Title: Analysing a data of Pedestrian Dynamics

Name: Sebastian Suwada

Date: 20.05.2023

Short description: Calculating value of real intersection angle between to crossing pedestrian at various theoretical angles and defining deviation angle. Visualization of pedestrian velocity and deviation angle.

1. Introduction

Through the increased process of urbanisation and the rapid growth of the population, we can encounter more and more movement problems in some places since the places are overcrowded. Because of this, studies related to the dynamics of pedestrian movement are being undertaken. In 2016 group of researchers from different fields of study, decided to record sample data of two groups crossing each other in given order of direction. It was performed for 7 different crossing angles, from 0 degree to 180 degree.

Project is divided into three sections – finding crossing angle, calculating velocity and defining deviation angle. Each of the data files provided has crossover coordinates for a given moment in time. In every one of them there are two groups that intersect. One of the most challenging things is to cluster those groups before starting any calculations. The entire set of data was recorded in two days using an infrared camera at 120 Hz. 38 volunteers participated on the first day and 36 on the second day. For all my calculations, graphs and plots or even table I used Visual Studio environment with Python language.

It is necessary to remember that this is a real data, that can include any human factor.

2. Methodology and results

a) Find the crossing angle α for each of the data files and summarize your findings.

First, the data had to be imported into python and prepared for its smooth use. We then extract the start and end values from the data files. Keeping in mind that in each of the files we are dealing with two groups, whose order in the file is not arranged in the respective order for each group. In (fig. 1) it's seen that when data is not clustered, it's necessary to perform clustering algorithm. In this case "**KMeans**" algorithm (fig. 2) was used with restricted values: "**n_clusters** = 2" and "**max_iter** = 300". This process continues until the cluster centers no longer change significantly, or until the maximum number of iterations "**max_iter**" is reached. To check our clustering performance, we can use silhouette score which can give information how well clustering results evaluate. It's ranged from -1 to 1 and best results we obtain when we get value higher 0,5 of silhouette score. In most of the file's after performing KMeans value of silhouette score is higher or close to 0,5.

After developing clusters, it is possible to calculate the barycentre of the assigned groups for the initial value and the final value and find the point of intersection of the lines determined from the two known points.

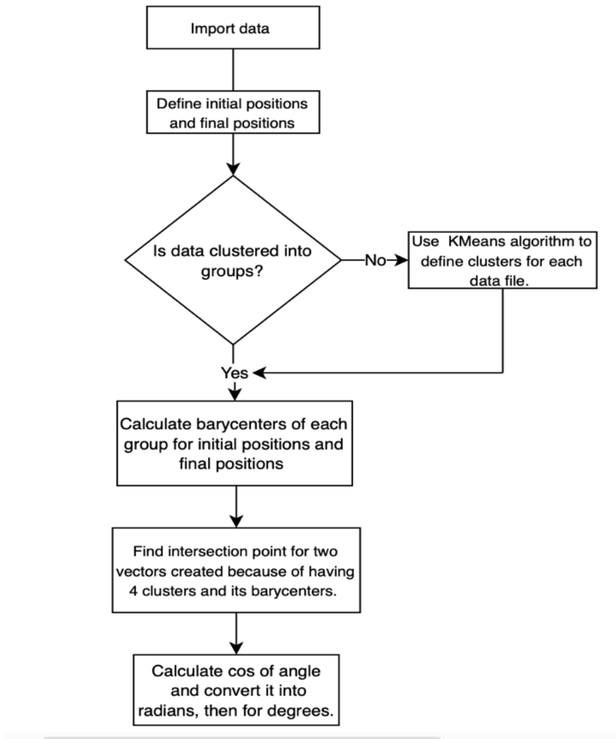

Figure 1 Algorithm for calculating angle of pedestrians crossing.

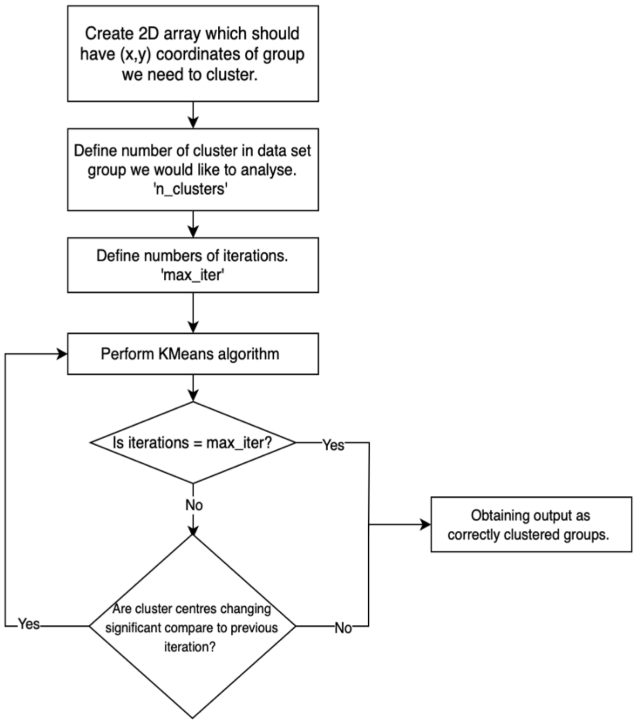

Figure 2 Algorithm of clustering data.

To calculate crossing angle between two sets of groups, below formula is used:

$$\cos(alfa) = \frac{(B1 - INT)dot(B2 - INT)}{\|B1 - INT\| * \|B2 - INT\|}$$

where:

alfa – α – is crossing angle between two lines created from clustered groups

B1 – barycentre of first group

B2 – barycentre of second group

INT – intersection point

dot – product of two vectors

Results:

File_Name	Computed_Angle	File_Name	Computed_Angle
CF2008.csv	153.84597838167053	CF217.csv	90.95821195453296
CF2020.csv	63.99996751071707	CF1012.csv	63.312717645139145
CF116.csv	64.37408078558425	CF1006.csv	25.875741054422175
CF102.csv	0.6581153549827288	CF1007.csv	24.81374251112908
CF103.csv	0.9237548558492596	CF1013.csv	91.31522041191147
CF117.csv	65.07670338658532	CF216.csv	89.78313385475738
CF2021.csv	27.791618323661346	CF202.csv	89.472563774193
CF2009.csv	116.14053807961449	CF214.csv	154.56549793966116
CF2023.csv	26.403467698670603	CF1005.csv	23.88192965365982
CF129.csv	116.85515789405405	CF1011.csv	64.08829367225563
CF101.csv	0.32134139794926553	CF1010.csv	64.37966634047875
CF115.csv	63.72654607262561	CF1004.csv	2.50110628766483
CF114.csv	63.109558964082765	CF201.csv	88.43150953095954
CF128.csv	115.41436678153002	CF211.csv	116.90625863163748
CF2022.csv	26.079371024953794	CF205.csv	0.08574165885300467
CF104.csv	1.103058683649014	CF1028.csv	0.5148044618136083
CF110.csv	26.131850348635087	CF1014.csv	89.74545678254897
CF138.csv	0.3751279293169509	CF1015.csv	90.35647201901423
CF139.csv	179.3265277440958	CF1001.csv	3.5172814086885165
CF111.csv	26.047938496346216	CF204.csv	0.0834979997390136
CF105.csv	0.8736816811465442	CF210.csv	117.92547141541382
CF2019.csv	63.473100157185684	CF206.csv	0.6143941256570251
CF113.csv	63.94773563117542	CF212.csv	153.6407417192251
CF107.csv	26.344488925561887	CF1017.csv	117.51737803782517
CF106.csv	0.37419045616627195	CF1003.csv	1.7126076125389755
CF112.csv	25.587061193223636	CF1002.csv	0.9674980373412171
CF2018.csv	65.09560280566082	CF1016.csv	89.44109071195882
CF2024.csv	26.46724015546619	CF213.csv	154.67563709416456
CF203.csv	89.85632764522875	CF207.csv	0.9138761912965288

File_Name	Computed_Angle	File_Name	Computed_Angle
CF222.csv	63.73079348409858	CF136.csv	154.27771830957815
CF140.csv	0.3201353157892216	CF2014.csv	90.7062884576777
CF141.csv	0.2936529924921281	CF2016.csv	90.23405891803422
CF1026.csv	0.323453624278581	CF2002.csv	0.1619263379871149
CF223.csv	63.19759072788728	CF108.csv	27.702985404953466
CF209.csv	115.67569183562689	CF120.csv	89.91040645921095
CF221.csv	64.72577990048852	CF134.csv	155.03297965631916
CF1024.csv	154.0408676092559	CF135.csv	153.68254044063772
CF1018.csv	116.3146259405969	CF121.csv	90.64757139791149
CF142.csv	0.19007688109941204	CF109.csv	26.132940278363126

CF1019.csv	116.82122696185768	CF2003.csv	0.12065153598446528
CF1025.csv	0.19177407163018131	CF2017.csv	63.622032973446466
CF220.csv	62.336660672169764	CF2013.csv	89.5193588981667
CF208.csv	118.1261753946133	CF2007.csv	153.21332314570336
CF224.csv	25.190517013021832	CF125.csv	116.8160956944927
CF218.csv	90.46284571605901	CF131.csv	153.49212408683837
CF1009.csv	64.38753063169722	CF119.csv	72.45115027317428
CF1021.csv	153.84914760049364	CF118.csv	63.42658490244947
CF1020.csv	116.85538656710754	CF130.csv	116.13771943827223
CF1008.csv	25.093921639534308	CF124.csv	90.1146024755682
CF225.csv	25.392629489040537	CF2006.csv	154.30024883704405
CF227.csv	25.20395842070522	CF2004.csv	0.060884398888885519
CF1022.csv	153.98202755328393	CF2010.csv	117.07226977571248
CF1023.csv	153.99363584641185	CF132.csv	155.02981287679617
CF226.csv	26.04227641032014	CF126.csv	115.86612457318599
CF2001.csv	0.33235901890297953	CF127.csv	118.1457962990417
CF2015.csv	91.01858260924061	CF133.csv	154.47345321338867
CF137.csv	0.1185590914537751	CF2011.csv	116.83654518102283
CF122.csv	90.47904192086118	CF2005.csv	154.7469821461088

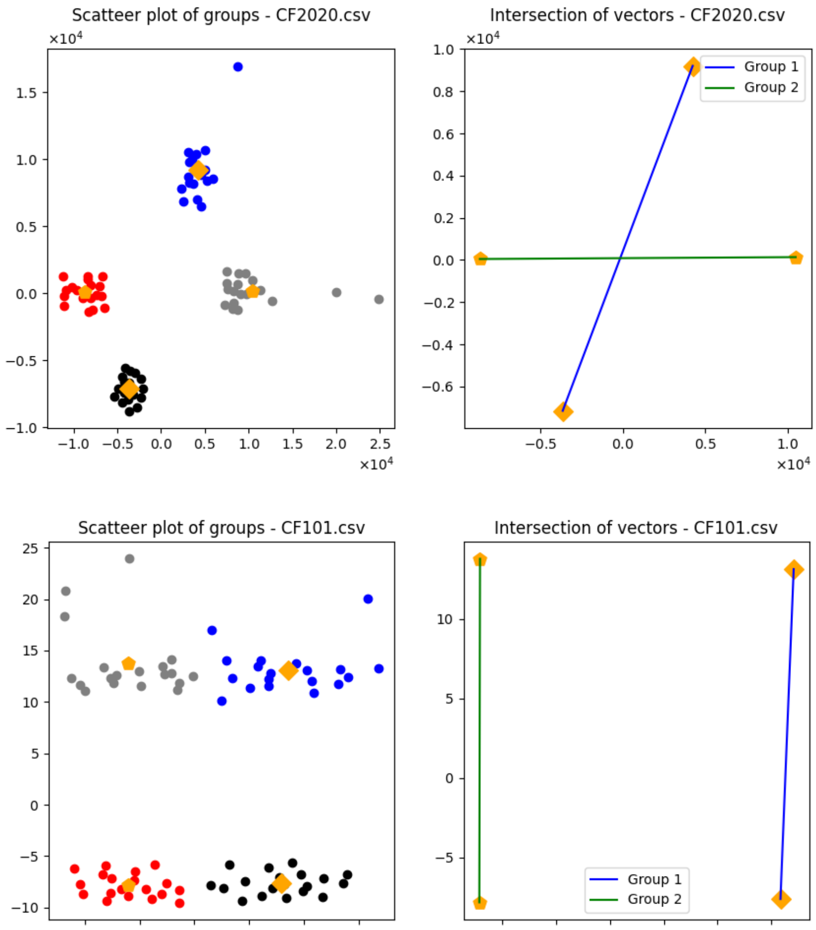

Figure 3 Example of plots. On left picture we can see clustered groups with barycentre and on right side graph with lines connecting its barycentre. On axis X and Y we have position of pedestrian (x,y) defined in [mm].

b) Calculate velocities of all the pedestrians at all instances along their trajectories.

In every file there are 36 or 38 agents. Their every movement is recorded by a sensitive thermal imaging camera at 120 Hz. Which gives 120 positions of a single agent per second. To calculate velocity of every agent in every time, formula shown below is used. Methodology for calculating velocity of pedestrians along their trajectories is less complicated than previous example of finding groups and calculating crossing angle. First of all, it's necessary to three loops one for every file, next for every agent in one file and last one for every tick in csv file. For each agent in every tick, we calculate velocity for every participant using formula shown below.

$$v_i(t) = \left[\frac{\sqrt{(x_i(t) - x_i(t-1))^2 + \left(y_i(t) - y_i(t-1)\right)^2}}{1000} * 120 \right] {}^{m}/_{s}$$

After this we segregate all the values depend on what group it is assigned to. When this step is done it's possible to plot normalised histogram of velocity for every angle group.

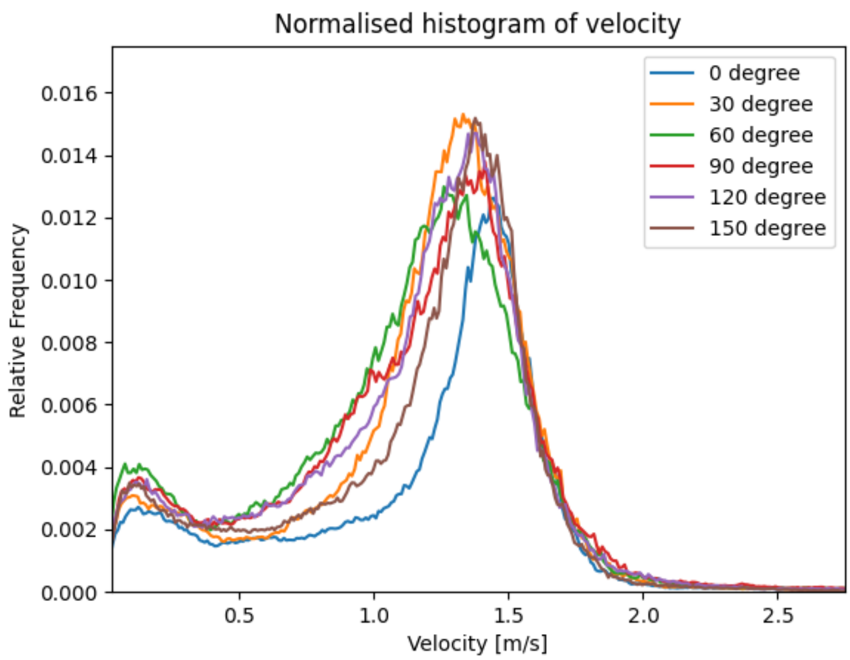

Figure 4 Normalised histogram of velocity for various alfa angles. On x-axis Values in unit m/s, on y-axis relative frequency.

From plot we can see that estimated value of average speed for walking of humans is around 1.3 m/s. Noticeable is the beginning process which we can confidently describe as a start-up of our participants.

c) Define δ as a measure of the deviations that a pedestrian makes with his/her originally instructed direction of motion.

Methodology for calculating deviation angle. The procedure is analogous to calculating the velocity of each agent. First, we obtain the initial and final values of each agent. For each participant, we calculate the ‘**angle_deviat_overall**’, which is the value of the angle on which we will calculate the deviation angle for a given agent in each iteration. Then, for each moment of time, we calculate the angle of deviation similarly to what we did for the ‘**angle_deviat_overall**’ angle using the '**atan2()**' function. Our final deviation angle '**deviation_final**' is the difference between '**angle_deviat**' and '**angle_deviat_overall**'.

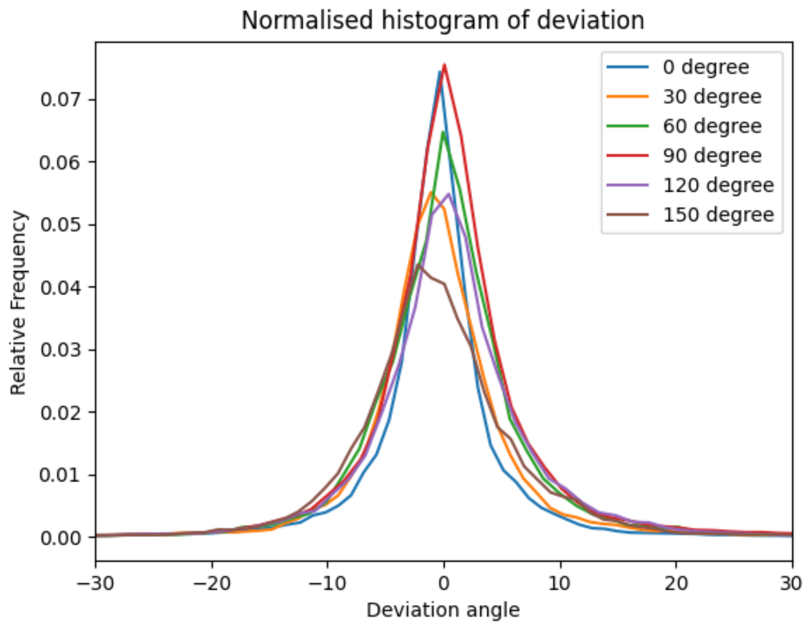

Figure 5 Normalised histogram of deviation angle calculated for every theoretical angle.

The analysis of the provided graph reveals several interesting observations about the deviation angle values. The graph demonstrates that the deviation angle oscillates within the range of -10 degrees to 10 degrees. This indicates that most of the observed data points exhibit relatively small deviations from the reference point. This symmetry suggests that when two groups intersect at a 90-degree angle, they create a reference point in the form of the second group. This reference point likely serves as a guiding influence for the individuals within each group, facilitating a more coordinated movement pattern and resulting in a smoother trajectory.

Conclusion

The task at hand requires an interdisciplinary approach, combining knowledge from various fields. One crucial aspect is the ability to work with visualization tools and programming languages such as Python or MATLAB, which enable us to analyze and interpret the obtained results effectively. It's important to acknowledge that the data used in this research is derived from real-life scenarios, where factors like social movements and casual walking can influence the observed velocities and frequencies. Therefore, it's crucial to account for the inherent variability and common deviations that may arise. Conducting this type of social experiment on real data provides valuable insights into human movement patterns and contributes to our understanding of how individuals navigate and interact within their social environments.