**Issue:** There are already method using sensors for motion capture. However, as can be seen below two pictures, sensors are located on body roughly (picture on the left from stt system and on the right from thesis.

A person standing posing for the camera

Description automatically generatedA person standing in a room

Description automatically generated

**Solution:** by finding the optimised sensor locations, prediction can be optimised.

**Method:**

**First step:** dividing a section of 3D coordinates in 13 sections (body, the upper arms, the below arms, two hands, thigh, calf and food as like the left picture the guy wearing black guy)

**Second step:** set reference sensor which is at the centre of each section and calculate mutual information in terms of the reference sensor

(I am still thinking what I must use between correlation and mutual information because, for example. A = [0,0,1,1], B = [1,1,0,0], A and B are highly dependent in mutual information, but A and B are not correlated if I use correlation)

**Third step:** With N number of independent sensors calculated from second step, I will calculate accuracy between N sensors to find optimized sensor.

Feature: Mean, Variance, Skewness, Kurtosis, Dominant 3frequencies in the DFT, Energy of the 3 dominant frequencies, Max Value, Min Value and Median of the sensor traces made using PCA.

I will use many classification methods such as delta sequence learning, SVM, decision tree etc. and I will score them like ensemble to find the best sensor.

Test data: N number of independent sensors

Training data: sensors in the sections where N number of independent sensors belong to apart from N number of independent sensors.

**Method 2:**

M1 = {sensor 1, sensor2 ………. sensorN}

` M2 = {sensor 1, sensor2 ……... sensorN}

Calculate mutual information and correlation between both sensor 1 from M1 and M2

(mutual information: to see probability distribution between two sensors

Correlation: to see the similarity of both sensors)

And then, score them such as

{Sensor 1 between M1 and M2: correlation: something, mutual information: something

Sensor 2 between M1 and M2: correlation: something, mutual information: something

…

Sensor N between M1 and M2: correlation: something, mutual information something}

After that, I will find the sensor which distinguish M1 and M2 with the biggest difference.

Then I will be able to use classification using the sensor found like in Method1