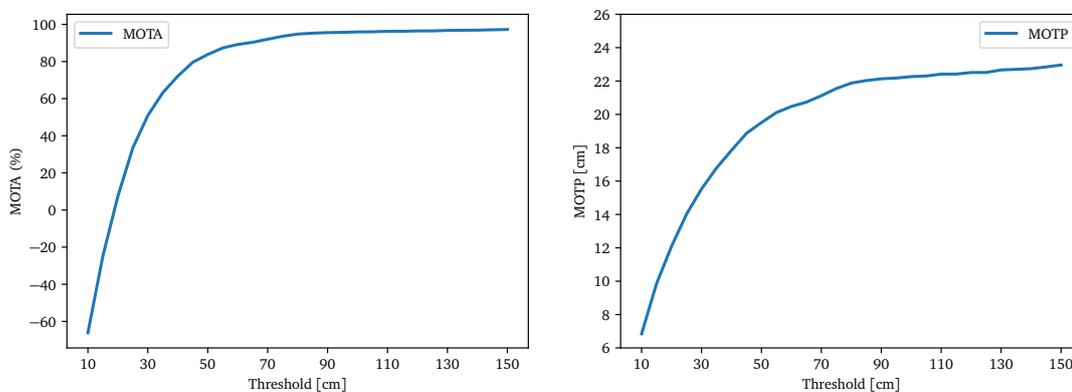


Figure 4.3: Shows what happens to the MOTA score and MOTP value when adjusting the threshold over a range from 10 to 150 cm.

The figure is shaped in the same way as the previous, Figure 4.2, where it is mainly around threshold  $T = 50$  where the MOTA score stops increasing.

The next section will be the full system test, which will show how robust the tracker is to provide information which can be estimated to an acceleration and then matched to the accelerometers.

## 4.2 Matching Tracking Results to Accelerometers

This test is using the same methods from section 3.5, but instead of using the labelled positions of each person, the automatic tracking data have been used, presented in section 4.1.1. The test is done to test how well the system performs in a "chaos"-scenario. Each tracklet and their length is shown in Figure 4.4. The test will be conducted in a similar way as in section 3.5 and the results can be seen in Table 4.3.

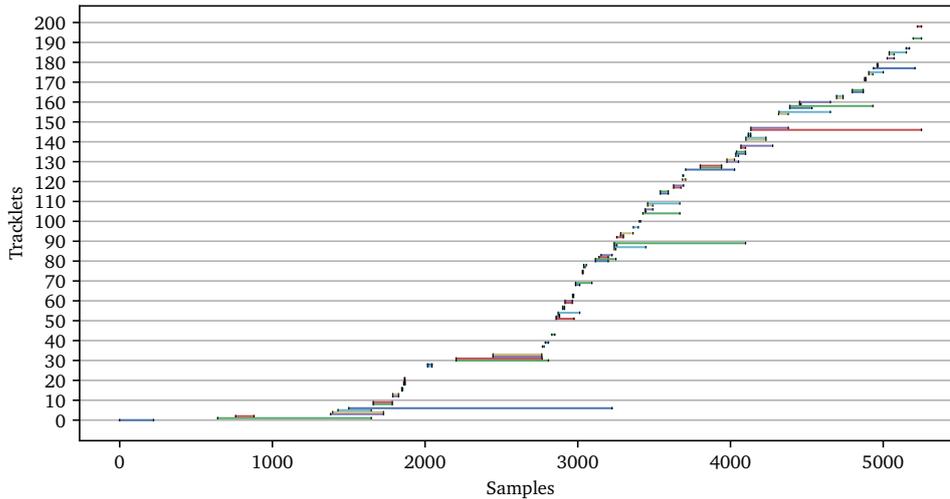


Figure 4.4: Figure shows each tracklet start and end frame of the data generated from section 3.2, tracklets smaller than 2 samples is removed as well as tracklets with occlusions.

Table 4.3: The results of 105 tracklets are listed as true match or false match, over the seven different methods done in section 3.5. To summarise the methods; Method 1: Dot product only. Method 2: Hungarian no learn. Method 3: Hungarian no learn with row normalized data. Method 4: Hungarian learn. Method 5: Hungarian learn with row normalized data. Method 6: Hungarian no learn, Russell-Rao. Method 7: Hungarian learn, Russell-Rao

Match	Method 1	Method 2	Method 3	Method 4	Method 5	Method 6	Method 7
True (✓)	24	24	18	25	21	28	33
False (✗)	81	81	87	80	84	77	72
Correct	22.86%	22.86%	17.14%	23.81%	20.00%	26.67%	31.43%

The results shows that the data is challenging considered that the system have to match tracklets with as few samples as 3. These are extremely difficult to match especially when the system relies on different activity and with 3 samples it is hard to judge which person the trajectory belongs to. Out of the 105 trajectories there are a total of 87 tracklets which have less than or equal to 100 samples, equivalent of 4 seconds.

In Table 4.3 it is interesting to note that the row normalized data does not improve any results compared to using dot product, as the opposite were experienced in section 3.5, where the *Method 4* was improved with 9.52% but in the "Chaos"-scenario it gets worse than using Hungarian on the dot products. Furthermore the Russell-Rao is in general better than the other methods, which is also shown in section 3.5. It is seen that with Hungarian no learn, *Method 6* scores nearly 3 percentage point more than *Method 4*,

which is dot product with Hungarian learn. The best results is achieved is *Method 7*, where Russell-Rao with Hungarian learn scores approximate 5 percentage point more than the second best.

#### 4.2.1 Conclusion of the Test

The results show that Russell-Rao is the best approach when handling this type of data where there are clear areas with no activity. Method 7 scores approximate 31 % correct matches, which is not enough to say the approach can be used to successfully track a sport during practice, but it is also important to note that these results are on a "Chaotic"-scenario, with a lot of broken trajectories due to the tracker. This also implies that it is worth to improve the method used for tracking.

### 4.3 Test of Minimum Samples Required for Matching

In this section, the robustness in matching between accelerometer and tracklet is being tested.

The idea behind this test is to take two persons on the field, one person moving a bit, call it *ID0*, and another person moving fast, call it *ID1*. Then the *ID0*'s trajectory will be kept as it is, but *ID1* trajectory will be split in the middle, so it creates two tracklets, lets call the tracklets for *ID1*, *ID1(a)* and *ID1(b)*.

Then it will be matched to the two accelerometer representing the two persons on the field. It is expected that trajectory *ID0* will be matched to its own accelerometer, *acc0* and both tracklets *ID1(a)* and *ID1(b)* will both be matched to its own accelerometer, *acc1*.

Then one sample will be removed in the end of *ID1(a)* and the beginning of *ID1(b)* and it will run again.

Then an extra sample will be removed and the step will be repeated until the two tracklets from *ID1* will be matched to the wrong accelerometer.

The test will then show how many samples is necessary in a tracklet for the system to match it correct. The necessary steps for the test can be summed up:

- Step 1:** Find two trajectories  $traj_0, traj_1$ , in the same timespan  $(t_0, \dots, t_n)$
- Step 2:** Split  $traj_1$  into two tracklets,  $tracklet_0, tracklet_1$
- Step 3:** Remove  $q$  sample from end of  $tracklet_0$  and beginning of  $tracklet_1$
- Step 4:** Match  $traj_0, tracklet_0$  and  $tracklet_1$  to the two accelerometers  $acc_0, acc_1$
- Step 5:** Repeat step 3-4 with a new,  $q$ -value.

To be able to investigate how robust it is, a simple scenario will be used: A snippet of the Gigantium test, Dataset 2, have been extracted, which consist of one person mainly standing still and then a person running across the field. The total tracklets length is 100 frames (4.04 seconds) and goes from frame 760 to 860, the snippet start and end can be seen at Figure 4.5, and each ID's trajectory.

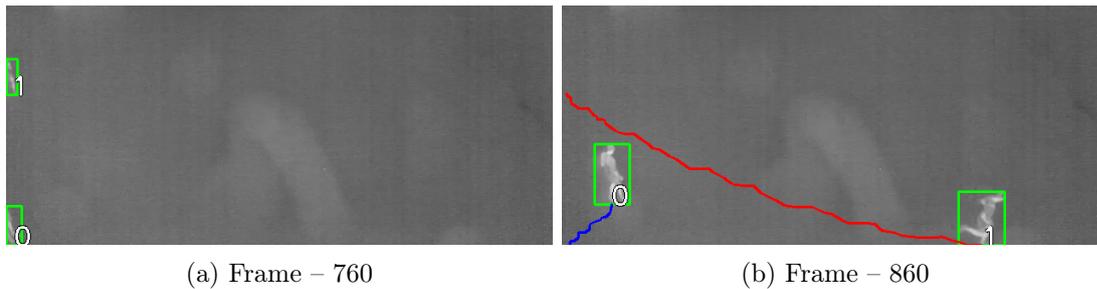


Figure 4.5: Each image is cropped and shows a bounding box around each contour and the text indicates the contours corresponding ID. (a) is the start of the snippet (b) is the end of the snippet, each ID and their corresponding path through the snippet is indicated by the line,  $ID0$  trajectory is the blue line and  $ID1$  is the red line.

It is important to note that the trajectory in Figure 4.5b is only representing their position through the snippet. The position is calculated to an acceleration and is then matched to the two persons accelerometers.

The results for the test gives that the first tracklet  $ID0(a)$ , which lose samples in the end, get matched wrong after it lost 16 samples which gives a remaining 34 samples in the tracklet. Furthermore  $ID0(b)$  which lose samples from the beginning, get matched wrong after it lost 25 samples from the tracklet, resulting in 25 samples remaining.

The test have only been conducted on one snippet, which gives limited results to conclude on. Though the test shows that it is desirable to have a tracklet of more than 25 samples and even better if it is above 34 samples. The data, the test have been conducted on is consisting of a running person  $ID1$  and a slow walking person  $ID0$ , which is the two states a person is usually in. This puts even more weight on that it is desirable to have more than 34 samples of a tracklet, to be able to match correct.

#### 4.4 Connecting Tracklets After an Occlusion

In this section, the systems ability to reconnect tracks after an occlusion has occurred is tested, which is related to the question formulated in the Problem Statement, section 2.5.

In order to test whether its even possible a snippet of the video have been selected, where an occlusion happens. The positional data is extracted from the automatic tracking system made in section 3.2. The snippet is chosen such that the players are tracked before and after the occlusion. The snippet contains 3 players on the field. One of the players is the goalkeeper and the two other players are occluding. In Figure 4.6a the first frame of the snippet is seen. In Figure 4.6b the first frame where the occlusion is happening is seen. In Figure 4.6c the first frame after the occlusion is splitting is seen. In Figure 4.6d the last frame of this snippet is seen. The snippet spans over total of 81 frames.

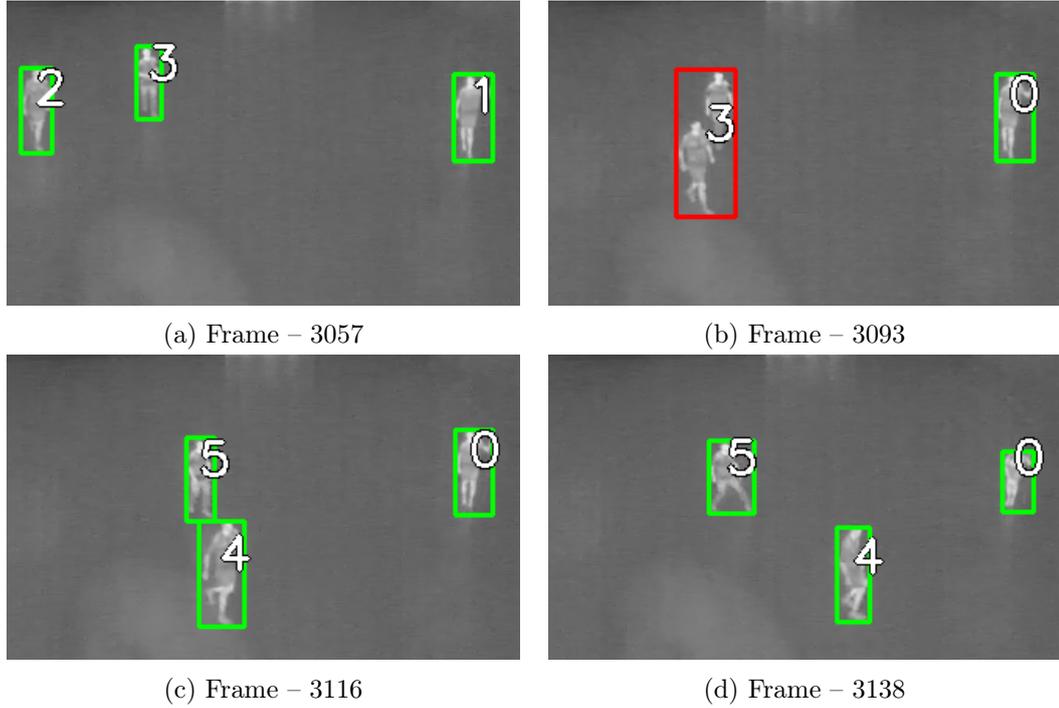


Figure 4.6: (a) first frame of snippet. (b) occlusion starts (c) occlusion stops (d) last frame of snippet. From the first frame to the start of the occlusion takes 35 frames. The occlusion occurs over a span of 23 frames. From the occlusion end to the last frame in the snippet 16 frames occurs. The green marks the boundary boxes of each person or red boundary boxes marks the occlusion and the number indicates their ID dedicated by the system. *Note: All subfigures have been cropped.*

As seen the number of frames before and after the occlusion are sparse, which means that tracklets before and after the occlusion is less than two seconds long.

In this test the row normalized dot product is used, as this proved, in section 3.5, to be superior to the dot product. All though Russell-Rao proved superior than the row normalized dot product it would not make sense to look at it in this example. As all 3 accelerometers are within the snippet, the binarized data hereof will be 1 over the entire snippet. Because of this the result of Russell-Rao will be the same for all 3 accelerometers to each tracklet. The results of the row normalized dot product can be seen in Table 4.4.

Table 4.4: Row normalized dot product for the tracklet to each accelerometer. As seen the occlusion  $ID3$ , is without any numbers as the occlusion is excluded as explained in appendix A.3.

Track. / Acc.	GT	acc0	acc1	acc2	acc3	acc4	acc5	acc6	acc7	acc8
ID0	7	0.00	0.00	0.58	0.00	0.00	0.00	0.00	1.00	0.66
ID1	2	0.00	0.00	0.86	0.00	0.00	0.00	0.00	0.98	1.00
ID2	8	0.00	0.00	0.70	0.00	0.00	0.00	0.00	0.93	1.00
ID3	occ.									
ID4	2	0.00	0.00	0.53	0.00	0.00	0.00	0.00	1.00	0.74
ID5	8	0.00	0.00	0.65	0.00	0.00	0.00	0.00	0.99	1.00

As can be seen in Table 4.4 all of the tracklets have a tendency to get a high normalized dot product when matched with *acc7*. However it can also be seen that *ID0* is matched with all accelerometers the tracklet, but is only a good match with *acc7*. Also note that *ID3* is coloured dark and has no data. This is because the occlusion is excluded from the test.

When performing Hungarian algorithm on the 5 tracklets it is important to notice that *ID0* is present over the entire snippet as seen in Figure 4.6. *ID1* and *ID2* is ending at the same time as these two occlude. The first Hungarian is therefore performed between *ID0*, *ID1* and *ID2*. The result of this is seen in Table 4.5.

Table 4.5: First Hungarian algorithm input matrix. The green cells indicates how the Hungarian algorithm has optimised.

Track. / Acc.	GT	acc0	acc1	acc2	acc3	acc4	acc5	acc6	acc7	acc8
ID0	7	0	0	0.58	0	0	0	0	1	0.66
ID4	2	0	0	0.86	0	0	0	0	0.98	1
ID5	8	0	0	0.70	0	0	0	0	0.93	1

From ground truth seen in Table 4.4 it can be seen that the Hungarian algorithm correctly classify the three tracklets. However since only tracklets, *ID1* and *ID2* ends here, the Hungarian result from these two is accepted. The second Hungarian calculated is between tracklets, *ID0*, *ID4* and *ID5*, as can be seen in Table 4.6.

Table 4.6: Second Hungarian algorithm input matrix. The green cells indicates how the Hungarian algorithm has optimised.

Track. / Acc.	GT	acc0	acc1	acc2	acc3	acc4	acc5	acc6	acc7	acc8
ID0	7	0	0	0.58	0	0	0	0	1	0.66
ID4	2	0	0	0.53	0	0	0	0	1	0.74
ID5	8	0	0	0.65	0	0	0	0	0.99	1

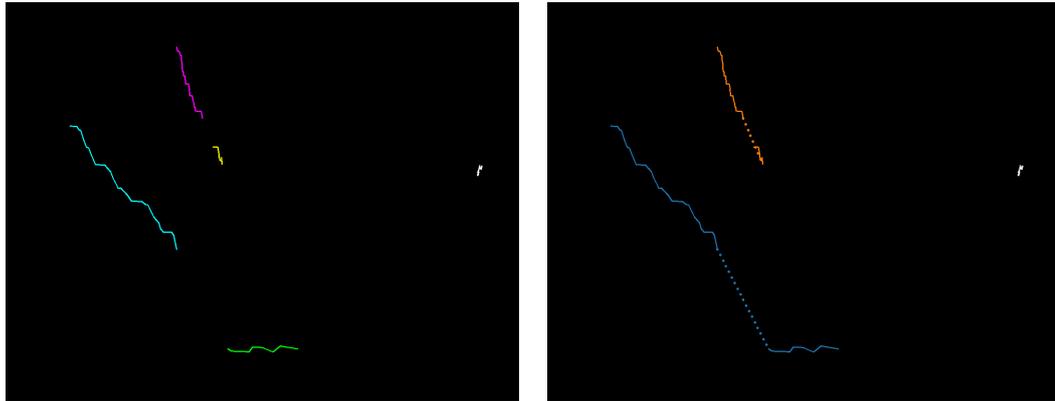
As seen in Table 4.6 the result of the Hungarian does not match with the ground truth. As seen here the no learn method is applied, but if learning method, as explained in section 3.5, Method 4, is applied to the Hungarian results from the first input matrix. The values from *acc2* and *acc8* is removed from *ID0*, which can be seen in Table 4.7.

Table 4.7: Here the results of Hungarian is seen when learning from the first Hungarian is applied.

Track. / Acc.	GT	acc0	acc1	acc2	acc3	acc4	acc5	acc6	acc7	acc8
ID0	7	0	0	0	0	0	0	0	1	0
ID4	2	0	0	0.53	0	0	0	0	1	0.74
ID5	8	0	0	0.65	0	0	0	0	0.99	1

As seen in Table 4.7 it is seen by applying the learning the results of the Hungarian algorithm results in the correct ground truth. In Figure 4.7a the tracklets found by the

tracking, which has been tested in this section is seen. By using the results of the hungarian with learn the 4 broken tracklets can be combined as seen in Figure 4.7b.



(a) The tracklets found by the tracking system are seen. each tracklet are represented by a different colour as their coherence is unknown.

(b) Here tracklets are connected by the results of the Hungarian algorithm. The tracklets are coloured, such that matched tracklets share the same colour.

Figure 4.7: Here the tracklets of the occlusion snippet is seen. (a) Shows the tracklets before the matching is performed. (b) Shows the coherence between tracklets when the matching after the matching is performed.

To be able to make a defining conclusion about if the system is capable of connecting broken tracklets after an occlusion occurs in a convincing matter. It is necessary to have more data to test on, unfortunately the snippet of the Gigantium data, Dataset 2 have really limited scenarios where there are 35+ samples before and occlusion and also after they separated from the occlusion. Instead after the split they tend to occlude within a few samples again.

## 4.5 Conclusions of the Acceptance Test

In this section the results in chapter 4 will be concluded together with the specifications from section 2.6, to control if all the specifications have been kept.

Requirement 1 and 2 states that the detection must be able to segment persons from the background and be able to detect all persons present on the field have been succeeded. In sections 4.1 and 4.2 a snippet of the Gigantium dataset, Dataset 2 have been used which have manually been labelled. All persons are detected correctly on the field, and no false positives have been detected.

Requirement 3-6 states that the system should be able to track a single person, detect when an occlusion occurs, detect when an occlusion separate into individuals (also known as a split) and lastly should be able to if an occlusion separate into individuals, but one of the individuals become a part of a new occlusion.

All these scenarios are presented in the design and implementation, sections 3.2.2 and 3.2.4, where an example of each scenarios is seen. It is also seen in section 4.2, where each tracklet have been manually labelled to ensure a ground truth. This was done by looking through all frames to ensure the tracker is working correctly.

The system must be able to go from a detected pixel coordinates to its corresponding world coordinate, requirement 7. This was designed in section 3.3, but also been proved in section 4.1, where Figure 4.1 shows the movement of all tracklets and trajectories on the field in world coordinates.

Requirement 8 states that the system must be able to calculate an acceleration based on tracklets. The theory behind this calculation is presented in section 2.2.3 and has been implemented into the tracker. In section 3.4 it is seen that the system is able to synchronise the accelerometers with the video data, as stated in requirement 9. The acceleration is used in section 4.2 to make the final test, where the system can match tracklets with the accelerometers. Requirement 10 states that the system should be able to match accelerometers to their corresponding video data, which is successfully done with 42 video tracklets to 9 accelerometers in section 3.5. If the tracklets found by the tracker is used only 31.43 % of the tracklets are correctly matched.

# Conclusion and Discussion

---

# 5

## 5.1 Project Summary

In chapter 1, the project proposal was introduced which lead to an initial problem statement.

The initial problem statement was base for the technical analysis, chapter 2 which looked into previous work in the field with focus on detection and feature-less tracking methods. Then the devices chosen for the project have been examined. The analysis found that the accelerometers are not synchronised between themselves, so this resulted in a need to solve this issue. Furthermore the datasets used in the project have been shown and how the ground truth labelling was conducted. Lastly the technical analysis ended with a problem statement, specification and requirements, used for further design and development.

In chapter 3 a tracking system has been developed which can detect and track individual players in an indoor sport scenario. The tracking algorithm is able to detect occlusions and occlusions separating into individual players. The tracker is able to provide information about each ID: Some of the key informations found for each tracklet is the world coordinates, number of people in the ID and the time span. A method for converting the tracklets into acceleration has been implemented, such that it can be compared to the acceleration measured by the accelerometers mounted on the waist of the players. In order to synchronise the accelerometers and video, a dot product maximisation is used. Once the data is synchronised the tracklet's acceleration from the video can then be matched to the accelerometers data. Several methods have been tested and evaluated for matching the tracklets with the accelerometers, where Russell-Rao provided the best results.

In chapter 4 the system have been tested in parts, first the tracking system have been tested in regards of CLEAR MOT metrics, which gave results telling how robust the tracker is. Furthermore the tracker was tested with all the methods which have been used in the design and implementation chapter 3, these methods have been tested on the chaos scenario, where a snippet of the Gigantium, dataset 2, have been used.

Furthermore the matching between tracklets and accelerometers have been tested on how much of a tracklet is needed to make a correct match.

Lastly a test have been conducted to test if it is possible to connect broken tracklets after an occlusion have ended.

## 5.2 Assessment

The project have investigated if it is possible to develop a system which is able to connect broken tracklets into trajectories, as formulated in section 2.5. This resulted in a pipeline which contains a tracker capable of tracking players in a football match. As seen in the results the tracker is able to detect scenarios of occlusions and the separation of these. The chaotic scene however produce a vast amount of tracklets of varying lengths. When it is attempted to match the tracklets from the tracker using various different methods it is seen that 31.43% of the tracklets are correctly matched with the accelerometers. It is however found that some of the issue with matching these accelerometers is the sparse length of the tracklets. The methods proposed in this project shows a potential in matching inertia sensor data with thermal video data. The group evaluates a better implementation of the tracker can significantly improve the results of the system.

## 5.3 Discussion

In this section further work and different approaches which could have been used in this research is discussed.

**Time Synchronised Devices** In this research unsynchronised devices has been used. In order to create a more optimal system it would required that all devices would be synchronised to the same clock. By having synchronised devices it would also give a better view of how well the system developed would perform.

**Tracking system** Designed in this project is based on contours. Even though the tracker performs as intended it would be beneficial to look into other tracking methods. In previous work at AAU it was seen that both Kalman Filter an Continuous Energy Minimization was used with promising results. In the same work a method for splitting mildly occluded people have been implemented. Splitting these occlusions, could increase the length of the tracklets before the individuals enter the occlusion and therefore it could help with making the tracking system proposed in this report stronger.

**Parent Information** One of the informations which the current tracker is providing is the information about the parents. The thought behind the parent information was to create a sort of family tree. When two people occlude their IDs are saved in the occlusion ID. By creating a family tree it would limit the number of accelerometers which the tracklets can be matched too after the occlusion, since the track before occlusion might have been matched to an accelerometers before the occlusion happens.

**Dataset** The dataset presented in this research displays a very chaotic game of football on half an indoor field. Since the field which the players can move in is limited, occlusions are more likely to occur. It also propose the issue that the players left the camera view, when the ball was accidental shot off the field. A few things could have been done different.

**Scenario Set-up** One of the thing that could have been done was planning a few scenarios in form of planned routines, which would be planned to present occlusions and other scenarios.

**Full Field Recording** One of the reasons that a vast amount of occlusion occurs in the recordings from Gigantium is that the field is limited, which result in the players not being able to run longer distances.

**No Substitutions** Since substitutions happened through the recording and the accelerometers were recording during the time where the players was off the field. In order to prevent these unwanted measurements, it would have been preferred if no substitutions happened. In cases where the ball leaves the field a none participating person would get the ball.

**Graph Matching Methods** In this project three different graph matching methods have been investigated, dot product, normalized dot product and Russell-Rao. For further development alternative graph matching methods could be looked into.

**Accelerometer Sampling-Rate** Should have been higher, so it was possible to make synchronisation by looking at e.g. jump, with 25 Hz it is hard to distinguish the jump itself and synchronise it to video due to the interpolation the sensor chip is doing.



## Bibliography

- [1] Axivity, “Ax3,” <http://axivity.com/product/ax3>, (Accessed on 2017/04/04).
- [2] Berlingske, “"vi måler vores liv med mobilen",” <https://www.b.dk/personlig-udvikling/vi-maaler-vores-liv-med-mobilen>, (Accessed on 2017/02/28).
- [3] “Dagens medicin - self-tracking kan revolutionere forskning og behandling,” <http://dagensmedicin.dk/self-tracking-kan-revolutionere-forskning-behandling/>, (Accessed on 2017/02/27).
- [4] “Trackman A/S, unleash your potential,” <https://trackmangolf.com/products/software>, (Accessed on 2017/02/28).
- [5] “Sport performance tracking tm,” <https://www.sportsperformancetracking.com/#CVOk8m675xwHT6ti.97>, (Accessed on 2017/02/27).
- [6] R. Gade, A. Jørgensen, and T. B. Moeslund, “Occupancy Analysis of Sports Arenas Using Thermal Imaging,” *Proceedings of the International Conference on Computer Vision Theory and Applications*, pp. 277–283, 2012.
- [7] R. Gade and T. B. Moeslund, “Sports type classification using signature heatmaps,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2013, pp. 999–1004.
- [8] R. Gade and T. B. Moeslund, “Thermal tracking of sports players,” *Sensors (Basel, Switzerland)*, vol. 14, no. 8, pp. 13 679–13 691, 2014.
- [9] J. W. Davis, “Robust Detection of People in Thermal Imagery,” pp. 0–3, 2004.
- [10] T. T. Zin, H. Takahashi, and H. Hama, “Robust Person Detection using Far Infrared Camera for Image Fusion,” vol. 1, pp. 1–4, 2007.
- [11] C. Dai, Y. Zheng, and X. Li, “Pedestrian detection and tracking in infrared imagery using shape and appearance,” *Computer Vision and Image Understanding*, vol. 106, no. 2-3, pp. 288–299, 2007.
- [12] E. S. Jeon, J. S. Choi, J. H. Lee, K. Y. Shin, Y. G. Kim, T. T. Le, and K. R. Park, “Human detection based on the generation of a background image by using a far-infrared light camera,” *Sensors (Switzerland)*, vol. 15, no. 3, pp. 6763–6788, 2015.
- [13] O. W. Dietrich, “tyngdeacceleration | gyldendal - den store danske,” [http://denstoredanske.dk/It,\\_teknik\\_og\\_naturvidenskab/Fysik/Relativitetsteori\\_og\\_gravitation/tyngdeacceleration](http://denstoredanske.dk/It,_teknik_og_naturvidenskab/Fysik/Relativitetsteori_og_gravitation/tyngdeacceleration), (Accessed on 2017/05/20).

- [14] Axivity, “How to remove ‘gravity’ (dc component) from the acceleration data?” <http://axivity.com/help/18>, (Accessed on 2017/05/25).
- [15] “Microsoft word - ax3 user guide v1.2.docx,” <http://axivity.com/files/resources/AX3-User-Guide-v1-2.pdf>, (Accessed on 2017-06-07).
- [16] R. Gade and T. B. Moeslund, “Thermal cameras and applications: a survey,” *Machine vision and applications*, vol. 25, no. 1, pp. 245–262, 2014.
- [17] T. B. Moeslund, *Introduction to video and image processing: Building real systems and applications*. Springer Science & Business Media, 2012.
- [18] “Axis q1922 thermal network camera | axis communications,” <https://www.axis.com/dk/en/products/axis-q1922>, (Accessed on 2017-06-07).
- [19] D. Koutra, A. Parikh, A. Ramdas, and J. Xiang, “Algorithms for graph similarity and subgraph matching,” in *Technical report*. Carnegie-Mellon-University, 2011.
- [20] Z. Zivkovic, “Improved adaptive Gaussian mixture model for background subtraction,” in *Proceedings of the 17th International Conference on Pattern Recognition*, vol. 2, no. 2, 2004, pp. 28–31 Vol.2.
- [21] S. Suzuki and K. Be, “Topological structural analysis of digitized binary images by border following,” *Computer Vision, Graphics and Image Processing*, vol. 30, no. 1, pp. 32–46, 1985.
- [22] R. Gade, A. Jorgensen, and T. B. Moeslund, “Long-term occupancy analysis using graph-based optimisation in thermal imagery,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3698–3705.
- [23] “Geometric image transformations — opencv 2.4.13.2 documentation,” [http://docs.opencv.org/2.4/modules/imgproc/doc/geometric\\_transformations.html](http://docs.opencv.org/2.4/modules/imgproc/doc/geometric_transformations.html), (Accessed on 2017-06-07).
- [24] H. W. Kuhn, “The Hungarian method for the assignment problem,” in *50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art*, 2010, pp. 29–47.
- [25] J. Munkres, “Algorithms for the Assignment and Transportation Problems,” *Journal of the Society for Industrial and Applied Mathematics*, vol. 5, no. 1, pp. 32–38, 1957.
- [26] SciPy.org, “scipy.spatial.distance.russellrao — scipy v0.18.1 reference guide,” <https://docs.scipy.org/doc/scipy-0.18.1/reference/generated/scipy.spatial.distance.russellrao.html#scipy.spatial.distance.russellrao>, (Accessed on 2017/05/30).
- [27] K. Bernardin and R. Stiefelhagen, “Evaluating multiple object tracking performance: the clear mot metrics,” *EURASIP Journal on Image and Video Processing*, vol. 2008, no. 1, pp. 1–10, 2008.
- [28] K. Bernardin, A. Elbs, and R. Stiefelhagen, “Multiple object tracking performance metrics and evaluation in a smart room environment,” in *Sixth IEEE International Workshop on Visual Surveillance, in conjunction with ECCV*, vol. 90, 2006, p. 91.

# Appendix

---

# A

## A.1 Matching Between Devices - Data

These tables shows results from matching the videos full trajectory for each person to all 9 accelerometers. The top of each table shows which person's video track is getting matched. Then there are four main columns; First shows which accelerometer the person's video track is matched to. "No Shift"-column, shows the dot product between the match. "Positive Shift"-column, shows both shift and result of dot product maximisation after the video have been shifted from 0-500 samples in positive direction for all accelerometers. "Negative shift"-column, shows the same as positive shift but in negative direction. Furthermore each column shows the variance which consist of the result divided by the best match. The highest dot product for each column is marked in each table.

### A.1.1 Match

In this section it shows matches without any changes to the data.

Table A.1

Video Track for person0								
Acc.	No Shift		Positive Shift			Negative Shift		
	Dot	Var.	Shift	Dot	Var.	Shift	Dot	Var.
acc0	<b>14518.33</b>	1.00	5	<b>16062.74</b>	1.00	6	<b>16133.56</b>	1.00
acc1	10013.18	0.69	86	10534.03	0.66	199	11008.21	0.68
acc2	7064.71	0.49	36	7622.58	0.47	31	7437.08	0.46
acc3	7944.70	0.55	486	10999.28	0.68	111	9632.14	0.60
acc4	4113.92	0.28	15	4402.03	0.27	418	5051.19	0.31
acc5	4156.15	0.29	3	4383.18	0.27	345	4551.41	0.28
acc6	11094.11	0.76	327	12341.12	0.77	126	11805.19	0.73
acc7	8449.22	0.58	384	8861.37	0.55	478	9621.73	0.60
acc8	6033.23	0.42	0	6033.23	0.38	451	6523.80	0.40

Table A.2

Video Track for person1								
Acc.	No Shift		Positive Shift			Negative Shift		
	Dot	Var.	Shift	Dot	Var.	Shift	Dot	Var.
acc0	9246.81	0.91	297	10319.17	0.97	280	10693.90	0.85
acc1	<b>10193.76</b>	1.00	2	<b>10633.86</b>	1.00	109	<b>12642.14</b>	1.00
acc2	5152.05	0.51	15	5500.43	0.52	178	5693.24	0.45
acc3	3980.92	0.39	491	6112.25	0.57	499	4880.09	0.39
acc4	4360.99	0.43	127	4790.54	0.45	413	5747.41	0.45
acc5	3397.53	0.33	16	3563.21	0.34	36	3543.81	0.28
acc6	6745.27	0.66	410	7754.99	0.73	20	7040.93	0.56
acc7	6495.31	0.64	373	7310.07	0.69	181	7318.20	0.58
acc8	5381.63	0.53	369	5734.78	0.54	473	6179.61	0.49

Table A.3

Video Track for person2								
Acc.	No Shift		Positive Shift			Negative Shift		
	Dot	Var.	Shift	Dot	Var.	Shift	Dot	Var.
acc0	9572.46	0.74	371	10779.62	0.77	76	10091.03	0.70
acc1	8565.82	0.66	380	9023.84	0.65	140	8984.40	0.62
acc2	11275.50	0.87	1	11585.11	0.83	120	13600.88	0.94
acc3	<b>12976.63</b>	1.00	10	<b>13916.21</b>	1.00	62	<b>14393.84</b>	1.00
acc4	7280.23	0.56	382	8160.69	0.59	156	7991.78	0.56
acc5	3241.15	0.25	391	3779.66	0.27	490	3683.36	0.26
acc6	11527.48	0.89	9	11554.92	0.83	454	12327.27	0.86
acc7	9350.80	0.72	456	10266.03	0.74	469	10784.84	0.75
acc8	8216.35	0.63	7	8495.86	0.61	206	8544.52	0.59

Table A.4

Video Track for person3								
Acc.	No Shift		Positive Shift			Negative Shift		
	Dot	Var.	Shift	Dot	Var.	Shift	Dot	Var.
acc0	9940.59	0.55	350	10908.90	0.58	25	10636.09	0.43
acc1	6079.78	0.34	485	7616.87	0.41	407	6722.43	0.27
acc2	8366.98	0.46	442	9181.81	0.49	131	10067.61	0.41
acc3	<b>18088.68</b>	1.00	2	<b>18659.26</b>	1.00	73	<b>24559.12</b>	1.00
acc4	4584.85	0.25	249	6027.40	0.32	15	4934.58	0.20
acc5	3616.86	0.20	454	4269.11	0.23	499	3947.38	0.16
acc6	16832.42	0.93	17	16959.97	0.91	96	18956.44	0.77
acc7	9677.12	0.53	466	10507.62	0.56	342	10713.18	0.44
acc8	5968.89	0.33	349	7004.44	0.38	235	6851.99	0.28

Table A.5

Video Track for person4								
Acc.	No Shift		Positive Shift			Negative Shift		
	Dot	Var.	Shift	Dot	Var.	Shift	Dot	Var.
acc0	5351.59	0.99	73	5695.14	0.98	40	5682.25	0.88
acc1	4223.79	0.79	97	5219.35	0.90	6	4383.14	0.68
acc2	3850.50	0.72	231	4153.32	0.72	95	4515.25	0.70
acc3	4822.67	0.90	10	4982.36	0.86	76	5493.74	0.86
acc4	2507.01	0.47	370	3188.75	0.55	22	2579.50	0.40
acc5	859.35	0.16	2	884.97	0.15	494	1424.01	0.22
acc6	<b>5380.56</b>	1.00	11	<b>5783.74</b>	1.00	366	<b>6422.89</b>	1.00
acc7	3366.08	0.63	3	3520.93	0.61	427	4072.24	0.63
acc8	1590.03	0.30	489	2565.87	0.44	11	1629.10	0.25

Table A.6

Video Track for person5								
Acc.	No Shift		Positive Shift			Negative Shift		
	Dot	Var.	Shift	Dot	Var.	Shift	Dot	Var.
acc0	1653.83	0.48	424	2134.74	0.58	372	2199.42	0.54
acc1	1895.30	0.56	19	2095.40	0.57	491	2544.35	0.63
acc2	2010.10	0.59	486	2765.61	0.75	121	2215.17	0.55
acc3	3036.18	0.89	69	3410.58	0.92	26	3372.18	0.83
acc4	1842.45	0.54	349	2166.45	0.59	434	2353.20	0.58
acc5	591.16	0.17	458	648.54	0.18	6	611.41	0.15
acc6	3140.46	0.92	466	3464.75	0.94	6	3164.31	0.78
acc7	2118.22	0.62	194	2673.27	0.72	25	2362.93	0.59
acc8	<b>3413.56</b>	1.00	26	<b>3700.06</b>	1.00	191	<b>4038.75</b>	1.00

Table A.7

Video Track for person6								
Acc.	No Shift		Positive Shift			Negative Shift		
	Dot	Var.	Shift	Dot	Var.	Shift	Dot	Var.
acc0	9452.08	0.69	30	9760.12	0.67	369	10667.23	0.64
acc1	5725.93	0.42	495	6081.57	0.42	467	6484.00	0.39
acc2	6811.65	0.50	96	6857.15	0.47	214	7670.54	0.46
acc3	12250.47	0.90	141	13331.36	0.91	119	14587.63	0.88
acc4	3445.21	0.25	285	4324.61	0.30	1	3490.51	0.21
acc5	3469.00	0.25	69	3582.73	0.25	498	3912.58	0.23
acc6	<b>13642.12</b>	1.00	5	<b>14606.98</b>	1.00	72	<b>16654.60</b>	1.00
acc7	8363.50	0.61	463	8973.07	0.61	104	9704.11	0.58
acc8	3443.00	0.25	107	4115.22	0.28	491	5253.95	0.32

Table A.8

Video Track for person7								
Acc.	No Shift		Positive Shift			Negative Shift		
	Dot	Var.	Shift	Dot	Var.	Shift	Dot	Var.
acc0	11432.71	0.91	142	11966.32	0.88	36	12328.60	0.90
acc1	9667.81	0.77	428	10607.24	0.78	481	10644.01	0.77
acc2	9218.52	0.74	185	10154.48	0.75	489	9671.40	0.70
acc3	11707.23	0.93	495	12958.25	0.95	313	12908.31	0.94
acc4	7008.35	0.56	187	7344.67	0.54	449	7518.60	0.55
acc5	4288.88	0.34	87	4492.26	0.33	241	4800.07	0.35
acc6	<b>12530.91</b>	1.00	407	<b>13600.24</b>	1.00	75	13483.75	0.98
acc7	11836.52	0.94	9	12042.92	0.89	117	<b>13766.39</b>	1.00
acc8	8498.40	0.68	80	8837.49	0.65	432	9365.79	0.68

Table A.9

Video Track for person8								
Acc.	No Shift		Positive Shift			Negative Shift		
	Dot	Var.	Shift	Dot	Var.	Shift	Dot	Var.
acc0	4684.37	0.53	327	5630.38	0.61	123	5613.41	0.48
acc1	5272.46	0.60	11	5565.70	0.60	119	6011.92	0.52
acc2	5657.74	0.65	42	6241.22	0.68	109	6562.39	0.56
acc3	7796.05	0.89	273	8316.93	0.90	99	8566.18	0.74
acc4	5344.54	0.61	363	6402.69	0.69	445	6256.34	0.54
acc5	2582.09	0.29	122	2754.27	0.30	0	2582.09	0.22
acc6	7188.15	0.82	495	8207.96	0.89	5	7770.71	0.67
acc7	6272.59	0.72	332	7246.15	0.79	124	6763.16	0.58
acc8	<b>8758.56</b>	1.00	25	<b>9225.99</b>	1.00	135	<b>11654.01</b>	1.00

### A.1.2 Match with Threshold

In this section it shows matched where the data have been thresholded so if the value is less than 0.2 g it will be set to 0 g. These tables shows results from matching the videos full trajectory for each person to all 9 accelerometers with threshold. The top of each table shows which person's video track is getting matched. Then there are four main columns; First shows which accelerometer the person's video track is matched to. "No Shift"-column, shows the dot product between the match. "Positive Shift"-column, shows both shift and result of dot product maximisation after the video have been shifted from 0-500 samples in positive direction for all accelerometers. "Negative shift"-column, shows the same as positive shift but in negative direction. Furthermore each column shows the variance which consist of the result divided by the best match. The highest dot product for each column is marked in each table.

Table A.10

Video Track for person0								
Acc.	No Shift		Positive Shift			Negative Shift		
	Dot	Var.	Shift	Dot	Var.	Shift	Dot	Var.
acc0	<b>13482.49</b>	1.00	5	<b>15034.02</b>	1.00	6	<b>15090.97</b>	1.00
acc1	8396.24	0.62	86	8896.95	0.59	199	9436.05	0.63
acc2	5593.32	0.41	36	6216.12	0.41	6	6017.64	0.40
acc3	6808.12	0.50	486	9974.14	0.66	111	8530.26	0.57
acc4	3317.35	0.25	72	3651.97	0.24	418	4213.00	0.28
acc5	2651.33	0.20	3	2860.10	0.19	345	3062.89	0.20
acc6	10010.13	0.74	327	11312.18	0.75	126	10747.22	0.71
acc7	7110.61	0.53	384	7574.75	0.50	478	8323.38	0.55
acc8	4975.51	0.37	0	4975.51	0.33	451	5487.94	0.36

Table A.11

Video Track for person1								
Acc.	No Shift		Positive Shift			Negative Shift		
	Dot	Var.	Shift	Dot	Var.	Shift	Dot	Var.
acc0	8348.56	0.90	297	9495.35	0.98	280	9824.50	0.83
acc1	<b>9248.64</b>	1.00	2	<b>9708.80</b>	1.00	109	<b>11818.95</b>	1.00
acc2	3992.54	0.43	15	4383.08	0.45	162	4586.54	0.39
acc3	3135.89	0.34	491	5405.54	0.56	458	4024.81	0.34
acc4	3676.01	0.40	127	4148.92	0.43	413	5086.06	0.43
acc5	2131.09	0.23	16	2307.31	0.24	85	2298.65	0.19
acc6	5776.23	0.62	410	6877.93	0.71	20	6127.30	0.52
acc7	5350.84	0.58	373	6314.82	0.65	181	6283.94	0.53
acc8	4507.73	0.49	369	4945.88	0.51	473	5372.87	0.45

Table A.12

Video Track for person2								
Acc.	No Shift		Positive Shift			Negative Shift		
	Dot	Var.	Shift	Dot	Var.	Shift	Dot	Var.
acc0	7705.20	0.66	371	9020.29	0.72	76	8267.91	0.63
acc1	6055.54	0.52	380	6605.89	0.52	140	6457.80	0.49
acc2	9669.94	0.83	1	9978.49	0.79	132	12064.39	0.92
acc3	<b>11629.38</b>	1.00	10	<b>12606.48</b>	1.00	62	<b>13057.34</b>	1.00
acc4	5901.75	0.51	382	6998.31	0.56	156	6653.62	0.51
acc5	1255.73	0.11	391	1825.35	0.14	490	1688.94	0.13
acc6	10034.62	0.86	9	10074.74	0.80	454	10816.28	0.83
acc7	7615.66	0.65	456	8616.07	0.68	469	9100.29	0.70
acc8	6825.04	0.59	346	7155.79	0.57	187	7159.10	0.55

Table A.13

Video Track for person3								
Acc.	No Shift		Positive Shift			Negative Shift		
	Dot	Var.	Shift	Dot	Var.	Shift	Dot	Var.
acc0	8007.16	0.48	350	9009.55	0.52	25	8749.34	0.37
acc1	3194.52	0.19	485	4842.48	0.28	407	3982.50	0.17
acc2	6438.01	0.38	432	7280.25	0.42	131	8295.81	0.35
acc3	<b>16838.00</b>	1.00	2	<b>17414.73</b>	1.00	73	<b>23514.38</b>	1.00
acc4	3430.56	0.20	249	4938.37	0.28	14	3798.25	0.16
acc5	1585.16	0.09	459	2337.59	0.13	499	2000.26	0.09
acc6	15474.95	0.92	17	15602.96	0.90	96	17769.34	0.76
acc7	7920.80	0.47	441	8780.65	0.50	342	9052.71	0.38
acc8	4542.31	0.27	349	5650.95	0.32	235	5496.08	0.23

Table A.14

Video Track for person4								
Acc.	No Shift		Positive Shift			Negative Shift		
	Dot	Var.	Shift	Dot	Var.	Shift	Dot	Var.
acc0	4738.77	0.98	73	5061.57	0.96	40	5081.62	0.86
acc1	3388.15	0.70	97	4385.79	0.83	6	3537.83	0.60
acc2	3124.56	0.65	231	3413.39	0.65	95	3831.78	0.65
acc3	4261.25	0.88	10	4402.34	0.84	76	4951.19	0.84
acc4	2215.85	0.46	370	2897.63	0.55	22	2277.68	0.39
acc5	34.47	0.01	10	52.21	0.01	494	625.78	0.11
acc6	<b>4831.96</b>	1.00	11	<b>5271.13</b>	1.00	366	<b>5898.95</b>	1.00
acc7	2695.26	0.56	3	2834.98	0.54	427	3443.79	0.58
acc8	1057.13	0.22	489	2077.18	0.39	11	1084.91	0.18

Table A.15

Video Track for person5								
Acc.	No Shift		Positive Shift			Negative Shift		
	Dot	Var.	Shift	Dot	Var.	Shift	Dot	Var.
acc0	1212.62	0.39	424	1723.08	0.50	372	1788.45	0.47
acc1	1368.89	0.44	19	1558.31	0.46	491	2090.76	0.55
acc2	1630.00	0.52	486	2440.84	0.71	121	1846.23	0.49
acc3	2844.90	0.91	69	3224.09	0.94	26	3190.80	0.84
acc4	1407.73	0.45	349	1748.04	0.51	434	1960.78	0.52
acc5	128.21	0.04	458	202.21	0.06	84	141.64	0.04
acc6	2832.45	0.91	466	3171.29	0.93	15	2847.69	0.75
acc7	1723.54	0.55	194	2313.83	0.68	25	1990.17	0.53
acc8	<b>3127.04</b>	1.00	26	<b>3415.64</b>	1.00	191	<b>3781.23</b>	1.00

Table A.16

Video Track for person6								
Acc.	No Shift		Positive Shift			Negative Shift		
	Dot	Var.	Shift	Dot	Var.	Shift	Dot	Var.
acc0	8060.61	0.64	20	8372.94	0.62	369	9289.05	0.59
acc1	3470.85	0.28	495	3928.39	0.29	467	4257.00	0.27
acc2	5246.43	0.42	96	5307.46	0.39	214	6154.02	0.39
acc3	11177.32	0.89	127	12285.15	0.91	119	13537.01	0.86
acc4	2665.37	0.21	285	3732.26	0.28	1	2710.10	0.17
acc5	1882.52	0.15	68	2021.36	0.15	498	2303.75	0.15
acc6	<b>12513.83</b>	1.00	5	<b>13537.67</b>	1.00	72	<b>15660.75</b>	1.00
acc7	6940.11	0.55	463	7662.51	0.57	104	8384.90	0.54
acc8	2279.68	0.18	107	3002.91	0.22	498	4158.22	0.27

Table A.17

Video Track for person7								
Acc.	No Shift		Positive Shift			Negative Shift		
	Dot	Var.	Shift	Dot	Var.	Shift	Dot	Var.
acc0	9632.99	0.87	142	10195.44	0.84	36	10563.96	0.86
acc1	7348.61	0.66	428	8335.09	0.68	481	8389.39	0.68
acc2	7457.78	0.67	185	8449.42	0.69	489	7889.59	0.64
acc3	10376.23	0.93	495	11642.93	0.95	313	11668.48	0.95
acc4	5729.59	0.52	187	6100.73	0.50	449	6219.40	0.51
acc5	2312.49	0.21	392	2558.12	0.21	242	2819.21	0.23
acc6	<b>11104.21</b>	1.00	407	<b>12193.85</b>	1.00	75	12054.95	0.98
acc7	10231.86	0.92	9	10443.72	0.86	121	<b>12259.86</b>	1.00
acc8	7152.98	0.64	80	7539.54	0.62	432	8055.95	0.66

Table A.18

Video Track for person8								
Acc.	No Shift		Positive Shift			Negative Shift		
	Dot	Var.	Shift	Dot	Var.	Shift	Dot	Var.
acc0	3421.90	0.43	327	4410.44	0.53	180	4380.29	0.40
acc1	3667.41	0.46	3	4008.10	0.48	119	4510.87	0.41
acc2	4480.63	0.57	42	5110.15	0.61	109	5393.93	0.49
acc3	7114.00	0.90	103	7650.56	0.91	99	7875.96	0.72
acc4	4384.23	0.56	363	5477.61	0.66	445	5248.46	0.48
acc5	1335.24	0.17	294	1546.68	0.18	0	1335.24	0.12
acc6	6225.28	0.79	495	7287.92	0.87	5	6801.04	0.62
acc7	5167.80	0.65	332	6247.54	0.75	124	5744.61	0.53
acc8	<b>7890.72</b>	1.00	25	<b>8361.31</b>	1.00	135	<b>10899.52</b>	1.00

### A.1.3 Match figures

Figures of all persons video trajectory matched with all accelerometers, can be found in the extra material in the folder "Figures", where each folder is representing one person's trajectory. Inside the folder the match to each accelerometer can be found as a pdf file. For example to see *person6* video trajectory matched to *accelerometer2* can be seen like this: "Figures/person6/person2.pdf".

### A.1.4 Match using Hungarian Algorithm

In this section the intermediate results for the Hungarian Algorithm method is showed. The algorithm steps are explained in section 3.4.1, each step necessary on an example is shown in the upcoming tables, what happens from table to table is explained in its caption. The values used as input matrix are the results of a preliminary test, which shows each step.

Table A.19: The initial matrix.

Traj./Acc	acc0	acc1	acc2	acc3	acc4	acc5	acc6	acc7	acc8
person0	14518.33	10031.18	7064.71	7944.7	4113.92	4156.15	11094.11	8449.22	6033.23
person1	9246.81	10193.76	5152.05	3980.92	4360.99	3397.53	6745.27	6495.31	5381.63
person2	9572.46	8565.82	11275.5	12976.63	7280.23	3241.15	11527.48	9350.8	8216.35
person3	9940.59	6079.78	8366.98	18088.68	4584.85	3616.86	16832.42	9677.12	5968.89
person4	5351.59	4223.79	3850.5	4822.67	2507.01	859.35	5380.56	3366.08	1590.03
person5	1653.83	1895.3	2010.1	3036.18	1842.45	591.16	3140.46	2118.22	3413.56
person6	9452.08	5725.93	6811.65	12250.47	3445.21	3469	13642.12	8363.5	3443
person7	11432.71	9667.81	9218.52	11707.23	7008.35	4288.88	12530.91	11836.52	8498.4
person8	4648.37	5272.46	5657.74	7796.05	5344.54	2582.09	7188.15	6272.59	8758.56

Table A.20: All values are negated, because a profit Hungarian result is wanted.

Traj./Acc	acc0	acc1	acc2	acc3	acc4	acc5	acc6	acc7	acc8
person0	-14518.33	-10031.18	-7064.71	-7944.7	-4113.92	-4156.15	-11094.11	-8449.22	-6033.23
person1	-9246.81	-10193.76	-5152.05	-3980.92	-4360.99	-3397.53	-6745.27	-6495.31	-5381.63
person2	-9572.46	-8565.82	-11275.5	-12976.63	-7280.23	-3241.15	-11527.48	-9350.8	-8216.35
person3	-9940.59	-6079.78	-8366.98	-18088.68	-4584.85	-3616.86	-16832.42	-9677.12	-5968.89
person4	-5351.59	-4223.79	-3850.5	-4822.67	-2507.01	-859.35	-5380.56	-3366.08	-1590.03
person5	-1653.83	-1895.3	-2010.1	-3036.18	-1842.45	-591.16	-3140.46	-2118.22	-3413.56
person6	-9452.08	-5725.93	-6811.65	-12250.47	-3445.21	-3469	-13642.12	-8363.5	-3443
person7	-11432.71	-9667.81	-9218.52	-11707.23	-7008.35	-4288.88	-12530.91	-11836.52	-8498.4
person8	-4648.37	-5272.46	-5657.74	-7796.05	-5344.54	-2582.09	-7188.15	-6272.59	-8758.56

Table A.21: The absolute value of lowest element (18088.68) is added to all elements.

Traj./Acc	acc0	acc1	acc2	acc3	acc4	acc5	acc6	acc7	acc8
person0	3570.35	8057.5	11023.97	10143.98	13974.76	13932.53	6994.57	9639.46	12055.45
person1	8841.87	7894.92	12936.63	14107.76	13727.69	14691.15	11343.41	11593.37	12707.05
person2	8516.22	9522.86	6813.18	5112.05	10808.45	14847.53	6561.2	8737.88	9872.33
person3	8148.09	12008.9	9721.7	0	13503.83	14471.82	1256.26	8411.56	12119.79
person4	12737.09	13864.89	14238.18	13266.01	15581.67	17229.33	12708.12	14722.6	16498.65
person5	16434.85	16193.38	16078.58	15052.5	16246.23	17497.52	14948.22	15970.46	14675.12
person6	8636.6	12362.75	11277.03	5838.21	14643.47	14619.68	4446.56	9725.18	14645.68
person7	6655.97	8420.87	8870.16	6381.45	11080.33	13799.8	5557.77	6252.16	9590.28
person8	13440.31	12816.22	12430.94	10292.63	12744.14	15506.59	10900.53	11816.09	9330.12

Table A.22: The lowest value in each row is subtracted from all elements in row. The subtraction value can be see under the subVal column.

Traj. / Acc.	acc0	acc1	acc2	acc3	acc4	acc5	acc6	acc7	acc8	subVal
person0	0	4487.15	7453.62	6573.63	10404.41	10362.18	3424.22	6069.11	8485.1	(-3570.35)
person1	946.95	0	5041.71	6212.84	5832.77	6796.23	3448.49	3698.45	4812.13	(-7894.92)
person2	3404.17	4410.81	1701.13	0	5696.4	9735.48	1449.15	3625.83	4760.28	(-5112.05)
person3	8148.09	12008.9	9721.7	0	13503.83	14471.82	1256.26	8411.56	12119.79	(-0)
person4	28.97	1156.77	1530.06	557.89	2873.55	4521.21	0	2014.48	3790.53	(-12708.12)
person5	1759.73	1518.26	1403.46	377.38	1571.11	2822.4	273.1	1295.34	0	(-14675.12)
person6	4190.04	7916.19	6830.47	1391.65	10196.91	10173.12	0	5278.62	10199.12	(-4446.56)
person7	1098.2	2863.1	3312.39	823.68	5522.56	8242.03	0	694.39	4032.51	(-5557.77)
person8	4110.19	3486.1	3100.82	962.51	3414.02	6176.47	1570.41	2485.97	0	(-9330.12)

Table A.23: The lowest value in each column is subtracted from all elements in column.

Traj./Acc	acc0	acc1	acc2	acc3	acc4	acc5	acc6	acc7	acc8
person0	0	4487.15	6050.16	6573.63	8833.3	7539.78	3424.22	5374.72	8485.1
person1	946.95	0	3638.25	6212.84	4261.66	3973.83	3448.49	3004.06	4812.13
person2	3404.17	4410.81	297.67	0	4125.29	6913.08	1449.15	2931.44	4760.28
person3	8148.09	12008.9	8318.24	0	11932.72	11649.42	1256.26	7717.17	12119.79
person4	28.97	1156.77	126.6	557.89	1302.44	1698.81	0	1320.09	3790.53
person5	1759.73	1518.26	0	377.38	0	0	273.1	600.95	0
person6	4190.04	7916.19	5427.01	1391.65	8625.8	7350.72	0	4584.23	10199.12
person7	1098.2	2863.1	1908.93	823.68	3951.45	5419.63	0	0	4032.51
person8	4110.19	3486.1	1697.36	962.51	1842.91	3354.07	1570.41	1791.58	0
subVal	(0)	(0)	(-1403.46)	(0)	(-1571.11)	(-2822.4)	(0)	(-694.39)	(0)

Table A.24: It is attempted to cover all zero values with minimum amount of lines. In this case it requires minimum of 7 lines to cover all zeroes. If the minimum is lower than number of rows and columns next step is used.

Traj./Acc	acc0	acc1	acc2	acc3	acc4	acc5	acc6	acc7	acc8
person0	0	4487.15	6050.16	6573.63	8833.3	7539.78	3424.22	5374.72	8485.1
person1	946.95	0	3638.25	6212.84	4261.66	3973.83	3448.49	3004.06	4812.13
person2	3404.17	4410.81	297.67	0	4125.29	6913.08	1449.15	2931.44	4760.28
person3	8148.09	12008.9	8318.24	0	11932.72	11649.42	1256.26	7717.17	12119.79
person4	28.97	1156.77	126.6	557.89	1302.44	1698.81	0	1320.09	3790.53
person5	1759.73	1518.26	0	377.38	0	0	273.1	600.95	0
person6	4190.04	7916.19	5427.01	1391.65	8625.8	7350.72	0	4584.23	10199.12
person7	1098.2	2863.1	1908.93	823.68	3951.45	5419.63	0	0	4032.51
person8	4110.19	3486.1	1697.36	962.51	1842.91	3354.07	1570.41	1791.58	0

Table A.25: The lowest unmarked value, white cells, (28.97) is subtracted from all unmarked values and added to all double marked values, dark blue cells.

Traj./Acc	acc0	acc1	acc2	acc3	acc4	acc5	acc6	acc7	acc8
person0	0	4487.15	6050.16	6602.6	8833.3	7539.78	3453.19	5374.72	8485.1
person1	946.95	0	3638.25	6241.81	4261.66	3973.83	3477.46	3004.06	4812.13
person2	3375.2	4381.84	268.7	0	4096.32	6884.11	1449.15	2902.47	4731.31
person3	8119.12	11979.93	8289.27	0	11903.75	11620.45	1256.26	7688.2	12090.82
person4	0	1127.8	97.63	557.89	1273.47	1669.84	0	1291.12	3761.56
person5	1759.73	1518.26	0	406.35	0	0	302.07	600.95	0
person6	4161.07	7887.22	5398.04	1391.65	8596.83	7321.75	0	4555.26	10170.15
person7	1098.2	2863.1	1908.93	852.65	3951.45	5419.63	28.97	0	4032.51
person8	4110.19	3486.1	1697.36	991.48	1842.91	3354.07	1599.38	1791.58	0

Table A.26: The zeroes are again covered with minimum amount of lines. Again the number of lines is lower than number of rows and procedure is reapplied

Traj./Acc	acc0	acc1	acc2	acc3	acc4	acc5	acc6	acc7	acc8
person0	0	4487.15	6050.16	6602.6	8833.3	7539.78	3453.19	5374.72	8485.1
person1	946.95	0	3638.25	6241.81	4261.66	3973.83	3477.46	3004.06	4812.13
person2	3375.2	4381.84	268.7	0	4096.32	6884.11	1449.15	2902.47	4731.31
person3	8119.12	11979.93	8289.27	0	11903.75	11620.45	1256.26	7688.2	12090.82
person4	0	1127.8	97.63	557.89	1273.47	1669.84	0	1291.12	3761.56
person5	1759.73	1518.26	0	406.35	0	0	302.07	600.95	0
person6	4161.07	7887.22	5398.04	1391.65	8596.83	7321.75	0	4555.26	10170.15
person7	1098.2	2863.1	1908.93	852.65	3951.45	5419.63	28.97	0	4032.51
person8	4110.19	3486.1	1697.36	991.48	1842.91	3354.07	1599.38	1791.58	0

Table A.27: Lowest unmarked value (97.63) is subtracted from all unmarked values and added to double marked values.

Traj./Acc	acc0	acc1	acc2	acc3	acc4	acc5	acc6	acc7	acc8
person0	0	4389.52	5952.53	6602.6	8735.67	7442.15	3453.19	5277.09	8387.47
person1	1044.58	0	3638.25	6339.44	4261.66	3973.83	3575.09	3004.06	4812.13
person2	3375.2	4284.21	171.07	0	3998.69	6786.48	1449.15	2804.84	4633.68
person3	8119.12	11882.3	8191.64	0	11806.12	11522.82	1256.26	7590.57	11993.19
person4	0	1030.17	0	557.89	1175.84	1572.21	0	1193.49	3663.93
person5	1857.36	1518.26	0	503.98	0	0	399.7	600.95	0
person6	4161.07	7789.59	5300.41	1391.65	8499.2	7224.12	0	4457.63	10072.52
person7	1195.83	2863.1	1908.93	950.28	3951.45	5419.63	28.97	0	4032.51
person8	4207.82	3486.1	1697.36	1089.11	1842.91	3354.07	1697.01	1791.58	0

Table A.28: Minimum lines to cover is found to 8.

Traj./Acc	acc0	acc1	acc2	acc3	acc4	acc5	acc6	acc7	acc8
person0	0	4389.52	5952.53	6602.6	8735.67	7442.15	3453.19	5277.09	8387.47
person1	1044.58	0	3638.25	6339.44	4261.66	3973.83	3575.09	3004.06	4812.13
person2	3375.2	4284.21	171.07	0	3998.69	6786.48	1449.15	2804.84	4633.68
person3	8119.12	11882.3	8191.64	0	11806.12	11522.82	1256.26	7590.57	11993.19
person4	0	1030.17	0	557.89	1175.84	1572.21	0	1193.49	3663.93
person5	1857.36	1518.26	0	503.98	0	0	399.7	600.95	0
person6	4161.07	7789.59	5300.41	1391.65	8499.2	7224.12	0	4457.63	10072.52
person7	1195.83	2863.1	1908.93	950.28	3951.45	5419.63	28.97	0	4032.51
person8	4207.82	3486.1	1697.36	1089.11	1842.91	3354.07	1697.01	1791.58	0

Table A.29: Lowest unmarked value (171.07) is subtracted from unmarked values and added to double marked values.

Traj./Acc	acc0	acc1	acc2	acc3	acc4	acc5	acc6	acc7	acc8
person0	0	4389.52	5952.53	6773.67	8735.67	7442.15	3453.19	5277.09	8387.47
person1	1044.58	0	3638.25	6510.51	4261.66	3973.83	3575.09	3004.06	4812.13
person2	3204.13	4113.14	0	0	3827.62	6615.41	1278.08	2633.77	4462.61
person3	7948.05	11711.23	8020.57	0	11635.05	11351.75	1085.19	7419.5	11822.12
person4	0	1030.17	0	728.96	1175.84	1572.21	0	1193.49	3663.93
person5	1857.36	1518.26	0	675.05	0	0	399.7	600.95	0
person6	4161.07	7789.59	5300.41	1562.72	8499.2	7224.12	0	4457.63	10072.52
person7	1195.83	2863.1	1908.93	1121.35	3951.45	5419.63	28.97	0	4032.51
person8	4207.82	3486.1	1697.36	1260.18	1842.91	3354.07	1697.01	1791.58	0

Table A.30: Minimum number of lines is found to 8.

Traj./Acc	acc0	acc1	acc2	acc3	acc4	acc5	acc6	acc7	acc8
person0	0	4389.52	5952.53	6773.67	8735.67	7442.15	3453.19	5277.09	8387.47
person1	1044.58	0	3638.25	6510.51	4261.66	3973.83	3575.09	3004.06	4812.13
person2	3204.13	4113.14	0	0	3827.62	6615.41	1278.08	2633.77	4462.61
person3	7948.05	11711.23	8020.57	0	11635.05	11351.75	1085.19	7419.5	11822.12
person4	0	1030.17	0	728.96	1175.84	1572.21	0	1193.49	3663.93
person5	1857.36	1518.26	0	675.05	0	0	399.7	600.95	0
person6	4161.07	7789.59	5300.41	1562.72	8499.2	7224.12	0	4457.63	10072.52
person7	1195.83	2863.1	1908.93	1121.35	3951.45	5419.63	28.97	0	4032.51
person8	4207.82	3486.1	1697.36	1260.18	1842.91	3354.07	1697.01	1791.58	0

Table A.31: Lowest unmarked value (1030.17) is subtracted from unmarked values and added to double marked values.

Traj./Acc	acc0	acc1	acc2	acc3	acc4	acc5	acc6	acc7	acc8
person0	0	3359.35	5952.53	6773.67	7705.5	6411.98	3453.19	4246.92	7357.3
person1	2074.75	0	4668.42	7540.68	4261.66	3973.83	4605.26	3004.06	4812.13
person2	3204.13	3082.97	0	0	2797.45	5585.24	1278.08	1603.6	3432.44
person3	7948.05	10681.06	8020.57	0	10604.88	10321.58	1085.19	6389.33	10791.95
person4	0	0	0	728.96	145.67	542.04	0	163.32	2633.76
person5	2887.53	1518.26	1030.17	1705.22	0	0	1429.87	600.95	0
person6	4161.07	6759.42	5300.41	1562.72	7469.03	6193.95	0	3427.46	9042.35
person7	2226	2863.1	2939.1	2151.52	3951.45	5419.63	1059.14	0	4032.51
person8	5237.99	3486.1	2727.53	2290.35	1842.91	3354.07	2727.18	1791.58	0

Table A.32: Minimum number of lines is 8.

Traj./Acc	acc0	acc1	acc2	acc3	acc4	acc5	acc6	acc7	acc8
person0	0	3359.35	5952.53	6773.67	7705.5	6411.98	3453.19	4246.92	7357.3
person1	2074.75	0	4668.42	7540.68	4261.66	3973.83	4605.26	3004.06	4812.13
person2	3204.13	3082.97	0	0	2797.45	5585.24	1278.08	1603.6	3432.44
person3	7948.05	10681.06	8020.57	0	10604.88	10321.58	1085.19	6389.33	10791.95
person4	0	0	0	728.96	145.67	542.04	0	163.32	2633.76
person5	2887.53	1518.26	1030.17	1705.22	0	0	1429.87	600.95	0
person6	4161.07	6759.42	5300.41	1562.72	7469.03	6193.95	0	3427.46	9042.35
person7	2226	2863.1	2939.1	2151.52	3951.45	5419.63	1059.14	0	4032.51
person8	5237.99	3486.1	2727.53	2290.35	1842.91	3354.07	2727.18	1791.58	0

Table A.33: Lowest unmarked value (145.67) is subtracted from unmarked and added to double marked.

Traj./Acc	acc0	acc1	acc2	acc3	acc4	acc5	acc6	acc7	acc8
person0	0	3359.35	5952.53	6773.67	7559.83	6266.31	3453.19	4101.25	7211.63
person1	2074.75	0	4668.42	7540.68	4115.99	3828.16	4605.26	2858.39	4666.46
person2	3204.13	3082.97	0	0	2651.78	5439.57	1278.08	1457.93	3286.77
person3	7948.05	10681.06	8020.57	0	10459.21	10175.91	1085.19	6243.66	10646.28
person4	0	0	0	728.96	0	396.37	0	17.65	2488.09
person5	3033.2	1663.93	1175.84	1850.89	0	0	1429.87	600.95	0
person6	4161.07	6759.42	5300.41	1562.72	7323.36	6048.28	0	3281.79	8896.68
person7	2371.67	3008.77	3084.77	2297.19	3951.45	5419.63	1059.14	0	4032.51
person8	5383.66	3631.77	2873.2	2436.02	1842.91	3354.07	2727.18	1791.58	0

Table A.34: The minimum number of lines to cover all zeros is 9. This means that the algorithm can proceed to choose which persons belongs to which accelerometers.

Traj./Acc	acc0	acc1	acc2	acc3	acc4	acc5	acc6	acc7	acc8
person0	0	3359.35	5952.53	6773.67	7559.83	6266.31	3453.19	4101.25	7211.63
person1	2074.75	0	4668.42	7540.68	4115.99	3828.16	4605.26	2858.39	4666.46
person2	3204.13	3082.97	0	0	2651.78	5439.57	1278.08	1457.93	3286.77
person3	7948.05	10681.06	8020.57	0	10459.21	10175.91	1085.19	6243.66	10646.28
person4	0	0	0	728.96	0	396.37	0	17.65	2488.09
person5	3033.2	1663.93	1175.84	1850.89	0	0	1429.87	600.95	0
person6	4161.07	6759.42	5300.41	1562.72	7323.36	6048.28	0	3281.79	8896.68
person7	2371.67	3008.77	3084.77	2297.19	3951.45	5419.63	1059.14	0	4032.51
person8	5383.66	3631.77	2873.2	2436.02	1842.91	3354.07	2727.18	1791.58	0

Table A.35: By first looking at all the zeros which are alone in each row. These zeros is painted green first. The zeros for this example is corresponding to row 1, 2, 4, 7, 8 and 9. Since these zeros are alone in the row the accelerometers which they relate to are therefore chosen for the specific person. Because of this the zeros additional zeros in the same column as a green zero is painted red. Now the zeros that stand alone in the column can be painted green. This is happening in column 5. The additional zeros for the row can now be removed. By continuing in this pattern it can be seen that each accelerometer is matching the person that the zero is showing.

Traj./Acc	acc0	acc1	acc2	acc3	acc4	acc5	acc6	acc7	acc8
person0	0	3359.35	5952.53	6773.67	7559.83	6266.31	3453.19	4101.25	7211.63
person1	2074.75	0	4668.42	7540.68	4115.99	3828.16	4605.26	2858.39	4666.46
person2	3204.13	3082.97	0	0	2651.78	5439.57	1278.08	1457.93	3286.77
person3	7948.05	10681.06	8020.57	0	10459.21	10175.91	1085.19	6243.66	10646.28
person4	0	0	0	728.96	0	396.37	0	17.65	2488.09
person5	3033.2	1663.93	1175.84	1850.89	0	0	1429.87	600.95	0
person6	4161.07	6759.42	5300.41	1562.72	7323.36	6048.28	0	3281.79	8896.68
person7	2371.67	3008.77	3084.77	2297.19	3951.45	5419.63	1059.14	0	4032.51
person8	5383.66	3631.77	2873.2	2436.02	1842.91	3354.07	2727.18	1791.58	0

Table A.36: Here the results of the Hungarian algorithm is shown on the original matrix. As it can be seen all persons are matched with their own accelerometer.

Traj./Acc	acc0	acc1	acc2	acc3	acc4	acc5	acc6	acc7	acc8
person0	14518.33	10031.18	7064.71	7944.7	4113.92	4156.15	11094.11	8449.22	6033.23
person1	9246.81	10193.76	5152.05	3980.92	4360.99	3397.53	6745.27	6495.31	5381.63
person2	9572.46	8565.82	11275.5	12976.63	7280.23	3241.15	11527.48	9350.8	8216.35
person3	9940.59	6079.78	8366.98	18088.68	4584.85	3616.86	16832.42	9677.12	5968.89
person4	5351.59	4223.79	3850.5	4822.67	2507.01	859.35	5380.56	3366.08	1590.03
person5	1653.83	1895.3	2010.1	3036.18	1842.45	591.16	3140.46	2118.22	3413.56
person6	9452.08	5725.93	6811.65	12250.47	3445.21	3469	13642.12	8363.5	3443
person7	11432.71	9667.81	9218.52	11707.23	7008.35	4288.88	12530.91	11836.52	8498.4
person8	4648.37	5272.46	5657.74	7796.05	5344.54	2582.09	7188.15	6272.59	8758.56

## A.2 42 To 9 Trajectories

### A.2.1 Dotproducts for all tracklets matched with each accelerometer

Table A.37: Table showing each tracklet and their dot product with each accelerometer.

Tracklet	acc0	acc1	acc2	acc3	acc4	acc5	acc6	acc7	acc8
0	0.00	0.00	0.00	604.82	0.00	0.00	0.00	0.00	0.00
1	1760.71	1493.07	3723.54	3515.18	0.00	0.00	2984.49	3502.82	3961.70
2	0.00	0.00	130.91	658.67	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	1264.35	1653.77	0.00	0.00	1496.16	618.37	1101.34
4	0.00	0.00	1179.41	1845.11	0.00	0.00	1404.39	539.03	963.75
5	142.09	64.26	1991.73	3083.46	0.00	0.00	2347.30	1889.16	2395.72
6	9167.73	7586.81	8064.99	9911.97	1718.02	583.23	7095.64	11804.52	6220.07
7	101.73	0.00	1241.66	2100.36	0.00	0.00	1815.59	1345.77	1204.15
8	0.00	0.00	405.92	1158.32	0.00	0.00	1242.19	699.75	470.02
9	108.62	0.00	270.21	253.59	0.00	0.00	276.70	287.53	384.98
10	3915.10	3090.30	1241.09	1515.97	0.00	0.00	1311.26	3130.14	2153.26
11	4283.10	3807.79	1200.78	2103.18	3.57	0.00	2242.73	3454.00	1452.89
12	1426.66	1505.27	823.99	0.00	0.00	0.00	0.00	1088.85	1287.77
13	185.61	139.50	57.52	0.00	0.00	0.00	0.00	89.94	58.04
14	2760.70	2352.61	1247.83	4709.41	193.17	0.00	4759.96	4003.54	0.00
15	3259.44	3001.36	1083.89	4772.18	28.78	0.00	5049.56	3927.50	0.00
16	2215.06	1175.25	284.71	822.67	7.14	0.00	1299.30	1168.81	0.00
17	184.98	108.68	855.80	1630.59	27.22	0.00	1354.16	1251.74	0.00
18	2871.50	2762.36	3193.67	1893.48	2470.90	15.64	1718.43	2973.04	178.30
19	129.58	0.00	232.24	267.73	137.57	0.00	286.22	353.75	0.00
20	704.93	1043.37	1059.25	300.49	885.74	0.00	275.91	752.36	0.00
21	2886.37	3062.20	2369.06	117.66	2229.25	0.00	109.82	1503.83	0.00
22	4063.95	6081.45	2672.49	283.91	2689.70	309.39	0.00	2859.10	1929.17
23	2012.68	2780.31	1986.95	0.00	1825.68	5.93	0.00	987.19	94.25
24	402.06	642.86	70.00	0.00	670.54	124.01	0.00	570.69	1309.17
25	1118.13	1446.50	1513.42	2591.95	327.74	591.16	0.00	2118.22	3209.29
26	1068.62	1208.68	0.00	49.75	59.58	189.89	0.00	919.35	1179.79
27	902.89	1244.44	2180.84	3545.27	0.00	661.07	8.73	2416.97	3513.27
28	38.81	217.09	1569.95	3006.01	0.00	401.20	0.00	1352.77	2099.33
29	1400.03	58.88	2934.06	3101.32	0.00	357.90	1498.99	1546.62	1965.18
30	1302.70	0.00	1191.54	1122.28	0.00	19.52	1214.63	752.84	69.33
31	2744.64	0.00	2182.07	2060.86	9.26	0.00	2910.68	1481.47	15.06
32	4412.61	46.59	2157.36	3742.13	39.87	0.00	4243.51	1159.82	149.89
33	1527.92	192.94	712.99	1884.60	15.14	0.00	2026.17	528.50	151.69
34	3114.49	9.27	998.41	3912.46	44.74	0.00	3141.80	631.98	50.41
35	44.49	0.00	16.77	17.64	0.00	0.00	32.11	29.38	0.00
36	315.16	0.00	70.54	362.16	10.33	0.00	373.65	0.00	0.00
37	1348.23	173.64	528.45	1836.09	25.73	0.00	1648.05	0.00	133.10
38	878.97	0.00	214.77	972.47	20.00	0.00	1409.76	0.00	0.00
39	390.70	330.13	408.75	711.70	5.51	0.00	920.16	0.00	268.50
40	88.16	255.38	146.85	16.17	2.87	0.00	463.83	0.00	187.00
41	44.86	304.06	173.97	4.83	1.81	0.00	425.54	0.00	211.16

### A.2.2 Dotproducts for all tracklets matched with each accelerometer, normalized

Table A.38: Table showing each tracklet and their dot product normalized with each accelerometer.

Tracklet	acc0	acc1	acc2	acc3	acc4	acc5	acc6	acc7	acc8
0	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
1	0.44	0.38	0.94	0.89	0.00	0.00	0.75	0.88	1.00
2	0.00	0.00	0.20	1.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.76	1.00	0.00	0.00	0.90	0.37	0.67
4	0.00	0.00	0.64	1.00	0.00	0.00	0.76	0.29	0.52
5	0.05	0.02	0.65	1.00	0.00	0.00	0.76	0.61	0.78
6	0.78	0.64	0.68	0.84	0.15	0.05	0.60	1.00	0.53
7	0.05	0.00	0.59	1.00	0.00	0.00	0.86	0.64	0.57
8	0.00	0.00	0.33	0.93	0.00	0.00	1.00	0.56	0.38
9	0.28	0.00	0.70	0.66	0.00	0.00	0.72	0.75	1.00
10	1.00	0.79	0.32	0.39	0.00	0.00	0.33	0.80	0.55
11	1.00	0.89	0.28	0.49	0.00	0.00	0.52	0.81	0.34
12	0.95	1.00	0.55	0.00	0.00	0.00	0.00	0.72	0.86
13	1.00	0.75	0.31	0.00	0.00	0.00	0.00	0.48	0.31
14	0.58	0.49	0.26	0.99	0.04	0.00	1.00	0.84	0.00
15	0.65	0.59	0.21	0.95	0.01	0.00	1.00	0.78	0.00
16	1.00	0.53	0.13	0.37	0.00	0.00	0.59	0.53	0.00
17	0.11	0.07	0.52	1.00	0.02	0.00	0.83	0.77	0.00
18	0.90	0.86	1.00	0.59	0.77	0.00	0.54	0.93	0.06
19	0.37	0.00	0.66	0.76	0.39	0.00	0.81	1.00	0.00
20	0.67	0.99	1.00	0.28	0.84	0.00	0.26	0.71	0.00
21	0.94	1.00	0.77	0.04	0.73	0.00	0.04	0.49	0.00
22	0.67	1.00	0.44	0.05	0.44	0.05	0.00	0.47	0.32
23	0.72	1.00	0.71	0.00	0.66	0.00	0.00	0.36	0.03
24	0.31	0.49	0.05	0.00	0.51	0.09	0.00	0.44	1.00
25	0.35	0.45	0.47	0.81	0.10	0.18	0.00	0.66	1.00
26	0.88	1.00	0.00	0.04	0.05	0.16	0.00	0.76	0.98
27	0.25	0.35	0.62	1.00	0.00	0.19	0.00	0.68	0.99
28	0.01	0.07	0.52	1.00	0.00	0.13	0.00	0.45	0.70
29	0.45	0.02	0.95	1.00	0.00	0.12	0.48	0.50	0.63
30	1.00	0.00	0.91	0.86	0.00	0.01	0.93	0.58	0.05
31	0.94	0.00	0.75	0.71	0.00	0.00	1.00	0.51	0.01
32	1.00	0.01	0.49	0.85	0.01	0.00	0.96	0.26	0.03
33	0.75	0.10	0.35	0.93	0.01	0.00	1.00	0.26	0.07
34	0.80	0.00	0.26	1.00	0.01	0.00	0.80	0.16	0.01
35	1.00	0.00	0.38	0.40	0.00	0.00	0.72	0.66	0.00
36	0.84	0.00	0.19	0.97	0.03	0.00	1.00	0.00	0.00
37	0.73	0.09	0.29	1.00	0.01	0.00	0.90	0.00	0.07
38	0.62	0.00	0.15	0.69	0.01	0.00	1.00	0.00	0.00
39	0.42	0.36	0.44	0.77	0.01	0.00	1.00	0.00	0.29
40	0.19	0.55	0.32	0.03	0.01	0.00	1.00	0.00	0.40
41	0.11	0.71	0.41	0.01	0.00	0.00	1.00	0.00	0.50

### A.2.3 Russell-Rao for all tracklets matched with each accelerometer

Table A.39: Table showing each tracklet and their results from Russell-Rao when matching to each accelerometer.

Tracklet	acc0	acc1	acc2	acc3	acc4	acc5	acc6	acc7	acc8
0	0.000	0.000	0.000	0.008	0.000	0.000	0.000	0.000	0.000
1	0.072	0.067	0.210	0.118	0.000	0.000	0.103	0.182	0.170
2	0.000	0.000	0.006	0.006	0.000	0.000	0.000	0.000	0.000
3	0.000	0.000	0.050	0.050	0.000	0.000	0.050	0.043	0.048
4	0.000	0.000	0.051	0.051	0.000	0.000	0.051	0.044	0.050
5	0.010	0.004	0.119	0.109	0.000	0.000	0.098	0.114	0.119
6	0.302	0.285	0.417	0.366	0.124	0.134	0.279	0.564	0.295
7	0.006	0.000	0.065	0.065	0.000	0.000	0.053	0.065	0.065
8	0.000	0.000	0.035	0.035	0.000	0.000	0.035	0.035	0.035
9	0.007	0.000	0.022	0.021	0.000	0.000	0.022	0.022	0.022
10	0.121	0.114	0.078	0.049	0.000	0.000	0.048	0.121	0.060
11	0.160	0.177	0.092	0.098	0.011	0.000	0.097	0.177	0.052
12	0.041	0.041	0.041	0.000	0.000	0.000	0.000	0.041	0.041
13	0.009	0.009	0.009	0.000	0.000	0.000	0.000	0.009	0.009
14	0.076	0.090	0.075	0.157	0.077	0.000	0.139	0.157	0.000
15	0.080	0.099	0.041	0.126	0.038	0.000	0.126	0.126	0.000
16	0.036	0.036	0.007	0.036	0.004	0.000	0.036	0.036	0.000
17	0.007	0.014	0.052	0.052	0.048	0.000	0.039	0.052	0.000
18	0.076	0.085	0.142	0.072	0.150	0.007	0.057	0.150	0.009
19	0.004	0.000	0.023	0.023	0.024	0.000	0.024	0.024	0.000
20	0.016	0.011	0.035	0.022	0.035	0.000	0.022	0.035	0.000
21	0.071	0.065	0.069	0.003	0.071	0.000	0.004	0.071	0.000
22	0.142	0.160	0.080	0.013	0.077	0.089	0.000	0.160	0.079
23	0.053	0.060	0.060	0.000	0.060	0.003	0.000	0.060	0.005
24	0.018	0.023	0.005	0.000	0.008	0.022	0.000	0.023	0.023
25	0.065	0.078	0.074	0.072	0.007	0.141	0.000	0.141	0.134
26	0.075	0.075	0.000	0.002	0.004	0.075	0.000	0.075	0.063
27	0.040	0.049	0.075	0.077	0.000	0.116	0.001	0.118	0.118
28	0.002	0.012	0.070	0.075	0.000	0.075	0.000	0.075	0.075
29	0.025	0.007	0.097	0.095	0.000	0.068	0.029	0.097	0.070
30	0.035	0.000	0.041	0.043	0.000	0.004	0.038	0.043	0.006
31	0.087	0.000	0.088	0.086	0.016	0.000	0.090	0.055	0.001
32	0.153	0.005	0.151	0.145	0.086	0.000	0.125	0.053	0.006
33	0.078	0.024	0.100	0.071	0.055	0.000	0.090	0.021	0.025
34	0.117	0.001	0.117	0.117	0.085	0.000	0.089	0.012	0.002
35	0.002	0.000	0.002	0.002	0.000	0.000	0.002	0.002	0.000
36	0.022	0.000	0.022	0.022	0.022	0.000	0.016	0.000	0.000
37	0.064	0.017	0.076	0.060	0.076	0.000	0.061	0.000	0.018
38	0.021	0.000	0.021	0.021	0.021	0.000	0.021	0.000	0.000
39	0.030	0.049	0.075	0.027	0.042	0.000	0.075	0.000	0.050
40	0.006	0.040	0.042	0.002	0.013	0.000	0.041	0.000	0.042
41	0.005	0.046	0.046	0.001	0.016	0.000	0.045	0.000	0.045



## A.3 Matching Tracking Results to Accelerometers

Table A.40: Table 1 out of 2, next table is table A.41, this table is showing each trajectory from the automatic tracking section 3.2 and their ground truth person match. **Method 1** is the best match with dot product. **Method 2** is after an Hungarian with no learn have been applied. **Method 3** is Hungarian with no learn on normalized results. **Method 4** is where Hungarian with learn have been used. **Method 5** is Hungarian with learn on row normalized dot product, **Method 6** is Russell-Rao using Hungarian with no learn. **Method 7** is Russell-Rao using Hungarian with learn

Tracklet	GT	Method 1	Method 2	Method 3	Method 4	Method 5	Method 6	Method 7
0	3	3 ✓	3 ✓	3 ✓	3 ✓	3 ✓	3 ✓	3 ✓
1	2	2 ✓	2 ✓	2 ✓	2 ✓	2 ✓	2 ✓	2 ✓
2	3	3 ✓	3 ✓	3 ✓	3 ✓	3 ✓	3 ✓	3 ✓
3	6	2 ✗	2 ✗	2 ✗	7 ✗	7 ✗	2 ✗	6 ✓
4	8	3 ✗	3 ✗	2 ✗	8 ✓	3 ✗	2 ✗	8 ✓
5	3	3 ✓	3 ✓	3 ✓	3 ✓	8 ✗	3 ✓	3 ✓
6	7	6 ✗	6 ✗	8 ✗	6 ✗	6 ✗	2 ✗	7 ✓
8	3	3 ✓	2 ✗	3 ✓	2 ✗	2 ✗	3 ✓	2 ✗
9	2	3 ✗	3 ✗	3 ✗	3 ✗	8 ✗	3 ✗	3 ✗
12	8	3 ✗	3 ✗	3 ✗	3 ✗	3 ✗	3 ✗	2 ✗
13	6	7 ✗	2 ✗	2 ✗	2 ✗	2 ✗	3 ✗	3 ✗
15	3	7 ✗	7 ✗	7 ✗	7 ✗	7 ✗	3 ✓	2 ✗
16	2	7 ✗	7 ✗	7 ✗	0 ✗	2 ✓	3 ✗	3 ✗
18	3	7 ✗	2 ✗	2 ✗	2 ✗	2 ✗	3 ✓	2 ✗
19	2	7 ✗	7 ✗	7 ✗	7 ✗	3 ✗	3 ✗	3 ✗
20	8	7 ✗	4 ✗	8 ✓	4 ✗	7 ✗	7 ✗	6 ✗
21	6	7 ✗	8 ✗	6 ✓	8 ✗	8 ✗	7 ✗	8 ✗
27	3	7 ✗	8 ✗	7 ✗	8 ✗	7 ✗	3 ✓	2 ✗
28	2	7 ✗	7 ✗	7 ✗	7 ✗	2 ✓	3 ✗	3 ✗
30	8	8 ✓	8 ✓	8 ✓	2 ✗	3 ✗	3 ✗	2 ✗
31	6	8 ✗	7 ✗	7 ✗	7 ✗	7 ✗	6 ✓	3 ✗
32	3	8 ✗	6 ✗	2 ✗	3 ✓	2 ✗	7 ✗	6 ✗
33	2	8 ✗	8 ✗	8 ✗	8 ✗	8 ✗	7 ✗	8 ✗
37	2	2 ✓	2 ✓	2 ✓	8 ✗	2 ✓	6 ✗	3 ✗
39	3	6 ✗	6 ✗	6 ✗	8 ✗	8 ✗	3 ✓	3 ✓
43	3	3 ✓	3 ✓	8 ✗	3 ✓	3 ✓	3 ✓	2 ✗
51	8	3 ✗	6 ✗	3 ✗	2 ✗	3 ✗	3 ✗	2 ✗
52	2	8 ✗	8 ✗	2 ✓	8 ✗	7 ✗	6 ✗	3 ✗
53	3	8 ✗	8 ✗	8 ✗	7 ✗	8 ✗	6 ✗	6 ✗
54	6	3 ✗	3 ✗	6 ✓	3 ✗	7 ✗	3 ✗	8 ✗
56	3	6 ✗	7 ✗	6 ✗	7 ✗	7 ✗	7 ✗	3 ✓
57	2	6 ✗	2 ✓	6 ✗	8 ✗	8 ✗	7 ✗	6 ✗
59	3	6 ✗	7 ✗	6 ✗	7 ✗	7 ✗	7 ✗	3 ✓
60	2	3 ✗	2 ✓	3 ✗	8 ✗	8 ✗	7 ✗	6 ✗
62	3	3 ✓	7 ✗	2 ✗	8 ✗	2 ✗	7 ✗	3 ✓
63	2	6 ✗	2 ✓	2 ✓	7 ✗	8 ✗	7 ✗	6 ✗
66	2	3 ✗	2 ✓	3 ✗	7 ✗	3 ✗	3 ✗	6 ✗
68	3	3 ✓	3 ✓	3 ✓	2 ✗	8 ✗	7 ✗	2 ✗
69	8	3 ✗	7 ✗	2 ✗	7 ✗	3 ✗	3 ✗	8 ✓
74	6	2 ✗	2 ✗	2 ✗	3 ✗	2 ✗	3 ✗	6 ✓
75	3	2 ✗	7 ✗	6 ✗	7 ✗	3 ✓	6 ✗	6 ✗
77	6	6 ✓	2 ✗	8 ✗	3 ✗	2 ✗	3 ✗	8 ✗
78	3	8 ✗	8 ✗	8 ✗	8 ✗	8 ✗	7 ✗	2 ✗
80	2	8 ✗	6 ✗	6 ✗	7 ✗	2 ✗	2 ✓	8 ✗
81	8	3 ✗	8 ✓	3 ✗	2 ✗	3 ✗	8 ✓	6 ✗
82	3	3 ✓	3 ✓	3 ✓	3 ✓	7 ✗	7 ✗	3 ✓
83	6	3 ✗	2 ✗	2 ✗	2 ✗	3 ✗	2 ✗	2 ✗
86	3	3 ✓	6 ✗	6 ✗	6 ✗	7 ✗	2 ✗	3 ✓
87	6	3 ✗	6 ✓	3 ✗	8 ✗	6 ✓	3 ✗	6 ✓
88	2	3 ✗	3 ✗	3 ✗	3 ✗	0 ✗	2 ✓	7 ✗
89	7	3 ✗	8 ✗	7 ✓	8 ✗	6 ✗	6 ✗	2 ✗
92	3	7 ✗	8 ✗	7 ✗	2 ✗	2 ✗	6 ✗	6 ✗
93	2	7 ✗	7 ✗	7 ✗	7 ✗	3 ✗	6 ✗	8 ✗

Table A.41: Table 2 continuation of table A.40, this table is showing each trajectory from the automatic tracking section 3.2 and their ground truth person match. **Method 1** is the best match with dot prouduct. **Method 2** is after an Hungarian with no learn have been applied. **Method 3** is Hungarian with no learn on normalized results. **Method 4** is where Hungarian with learn have been used. **Method 5** is Hungarian with learn on row normalized dot product, **Method 6** is Russell-Rao using Hungarian with no learn. **Method 7** is Russell-Rao using Hungarian with learn

Tracklet	GT	Method 1	Method 2	Method 3	Method 4	Method 5	Method 6	Method 7
94	8	3 ✗	6 ✗	6 ✗	7 ✗	6 ✗	6 ✗	2 ✗
97	2	6 ✗	6 ✗	7 ✗	7 ✗	3 ✗	6 ✗	2 ✓
100	3	7 ✗	7 ✗	7 ✗	7 ✗	6 ✗	3 ✓	2 ✗
104	8	3 ✗	6 ✗	3 ✗	2 ✗	3 ✗	6 ✗	6 ✗
105	3	3 ✓	2 ✗	7 ✗	5 ✗	2 ✗	6 ✗	8 ✗
106	2	7 ✗	6 ✗	7 ✗	6 ✗	7 ✗	6 ✗	3 ✗
108	3	7 ✗	8 ✗	8 ✗	8 ✗	3 ✓	3 ✓	6 ✗
109	6	3 ✗	2 ✗	2 ✗	2 ✗	2 ✗	7 ✗	3 ✗
114	3	8 ✗	2 ✗	8 ✗	6 ✗	8 ✗	7 ✗	8 ✗
115	2	8 ✗	8 ✗	3 ✗	2 ✓	2 ✓	3 ✗	3 ✗
117	3	3 ✓	6 ✗	3 ✓	6 ✗	8 ✗	3 ✓	8 ✗
118	2	3 ✗	6 ✗	6 ✗	7 ✗	6 ✗	3 ✗	2 ✓
121	3	6 ✗	7 ✗	7 ✗	2 ✗	7 ✗	6 ✗	3 ✓
123	8	6 ✗	6 ✗	6 ✗	6 ✗	6 ✗	3 ✗	2 ✗
126	8	2 ✗	2 ✗	2 ✗	2 ✗	3 ✗	7 ✗	3 ✗
127	2	3 ✗	8 ✗	8 ✗	8 ✗	8 ✗	7 ✗	8 ✗
128	3	8 ✗	7 ✗	7 ✗	7 ✗	7 ✗	3 ✓	3 ✓
130	3	3 ✓	2 ✗	2 ✗	2 ✗	2 ✗	7 ✗	8 ✗
131	2	2 ✓	7 ✗	3 ✗	8 ✗	3 ✗	3 ✗	2 ✓
133	8	3 ✗	2 ✗	2 ✗	2 ✗	2 ✗	2 ✗	8 ✓
134	2	2 ✓	6 ✗	6 ✗	6 ✗	6 ✗	7 ✗	6 ✗
135	6	2 ✗	6 ✓	6 ✓	8 ✗	8 ✗	6 ✓	2 ✗
137	3	2 ✗	3 ✓	3 ✓	6 ✗	3 ✓	2 ✗	3 ✓
138	8	7 ✗	6 ✗	6 ✗	6 ✗	6 ✗	3 ✗	2 ✗
141	6	7 ✗	3 ✗	7 ✗	3 ✗	8 ✗	3 ✗	6 ✓
142	3	7 ✗	8 ✗	7 ✗	8 ✗	7 ✗	7 ✗	7 ✗
143	7	7 ✓	2 ✗	2 ✗	2 ✗	2 ✗	7 ✓	8 ✗
144	2	7 ✗	7 ✗	7 ✗	7 ✗	0 ✗	2 ✓	7 ✗
146	7	7 ✓	3 ✗	7 ✓	8 ✗	7 ✓	3 ✗	8 ✗
147	2	7 ✗	0 ✗	0 ✗	3 ✗	0 ✗	0 ✗	0 ✗
154	8	8 ✓	8 ✓	0 ✗	0 ✗	3 ✗	7 ✗	2 ✗
155	0	8 ✗	8 ✗	8 ✗	8 ✗	0 ✓	8 ✗	8 ✗
157	8	8 ✓	8 ✓	8 ✓	2 ✗	6 ✗	0 ✗	0 ✗
158	2	7 ✗	3 ✗	8 ✗	4 ✗	8 ✗	6 ✗	6 ✗
160	1	8 ✗	1 ✓	8 ✗	1 ✓	1 ✓	7 ✗	1 ✓
162	1	7 ✗	2 ✗	2 ✗	0 ✗	1 ✓	1 ✓	1 ✓
163	0	7 ✗	1 ✗	7 ✗	1 ✗	7 ✗	1 ✗	2 ✗
165	1	7 ✗	2 ✗	1 ✓	0 ✗	1 ✓	1 ✓	1 ✓
166	0	7 ✗	1 ✗	1 ✗	1 ✗	7 ✗	1 ✗	2 ✗
171	1	2 ✗	2 ✗	2 ✗	1 ✓	2 ✗	1 ✓	1 ✓
172	0	2 ✗	2 ✗	2 ✗	0 ✓	7 ✗	1 ✗	2 ✗
174	1	7 ✗	2 ✗	7 ✗	1 ✓	7 ✗	0 ✗	1 ✓
175	0	7 ✗	0 ✓	7 ✗	0 ✓	0 ✓	0 ✓	2 ✗
177	8	7 ✗	1 ✗	8 ✓	8 ✓	8 ✓	0 ✗	0 ✗
178	1	7 ✗	4 ✗	7 ✗	4 ✗	7 ✗	7 ✗	1 ✓
179	2	7 ✗	2 ✓	2 ✓	2 ✓	2 ✓	7 ✗	8 ✗
182	1	0 ✗	0 ✗	0 ✗	0 ✗	0 ✗	1 ✓	1 ✓
184	2	1 ✗	0 ✗	1 ✗	2 ✓	7 ✗	1 ✗	2 ✓
185	0	1 ✗	1 ✗	2 ✗	1 ✗	1 ✗	1 ✗	8 ✗
187	1	1 ✓	1 ✓	1 ✓	0 ✗	7 ✗	1 ✗	1 ✓
192	0	0 ✓	0 ✓	0 ✓	0 ✓	1 ✗	0 ✓	1 ✗
198	8	2 ✗	1 ✗	1 ✗	1 ✗	2 ✗	0 ✗	0 ✗