Data Processing for Eye Tracking with Low Sampling Rate

EECS PG MSC PROJECT - 2017/18

A dissertation in partial fulfilment of the requirements for the degree of

MASTER in BIG DATA SCIENCE

In

QUEEN MARY UNIVERSITY OF LONDON

Submitted by

Shubham Chandel

Student ID: 170801265

Under the Supervision of

Dr. Ildar Farkhatdinov

ACKNOWLEDGEMENT

The study has followed a long and winding path along which I met many people. Although this formal space to acknowledge their contributions is too less, yet I would say that the distribution of my gratitude to all these supporters from different places is proportionate and equitable.

I would like to express sincere gratitude to my teachers, friends and colleagues at Queen Mary University, for their help to make this endeavor possible.

First and foremost, I would like to thank my supervisor Dr. Ildar Farkhatdinov for his guidance and encouragement throughout the process.

This dissertation is completely dedicated to my parents and my sisters. Their constant love, support and motivation throughout have made this task possible.

ABSTRACT

This report explores the eye tracking technology and shows light on the analysis of pupil eye position data which is collected using an eye tracking device connected to a computer. For this report pupil labs headset is used which is head mounted device that contains two cameras; a world camera to capture the gaze and an eye camera to capture the positions of the eye. The data collected through the device is explored using plots and further pre-processing and filtering of the data is done to enhance its knowledge and understanding.

Optical illusions influence a person's mind to believe in the situation which is not even real. Vection which is a kind of optical illusion phenomenon causes disturbance in the pupil gaze positions. Thus an analysis of deviation caused in the pupil position due to vection is done to study the same. An experiment with four participants was performed by collecting the data using the device under both stationary and dynamic setup. Further, data was visualized and deviation for both the setups was calculated to reach the conclusion.

Finally the Conclusion and future works that can be done on eye tracking technology have been discussed in detail.

LIST OF CONTENT

Chapter 1: Overview

1.1 Introduction

When one talks about eye tracking, they mean the measurement of the eye activity. It also includes studying what all attracts our attention? What we tend to look at but ignore? How does the pupil respond to different stimuli? Even though the concept seems to be too basic, the process to achieve this can be quite complex. We are well aware of what significance does gaze holds in the virtual experience.[1] The sampling frequency of an eye tracking device is an important aspect for data processing. Sampling rate refers to the frequency of frames captured per second by the device you're using. High sampling rate makes the process more efficient by tracking the true eye movement positions, given that such devices usually cost much higher as they need better cameras to work with and other relevant features.[2] Thus, decision of the device should depend on your case study. For this project we'll be using pupil headset which has a comparatively low sampling rate.

1.2 Aim & Objective

The aim of the project is to get familiar with eye tracking and understand the pupil labs headset, that is how the eye tracking data is stored, retrieved and its important attributes. It also includes understanding Vection and how visual illusion effects the eye movements.

The main objective of this project is to explore the eye tracking data obtained from the pupil labs headset and analyze the pupil movements.

Following were the goals achieved through the project.

- Pupil Labs application was explored, understanding how the application works, also the various attributes of the data set were studied and explored for the further analysis.
- Data was per-processed to make the plots more clear. Over-sampling on the data set was implemented to increase the frame rate, thus to make the data more expressive and understandable.
- Noise in the data was filtered to remove the disturbance, and make the data smoother.
- Vection was studied and implemented on few participants to understand the effect of optical illusion on eye movements.
- Mean Squared Error and Ttest was performed to analyze the results obtained.

1.3 Literature Review

Eye tracking refers to tracking the movement of the eye positions. Human eyes hold a very important position for completing even the simplest of tasks in our daily life, for example read, write and sight.[3] We don't have the control over all the movements of the eye as it's controlled by the brain. Thus, even a task of reading which may appear simple is not as simple as you may think; it requires gathering the samples and also managing the fixation while doing that.[4] Analyzing the eye began way back in the 19th century, multiple techniques and systems were deployed like Delebarre, Dodge, and Cline.[1] Technological advancement over time allowed eye tracking to emerge and various scientific researches were made using advanced eye tracking systems. Today Research devices are available to analyze the eye positions to understand the point gazed depending upon the use cases.[5] The sampling rate of these devices is important. Higher the sampling rate, better the quality of the data exported from the device as the true pupil positions is better tracked in devices in higher sampling rate.[2]

Chapter 2: Eye Tracking Device

2.1 Hardware

The pupil headset device is a light weighted design carefully done to attain an unostentatious product which is easy to use with a plug and play USB mechanism. The strong and flexible frame is the result of 3d printing that holds 2 cameras; one to record your field of view and the other to capture the eye movement.[6]

It is a portable eye tracking headset and is provided with an infrared (IR) spectrum eye camera which recognizes the dark pupil and the other camera is for observing the field. Both cameras connect to the display devices via high speed USB 2.0. The video streams are read using Pupil Capture software for real-time pupil detection, gaze mapping, recording, and other functions.

2.1.1 Additional Parts

<table>
<tr>
<td>World Camera

Figure1. The front image of the Pupil headset, showing the front world camera.[3]</td>
<td>The high speed 2d world camera is provided with two lenses. A 60 degree FOV lens and the other lens of 100 degree FOV for wide angles. The lenses of the world camera are adjustable, so as to give you a choice to set the field of view as per the requirement. In case the lenses are changed, you need to re-calibrate the camera to update intrinsic. Otherwise the 3d calibration and accuracy test will not work properly![7]</td>
</tr>
</table>

Arm extender  Figure 2. Arm Extender. [4]	The pupil headset device, when ordered is provided with an extendable arm as a part of the package. This arm can be used if you need to adjust the eye cameras further away than the built in regulation range. For adjustment, slide the current eye camera arm off the headset. Slip the arm extender onto the triangular base rail on the headset frame. Slide the camera onto the extended mount rail. Plug the camera back in. This eye camera arm extender works for all existing 120 and 200 Hz systems.
Nose pads  Figure 3. Nose Pads. [5]	All the pupil headsets come with 2 different sets of nose pads. These can be customized as per every individuals fit so as the regulate comfort while wearing.

2.1.2 Pupil Headset Adjustment

A lot of creative and technical thoughts have been given while designing the headset so as to make it just right for the user. It is designed to fit comfortably and securely. The headset and the cameras can be well customized to assist a wide range of users.

While using the device, one should make sure that all the cameras are in focus to a good field of view, so as to ensure a precise eye tracking performance.

Slide Eye Camera

The eye camera can be slid along the track. It can be pushed in and out of the frame.[6]

Rotate Eye Camera

The eye camera is connected to the eye camera arm via a ball joint about which the camera can be rotated.

The Ball Joint Set Screw

The ball point set screw helps you to secure the position of the camera and making it still, so that it does not moves when you move your head. Although, it is recommended that you keep it slightly loose so that it can be adjusted by your hand if needed.

2.2 Software

The source code is open source, and anyone can make changes to enhance the program, also share the work.

Pupil software has two main parts

- **Pupil Capture:** Captures run time video stream of the world and the eye, and
- **Pupil Player:** Used to play the video recorded and export the gaze data recorded by the Pupil Capture.

The software is compatible to run on Linux, Windows and MacOS operating systems.

2.2.1 Pupil Detection Algorithm

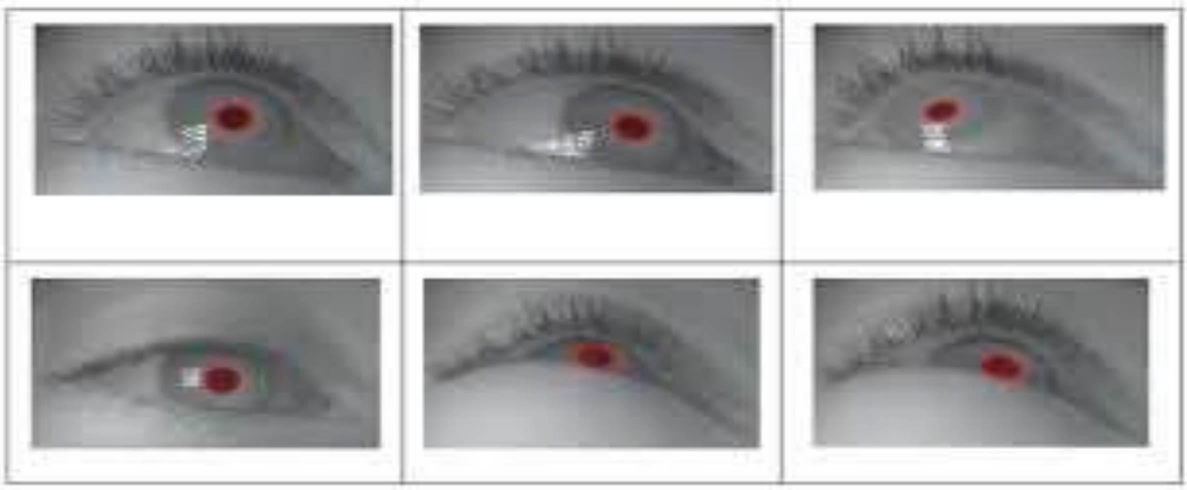

Figure 4. Pupil eye detection

The function of the pupil detection algorithm is to locate the dark pupil in the IR illumination eye camera image. The algorithm is vigorous against reflections in the pupil area in most of the environments. Moreover, the algorithm does not get hampered by the corneal reflection for detection of the pupil, and works even if the user is wearing contact lenses and eyeglasses. It first converts the eye camera image in to grayscale. The preliminary region assessment of the pupil is found via the strongest response for a center-surround feature as proposed by Swirski et al. [8]

2.2.2 Gaze Mapping

Gaze mapping is implemented using two functions of adjustable degree .The function parameters for users are obtained by running calibration routines obtained by two markers; screen and manual or natural features. The underlying models can be easily replaced and also modified according to the need.

2.2.3 Surface Detection

Markers are used to detect the planar surfaces identified in the corresponding scene using the makers. There are 64 such unique markers present and only one such unique marker is required to define an identified surface. Mapping of the gaze positions into the corresponding surface coordinate is done using homographic transformations between the corresponding surface and the screen camera plane.[9]

2.3 Building the Environment (Getting Started)

To run the pupil headset device thought your device you first need to install the pupil app running Linux, MacOS or Windows.

The developers are always working on few features, bug fixing and improvements. Therefore, you should make sure to visit the release page frequently to download the latest versions as on their launch.[10]

2.3.1Capture workflow

By going thought the following data you can get familiar with the workflow of the pupil device.

2.3.1.1 Put on pupil

Wear the pupil headset device and plug it on to your computer. Make sure there is gap between your forehead and the pupil device. The headset can be adjusted and also comes with additional parts.

2.3.1.2 Check pupil detection

When you'll look at the eye window, you may notice a red circle around the edge of your pupil and a red dot at the center of the pupil on its detection.

In case the confidence of the algorithm's detection is high, the circle will be opaque. Whereas, if the confidence is weak the circle will be converted into transparent.

You should experiment with the device by moving you head around a bit while looking at your eye. This way you can make sure that the pupil is effectively detected in various orientations.

2.3.1.3 Calibrate

To know what the person is looking at, you must establish a mapping gaze and pupil position. This is called as calibration.

To start the calibration either click of the "c" on the world screen or press "c" of your keyboard. While keeping your head stationary, follow the marker on the screen.

2.3.2 Record and Export

Pupil capture saves the world video streams along with all this corresponding gaze data to its default location i.e. in a folder in your directory named "recording"

- Start recording: to start you can either press the "r" button from your keyboard or on the screen you can find a press key on the left hand side of the world camera.

Figure5. Pupil player Record hot key

- The recording time will appear next to the record button on your screen.
- Stop recording: to do so, you again need to click on the "r" button on the screen or keyboard.

In the figure 2.3.2 you can see that every new recording is saved with a numeric label like 000, 001, 002 etc.

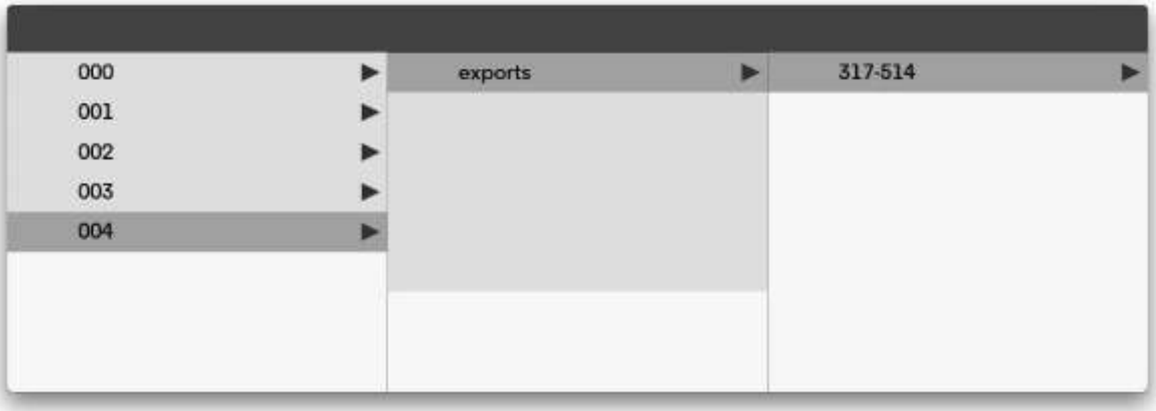

Figure6. Folders containing recorded files

- Data can then be exported using the Pupil Player by simply dragging and dropping the recorded file folder on the pupil player and an export folder will be created in the previous recorded folder(refer fig.7) which will contain the exported data with all the attributes explained later in the report.

Figure7. Pupil Player importing recoded folder window

Chapter 3: Project Details

3.1 Data set

You can export the data from the pupil labs application by pressing the ‘e’ key on the pupil player window. All video files that are appropriate will be exported and are stored in the exports folder, separated from the raw data. Each time you press the export key a new folder will be created inside the exports directory for example: 000 was your first exported file 001 is your second and so on.

Few files are created by default when export key is pressed, these are:

- **export_info.csv:** Contains the information regarding the creation date and data format.
- **pupil_position.csv:** This file contains the pupil data and its related attributes. Some of the attributes that are used for the analysis are:
  - norm_pos_x : containing the normalized x coordinates of the pupil
  - norm_pos_y : containing the normalized y coordinates of the pupil
  - Confidence: determining whether pupil was detected by the device (values closer to 1) or not (values closer to 0), and
  - Timestamp.
- **gaze_positons.csv:** This file contains gaze data and its related attributes.
- **pupil_gaze_positions_info.txt:** This file contains the documentation explaining the attributes for the pupil_positions.csv file and gaze_positions.info file.
- **world_viz.mp4:** This file contains the exported world video by the world camera of the headset.

3.2 Python libraries

Python provides various useful built in libraries and packages that offer wide range of facilities to the user. Mathematic calculations, scientific formulas, statistical calculations can be easily carried out by simply importing the relevant python library and it's corresponding method. Below are mentioned a brief description about some of the important python libraries used for the report.[11]

Pandas

Pandas is python package that involves different data structures making the data processing more effective and reliable. It's quite easy, intuitive, fast and flexible. It is regarded as very high quality tool for real world data analysis.

NumPy

NumPy is a another useful python package that provides N dimensional array which is a very efficient way to store data. Apart from this it provides powerful methods to perform arithmetic calculations like addition, subtraction, multiplication etc.

Matplotlib

Matplotlib is the most common tool used for visualizing the data through the plots. It includes fast and interactive functions for plotting the data and thus exploring the data effectively.

SciPy

SciPy is a python package built for scientific and technical computing, it contains packages for linear algebra, integration, differentiation, interpolation, signal and other useful science and engineering related packages.[12]

3.3 Project Implementation

3.3.1 Data Retrieval

Required packages are imported first.

Csv files are read using the pandas pd.read_csv command.

```
import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

from scipy.signal import butter

from scipy.signal import filtfilt

%matplotlib inline

plt.rcParams["figure.figsize"] = [8,6]
```

3.3.2 Data Preprocessing

- The timestamp attribute is processed so that time starts from 0.0 timestamp. This helps in avoiding the timestamp to start from the local PC time where the video was taken and exported.
- **Time Interpolation:** Data was filtered removing the disturbance caused due to the confidence variability and the values where confidence was low were filtered out. A threshold was put for the confidence and values below that threshold were replaced by the mean. Below is the snipped code of the function used to achieve the same.[13]

```
for ind in pupil[pupil["confidence"]<0.5].index:
  if ind-1 > 0 and ind+1 < pupil.shape[0]:
    pupil.ix[ind, "norm_pos_x"] = pupil['norm_pos_x'].mean()
    pupil.ix[ind, "norm_pos_y"] = pupil['norm_pos_y'].mean()
    pupil.ix[ind, "confidence"] = pupil['confidence'].mean()
```

3.3.3 Oversampling

Oversampling is a process to increase the data sample to improve the quality of the data. Noise in the data can then be explored and filtered out more efficiently. Pupil headset originally gave 30 frames per second, as seen in the snapshot below.

Frames per second

```
In [38]: # Frames per second

         diff = pupil['timestamp'].diff()
         mean = diff.mean()
         fs = 1/mean
         fs

Out[38]: 30.000784305720796
```

Figure7. Original Frame Rate

Frames rate were increased using the pandas interpolate function and linspace method. Below is the code used to achieve the result.[13]

```
f = interpolate.interp1d(pupil['timestamp'], pupil['norm_pos_x'])
num = 12000
xx = np.linspace(pupil['timestamp'].values[0],pupil['timestamp'].values[-1], num)
yy = f(xx)
# yy = np.linspace(pupil['timestamp'].values[0],pupil['timestamp'].values[-1], num)
plt.plot(pupil['timestamp'], pupil['norm_pos_x'], 'bo-')
```

```
plt.plot(xx,yy, 'g.-')

plt.show()
```

3.3.4 Noise

Noise in the data can be referred as the additional meaningless data points in the data set. These corrupt data points often make the data set more complex and may even decrease the quality of the data. Thus the analysis will not be as accurate and meaningful. [14]

The pupil data set also had some noise; hence panda's rolling average function was used to filter out the meaningless data. The rolling average calculates the average of the values occurring within a certain window, you can tune the effect of filtering on the data by sampling increasing or decreasing the window in the pandas.rolling_mean function.[15]

Below is the relative plot for normal data with the filtered (smooth) data for the x coordinate.

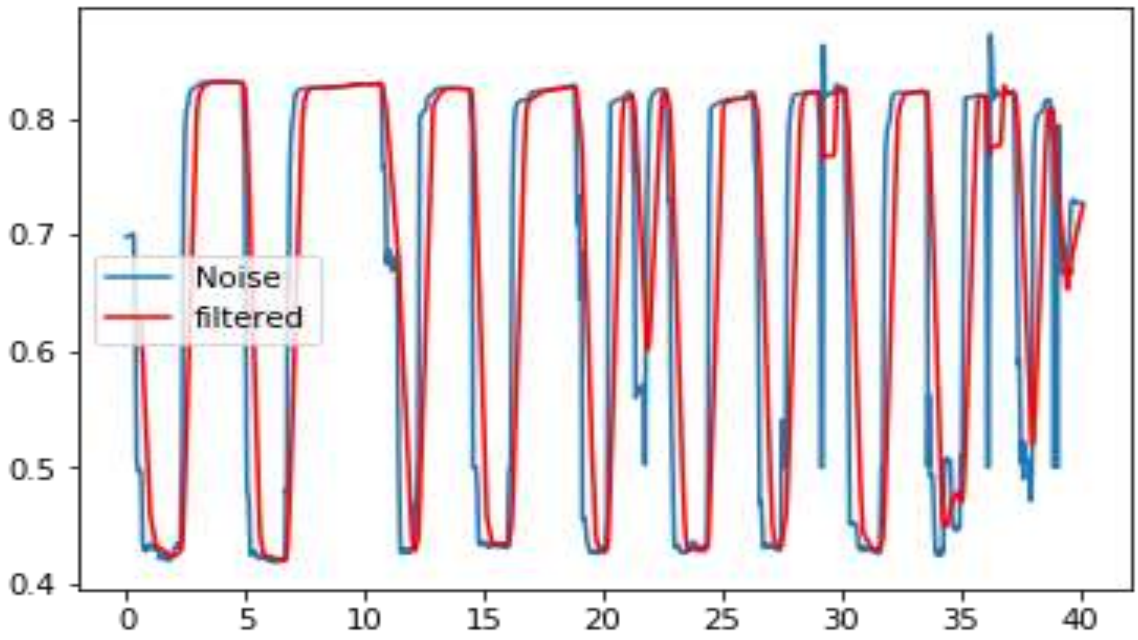

Figure8. Noise Vs filtered data for the coordinate

Below is the relative plot for normal data with the filtered (smooth) data for the y coordinate.

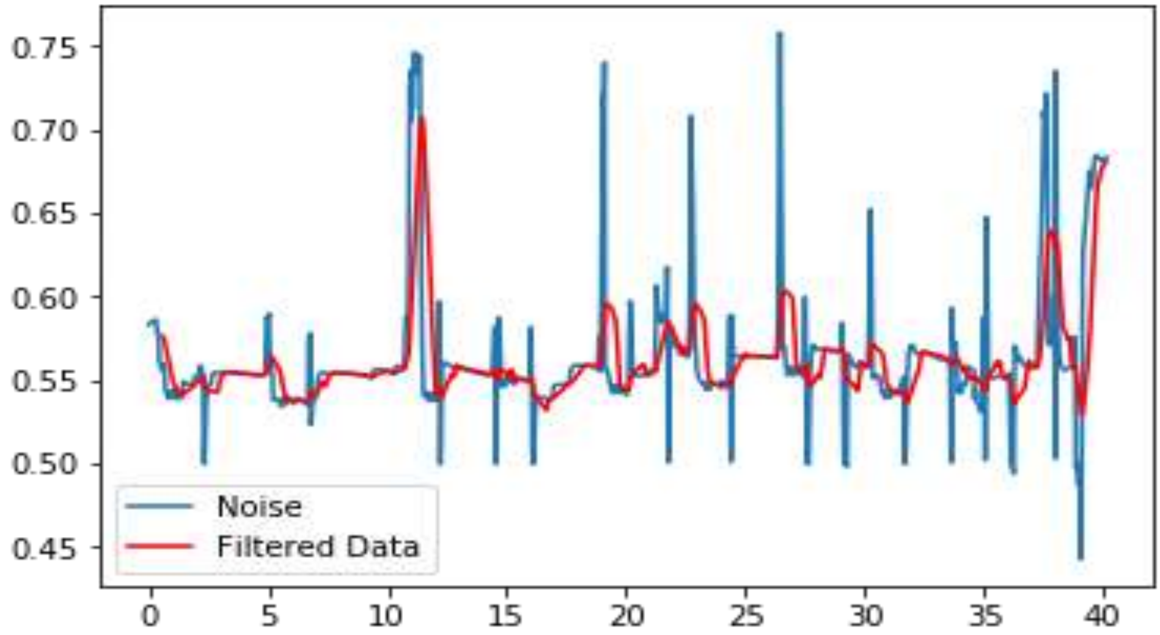

Figure9. Noise Vs filtered data for the Y coordinate

3.3.5 Vection

Vection is the phenomenon that makes a person feel sensations of movement when actually there's been no movement in their surroundings. This can be referred to a self-illusion of motion caused through optical illusions, for instance a watching a moving train from window of a stationary train.[16]

Common optical illusions include: linear, circular, and maybe roll. During a linear vection the observer feels as if he is moving back and forward, while in a circular vection the observer feels he is moving in a circle and similarly in a roll vection the observer feels he is been rolling.

For the analysis purpose, vertical vection slide is used which has black and white vertical strips that moves forward.

3.3.6 Mean Square Error

Mean Squared Error is the squared error deviation of estimated points from the expected value. It's a process of gathering and analyzing the data using formulas to calculate and summarize the statistical analysis.

Mean, Mode and Median are some centrality measures in statistics, that provide the central positions of a given distribution.

Mean Squared Error analysis was performed on the pupil data of 4 participants to get some knowledge of deviation caused through optical illusion. Sk-learn.mean_squared_error function was implemented to get the desired results.

Formula to calculate Mean Squared Error:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (\hat{X_i} - X_i)^2$$

3.3.7 Ttest

Ttest is a two sided test that compares the two samples and their averages and tells whether the corresponding samples have identical (expected) averages. It assumes that the two distributions have identical averages.[17]

We can use this test, if we observe two independent samples from the same or different population. The test measures whether the average (expected) value differs significantly across samples. When you perform a T test in statistics, a p-value helps you determine the significance of your results.[18]

A significance level of 0.05 is used to determine whether our null hypothesis is rejected or not, so if we observe a p-value;

- Larger than 0.05 or 0.1, then we cannot reject the null hypothesis of identical average scores.
- If the p-value is smaller than the threshold, e.g. 1%, 5% or 10%, then we reject the null hypothesis of equal averages.[19]

Chapter 4: Results and Analysis

4.1 Understanding the plots

4.1.1 Pupil left-Right movement

Analysis was performed using the pupil device to explore the data for the left-right pupil movements. For the analysis the head was kept stationary and only pupil was moved left-right (~1sec) and with a fixation of 2 sec for 10 times.

The Following plots were filtered by putting a threshold on the confidence attribute to reduce the noise in the data due blinking of eye or any other reason.

1. TIME & X coordinate: The following graph contains filtered normalized x coordinates of pupil with timestamp.

As seen in the graph, X coordinate changes when the pupil is turned to another direction. It shows a constant movement during the fixation period of pupil.

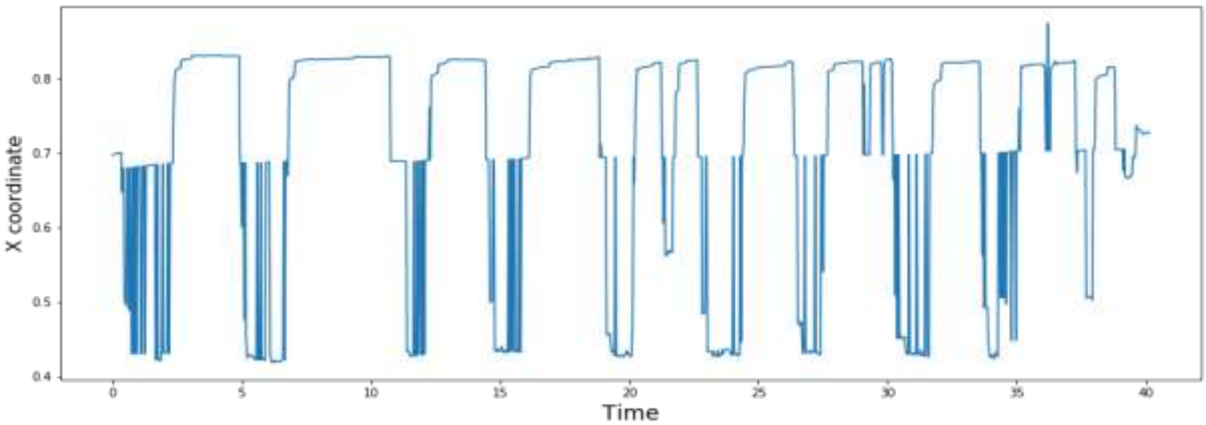

Figure10. X coordinate Vs Time plot

2. TIME & Y coordinate: The following graph contains normalized y coordinates of pupil with timestamp.

As seen from the graph, Y coordinate remains unchanged most of the time as pupil moves only horizontally. However there is noise which could be due to unwanted distortion in the movements of pupil detected by the device.

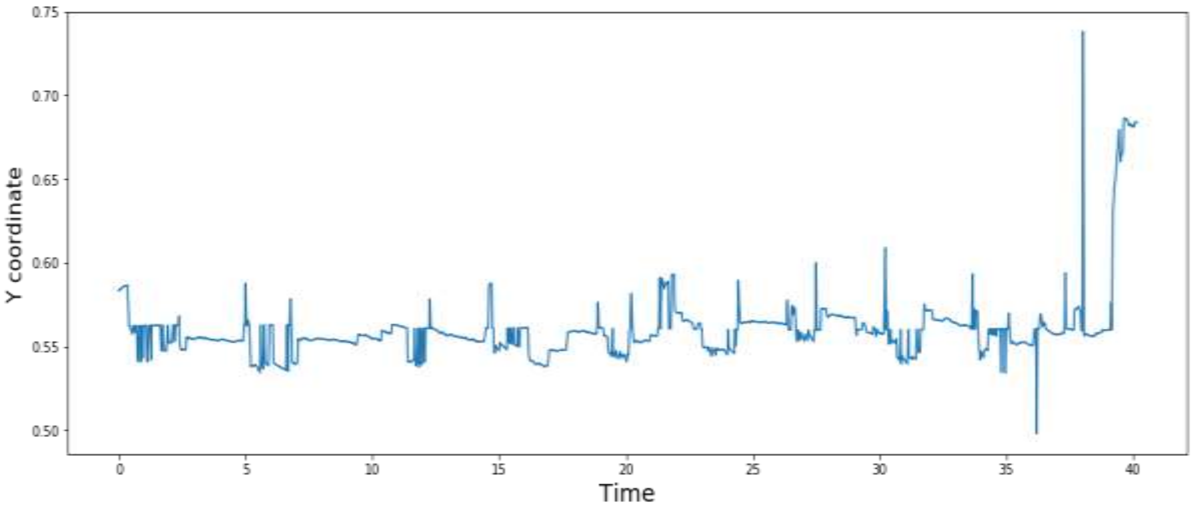

Figure11. X coordinate Vs Time plot

2. X & Y coordinate: The following graph contains normalized X&Y coordinates of pupil.

In this graph, the distortions in the horizontal movements of pupil are exaggerated. The graph shows multiple horizontal lines, since the movement of the pupil was only in the horizontal direction. But due to manual errors the graph is not as precise as it should have been.

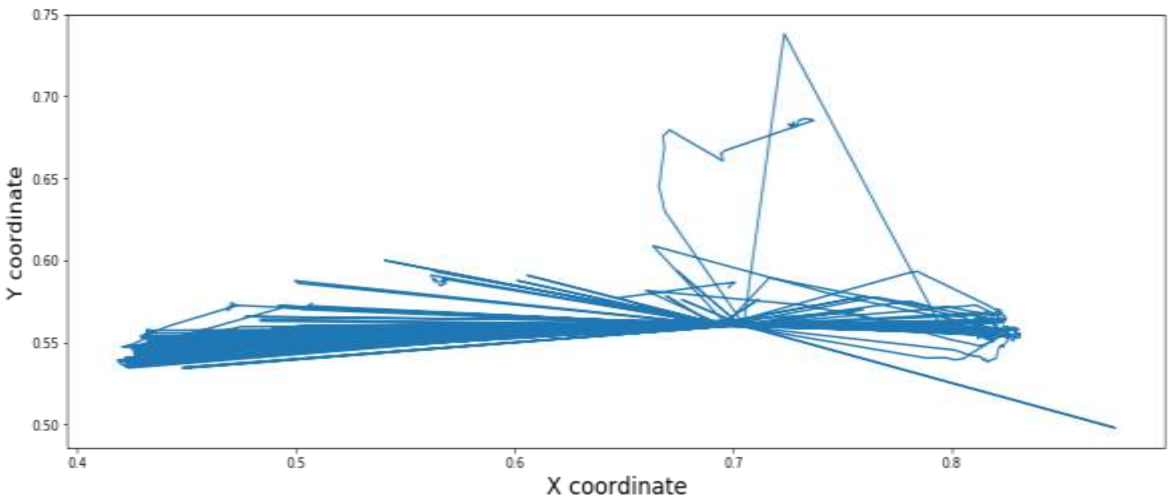

Figure12. X coordinate Vs Y plot

3. Over-sampling:

The data was over-sampled to increase the frame rate, below are the relative graphs for the over-sampled.

Original sample Vs Over-sampled: Pandas interpolate method was used to over sample the data.

Below is the relative graph for over-sampled X coordinates vs. the original.

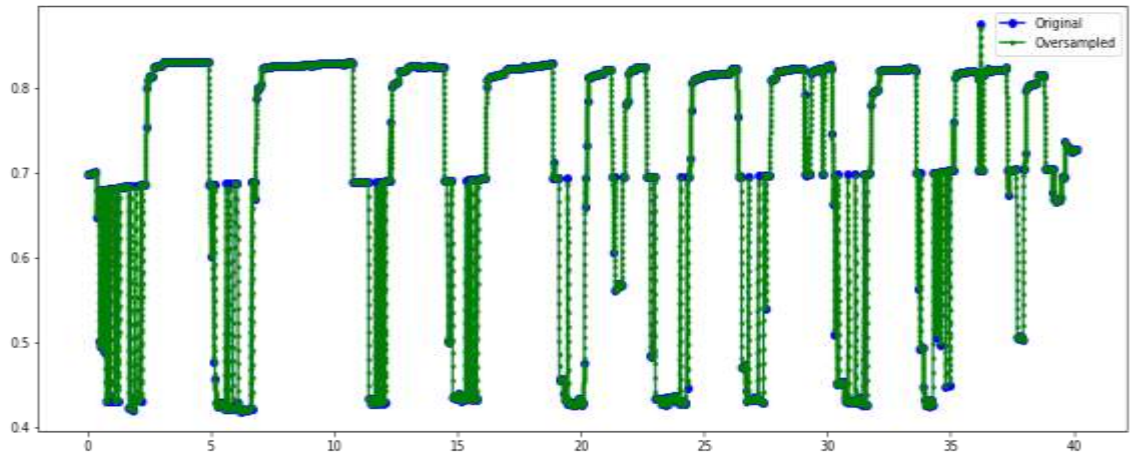

Figure13. X coordinate original sample Vs over-sampled

Below is the relative graph for over-sampled Y coordinates Vs the original.

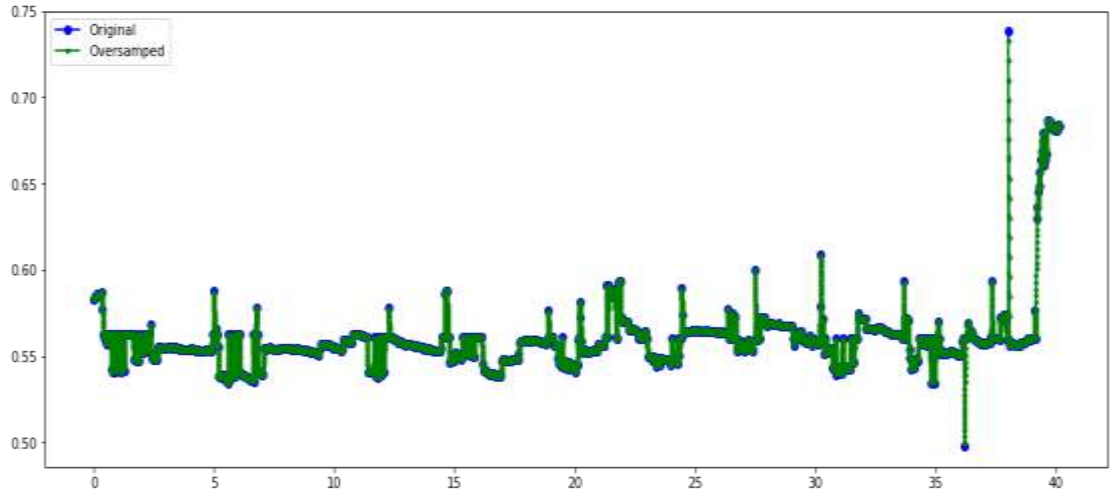

Figure13. Y coordinate original sample Vs over-sampled

4. Data Filtering: Data was smoothed using the pandas rolling window function, that provides rolling window calculations.

Below is relative graph for smoothed x coordinates. As seen in the graph, the filtered data (red line) is smoothened by the rolling function and thus decreasing the noise.

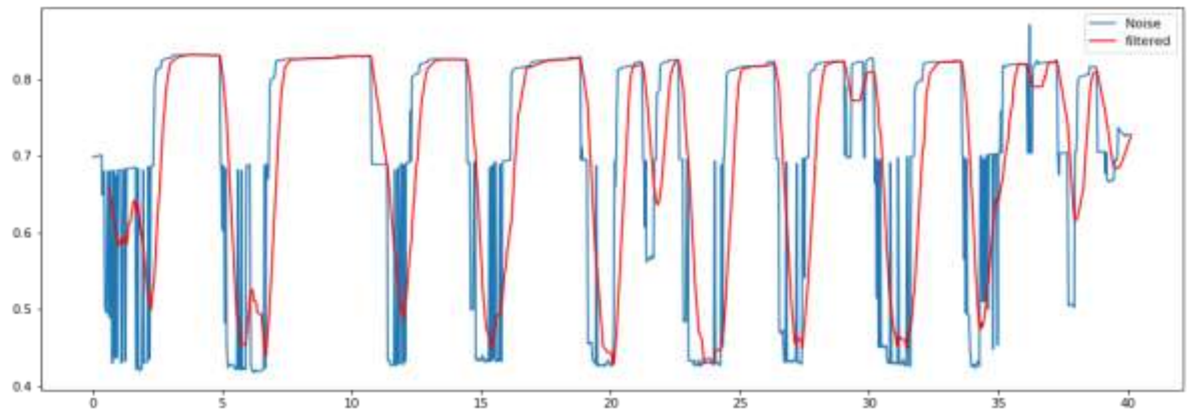

Figure14. X coordinate Noise Vs Filtered

Below is a relative graph for smoothed y coordinates.

Clearly, the filtered data (red line) is smoothened by large degree, thus reducing huge amount of noise.

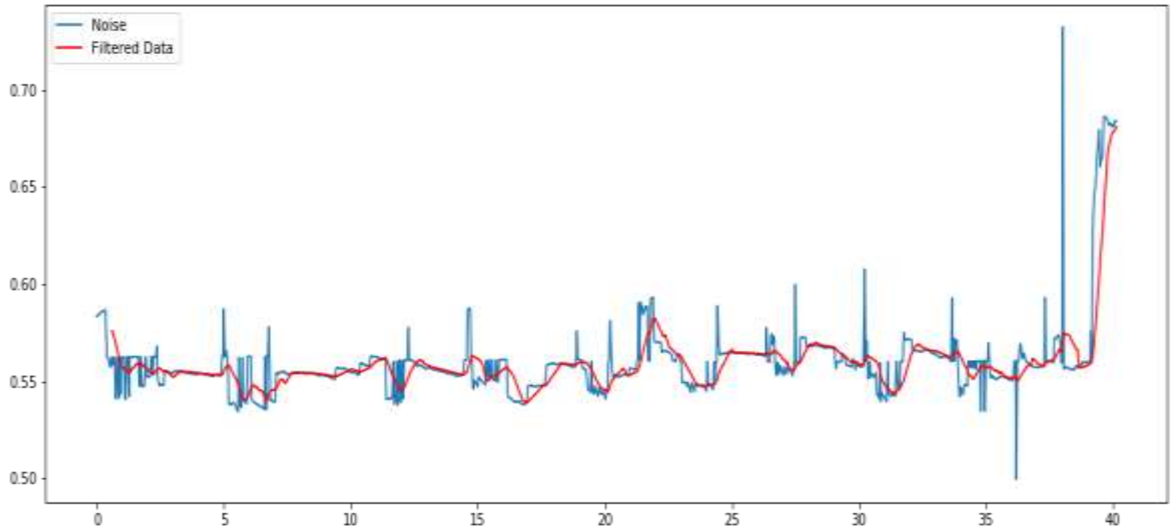

Figure14. Y coordinate Noise Vs Filtered

5. Derivatives: Provides the differential value for a specific function.

- **DX:** dx/dt is a differential function of an element x that changes with respect to time.
- **DY:** dy/dt is a differential function of an element y that changes with respect to time.

Below are the velocity plots.

DX

As seen from the below graph, during the fixation time the velocity in X direction is close to zero and it shoots as soon as the pupil moves in the other direction.

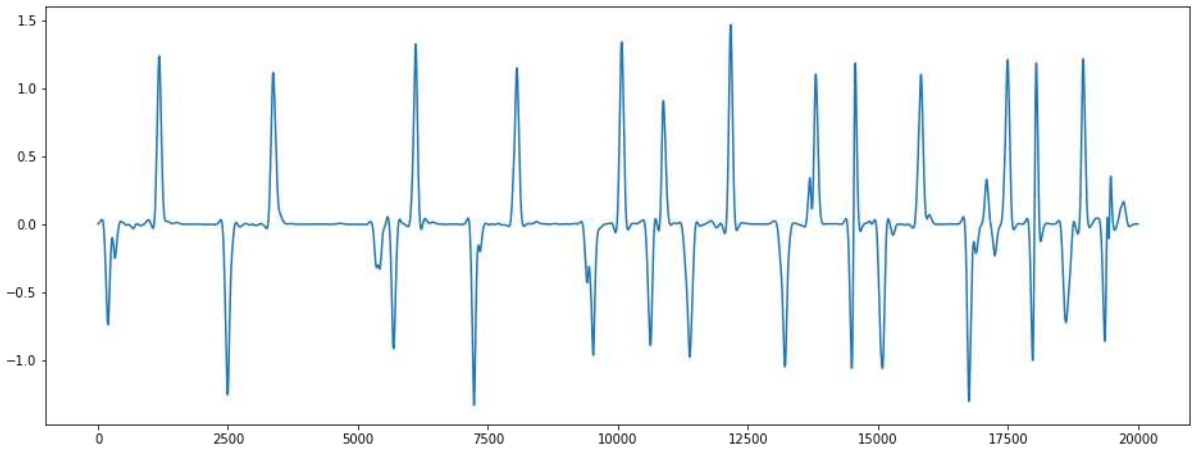

Figure15. Velocity in the X direction

DY

As seen from the below figure, the velocity in Y direction shoots up and down non-uniformly due to unwanted distortion in Y direction. If it was an ideal horizontal movement, it would have been zero throughout.

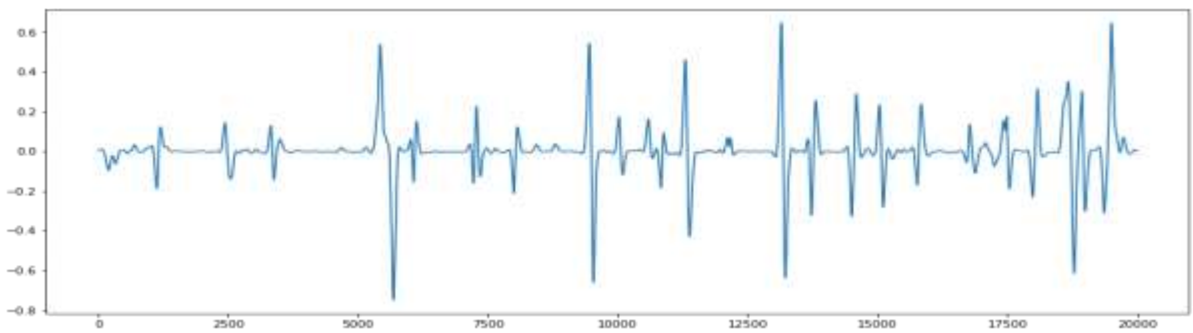

Figure16. Velocity in the Y direction

4.2 Experiment

In this section, experiment has been explained to find the deviation between static and dynamic setup and the results are discussed.

Analysis was conducted on 4 participants who were asked to gaze on a particular marked point on the screen for 60 seconds continuously **(Static)** and then a vertical vection slide was played on the same screen while the participant continues to gaze at the specific marked point for 60 secs **(Dynamic).** Mean Squared Error and ttest was conducted on each participant's pupil data to further understand the effect.

4.2.1 Participant A

Below are the plots observed and analysis for participant A Static(without vection) vs Dynamic(with vection).

1. Time Vs X coordinate

In the following graph, it can be seen that participant A was able to focus well on the target point while the vection slide was running., as both static and dynamic plots are quite similar.

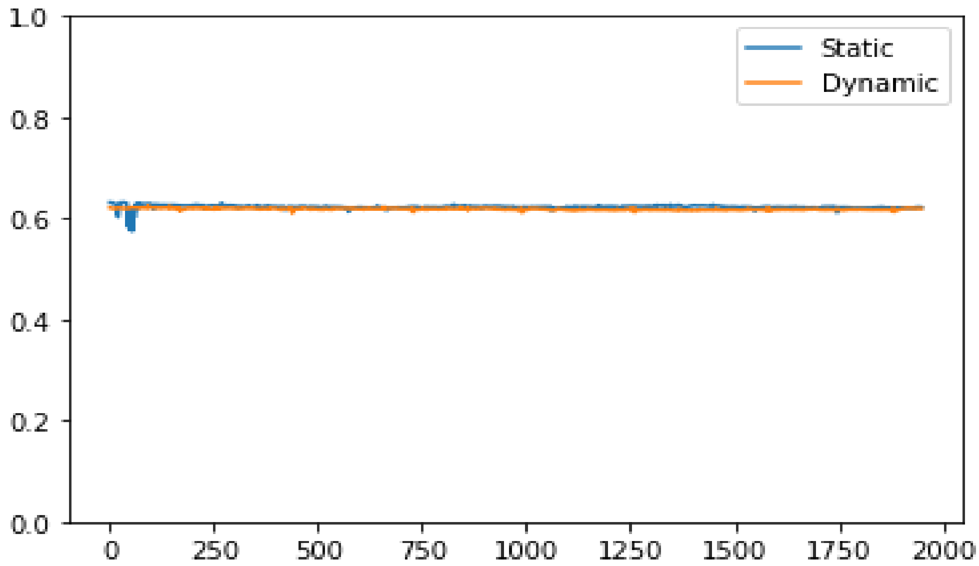

Figure17. Time Vs X coordinate plot

2. Time & Y coordinate

In the following graph y coordinate for the static and the dynamic setup have been analyzed.

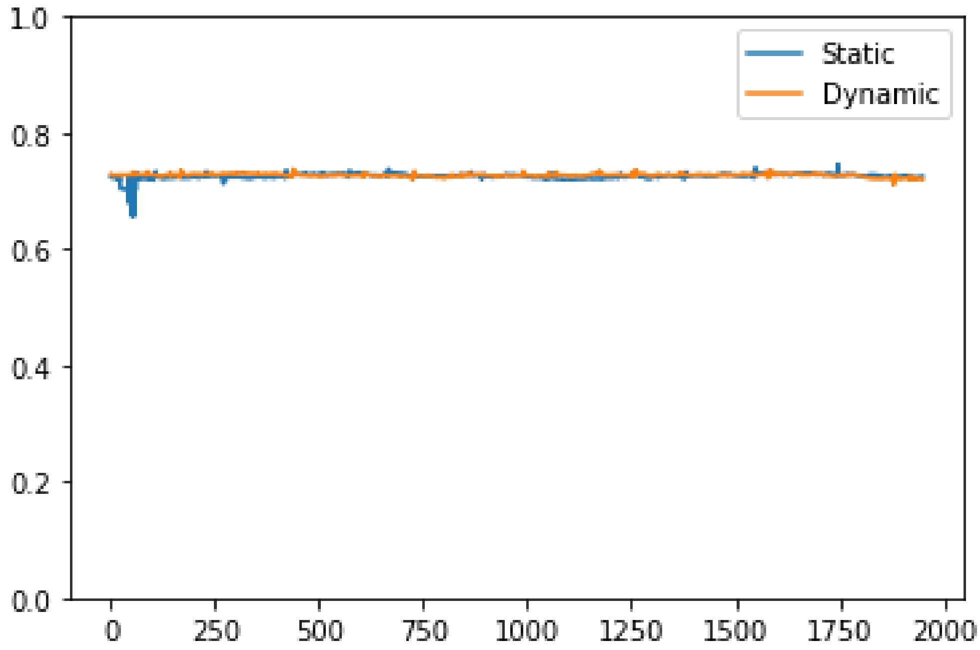

Figure18. Time Vs Y coordinate plot

3. X coordinate & Y coordinate

In the below graph the target point coordinates have been plotted, it's clear there's not much deviation in the static and dynamic setup for the participant A.

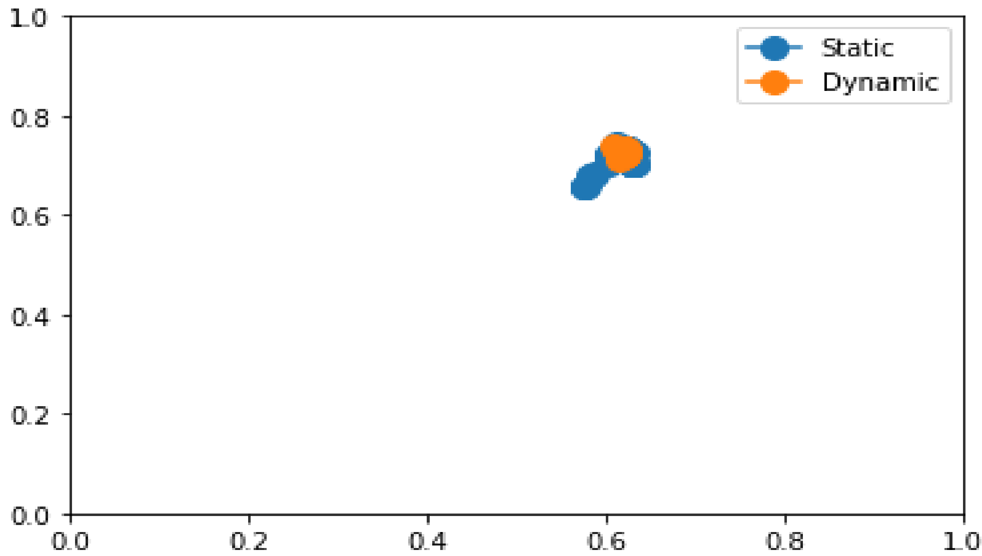

Figure19. X Vs Y coordinate plot

4. MSE

Mean Squared Errors were calculated for the participant A with the mean of the X and Y coordinate of static setup considered as the target point. Slight deviation in the X coordination was observed with a considerably high deviation in the y coordinate for participant A, refer the snipped image below.[20]

MSE

```
In [17]: X_mean = pupil_normal['norm_pos_x'].mean()
         mse = ((X_mean - pupil_normal['norm_pos_x']) ** 2).mean(axis = None)
         mse
Out[17]: 1.572233875549313e-05

In [18]: Y_mean = pupil_normal['norm_pos_y'].mean()
         mse = ((X_mean - pupil_normal['norm_pos_y']) ** 2).mean(axis = None)
         mse
Out[18]: 0.010592902470283664

In [19]: mse = ((X_mean - pupil_dynamic['norm_pos_x']) ** 2).mean(axis = None)
         mse
Out[19]: 2.0828498374033554e-05

In [20]: mse = ((Y_mean - pupil_dynamic['norm_pos_y']) ** 2).mean(axis = None)
         mse
Out[20]: 8.078930330821057e-06
```

Figure20. Mean Squared Errors for Participant A

Ttest

Ttest was conducted for the participant A, below is the snipped image of the result.

```
In [16]: from scipy.stats import ttest_ind
         t1 = ttest_ind(pupil_normal['norm_pos_x'], pupil_dynamic['norm_pos_x'])
         t1
Out[16]: Ttest_indResult(statistic=44.597405497232245, pvalue=0.0)
```

Figure21. Ttest Results for Participant A

4.2.2 Participant B

Below are the plots observed and analysis for participant B in both Static(without vection) and Dynamic(with vection) setups.

1. Time & X coordinate

In the following graph, it can be seen that participant B again showed not much variation in the graph in both the set ups.

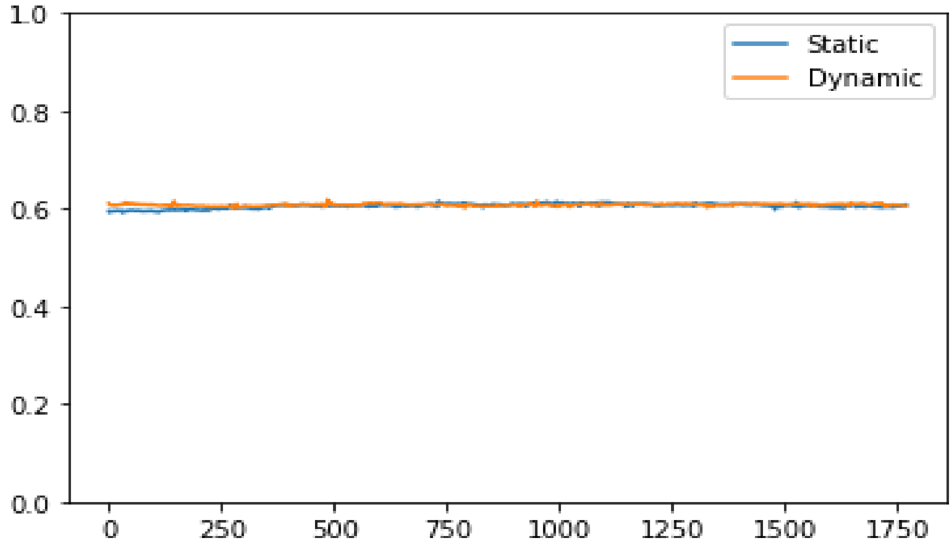

Figure22. Time Vs X coordinate plot for Static and Dynamic setups

2. Time & Y coordinate

Graph for the Y coordinate shows similar trend.

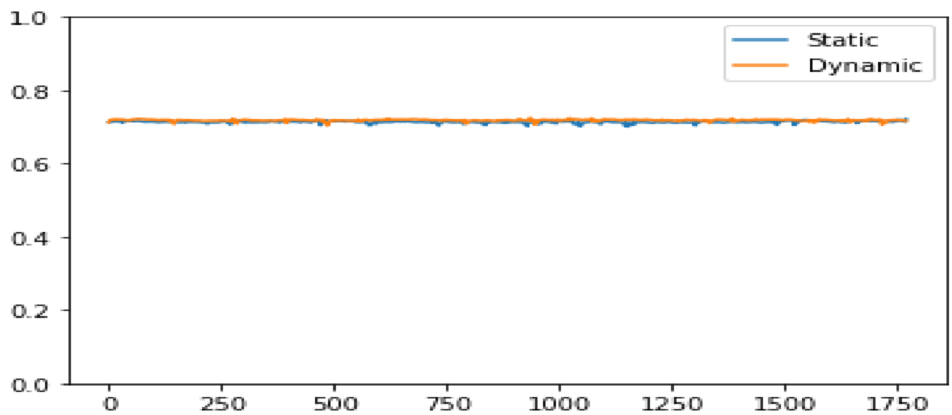

Figure23. Time Vs Y coordinate plot for Static and Dynamic setups

3. X coordinate & Y coordinate

In the below graph the target point coordinates have been analyzed and the plot is quite same in both the scenarios.

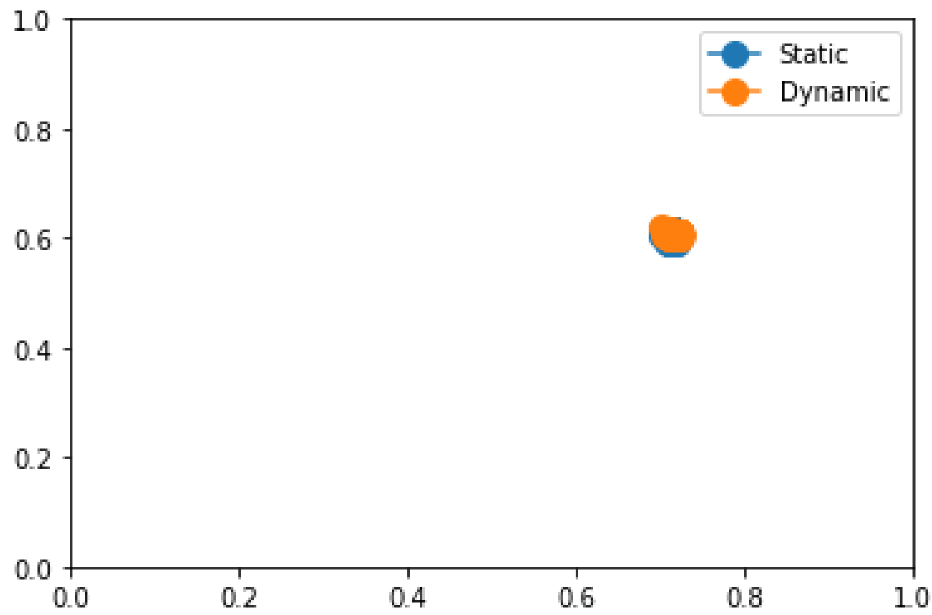

Figure24. XVs Y coordinate plot for Static and Dynamic setups

4. MSE

The Mean Squared Errors were calculated for the participant B with the mean of the X and Y coordinate of static setup considered as the target point. There was deviation in the X coordination and a considerably high deviation in the Y coordinate.

MSE

```
In [18]: X_mean = pupil_normal['norm_pos_x'].mean()
         mse = ((X_mean - pupil_normal['norm_pos_x']) ** 2).mean(axis = None)
         mse

Out[18]: 4.624232726348197e-06

In [19]: Y_mean = pupil_normal['norm_pos_y'].mean()
         mse = ((X_mean - pupil_normal['norm_pos_y']) ** 2).mean(axis = None)
         mse

Out[19]: 0.011617797476441268

In [20]: mse = ((X_mean - pupil_dynamic['norm_pos_x']) ** 2).mean(axis = None)
         mse

Out[20]: 2.665031462514830-05

In [21]: mse = ((Y_mean - pupil_dynamic['norm_pos_y']) ** 2).mean(axis = None)
         mse

Out[21]: 8.874767879073954e-06
```

Figure25. Mean Squared Errors for Participant B

5.Ttest

Ttest was conducted for the participant B, below is the snipped image of the result..

```
In [17]: from scipy.stats import ttest_ind
         t1 = ttest_ind(pupil_normal['norm_pos_x'], pupil_dynamic['norm_pos_x'])
         t1

Out[17]: Ttest_indResult(statistic=-69.73442182591334, pvalue=0.0)
```

Figure26. Ttest for Participant B

4.2.3 Participant C

Below are the plots observed and analysis for participant C in both Static(without vection) and Dynamic(with vection) setups.

1. Time Vs X coordinate

Slight deviation can be noticed in the X coordinate for the Participant C.Which could due to optical illusion as the vection slide was moving horizontally, though the deviation is not that strong.

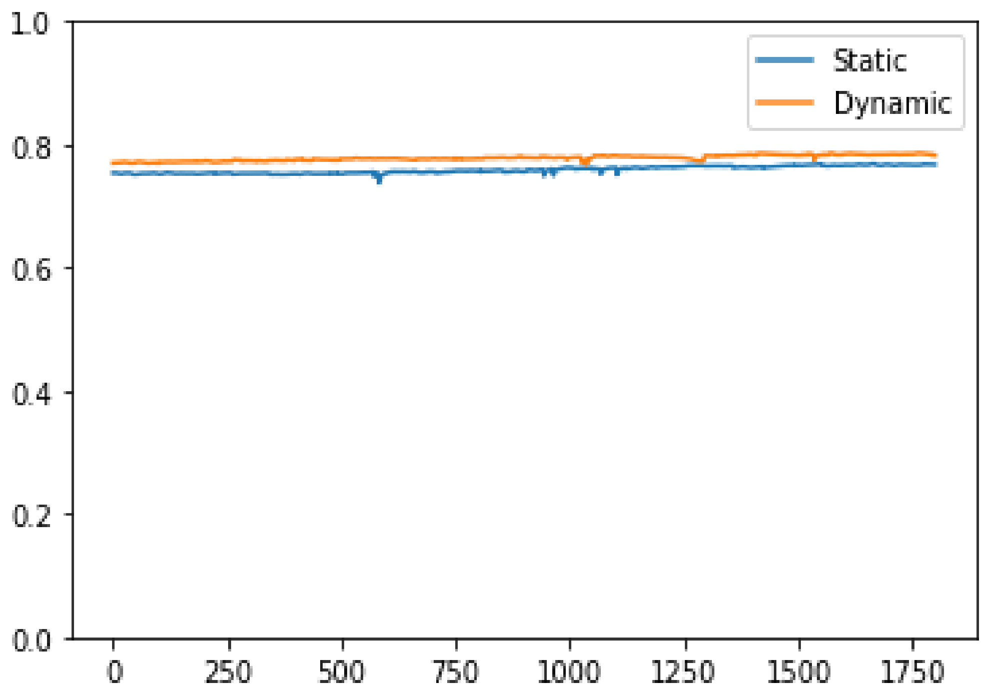

Figure27. Time Vs X coordinate plot in static and dynamic setups

2. Time Vs Y coordinate

Graph for the Y coordinate show comparatively less deviation, and right so as no deviation should happen in the Y coordinate.

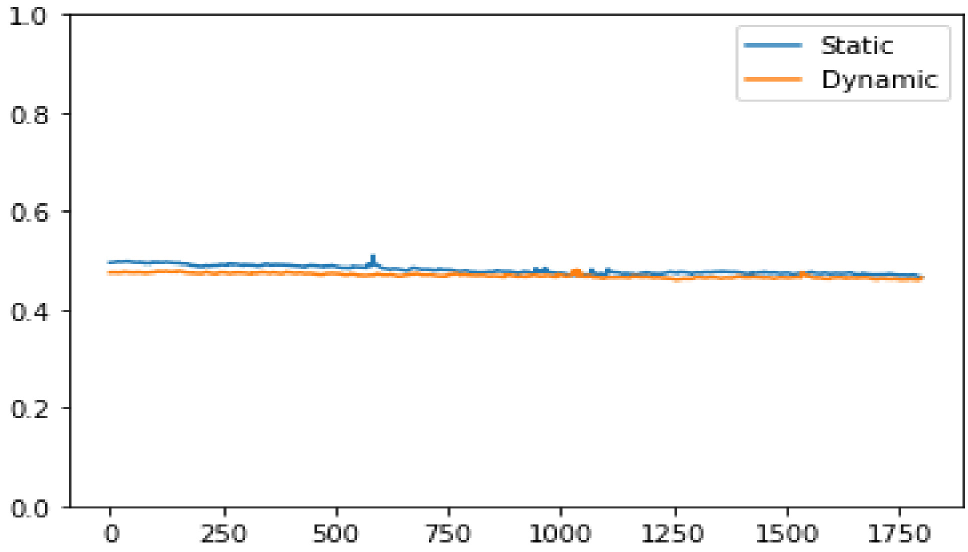

Figure28. Time Vs Y coordinate plot in static and dynamic setups

3. X coordinate & Y coordinate

The target point (x,y coordinates) of both the setups is different, that shows there was some effect caused in the vision by the vection slide.

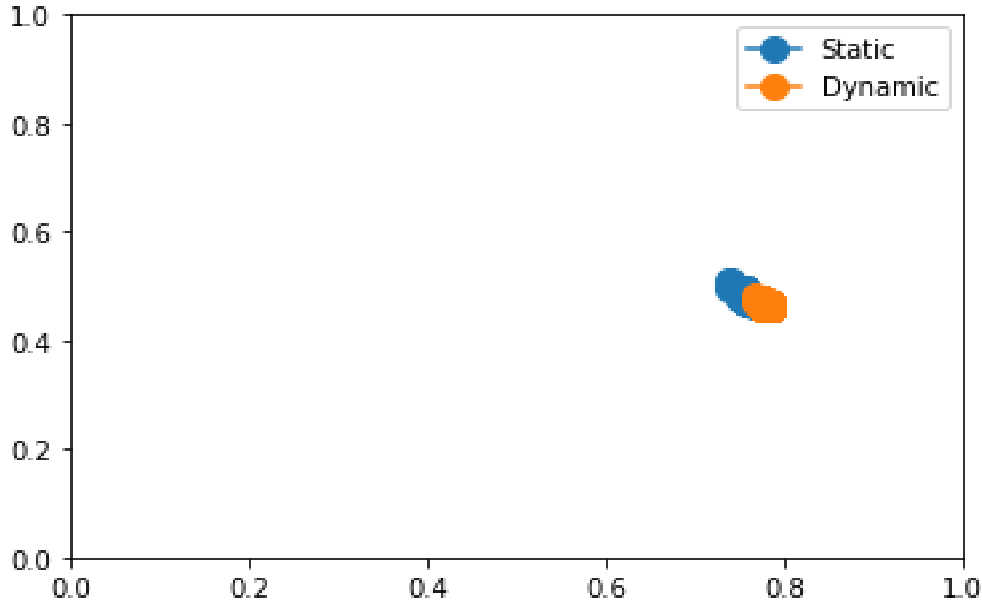

Figure29. X Vs Y coordinate plot in static and dynamic setups

4. MSE

Mean Squared Errors were calculated for both the participants, and as evident form the plots before the MSE for the X coordinate in both the static and dynamic case were different. While there's not much difference in the Y coordinate for both the cases.

MSE

```
In [17]: X_mean = pupil_normal['norm_pos_x'].mean()
         mse = ((X_mean - pupil_normal['norm_pos_x']) ** 2).mean(axis = None)
         mse

Out[17]: 3.1937537998420556e-05

In [18]: Y_mean = pupil_normal['norm_pos_y'].mean()
         mse = ((X_mean - pupil_normal['norm_pos_y']) ** 2).mean(axis = None)
         mse

Out[18]: 0.07802427686971934

In [19]:
         mse = ((X_mean - pupil_dynamic['norm_pos_x']) ** 2).mean(axis = None)
         mse

Out[19]: 0.00039106542292297904

In [20]:
         mse = ((Y_mean - pupil_dynamic['norm_pos_y']) ** 2).mean(axis = None)
         mse

Out[20]: 0.00014879222231906156
```

Figure30. Mean Squared Errors for Participant C

5. Ttest

Ttest was calculated and the value came below the significance level, there our null hypothesis was rejected.

ttest

```
In [16]: from scipy.stats import ttest_ind
         t1 = ttest_ind(pupil_normal['norm_pos_x'], pupil_dynamic['norm_pos_x'])
         t1

Out[16]: Ttest_indResult(statistic=-114.44034629299438, pvalue=0.0)
```

Figure30. Ttest Results for Participant C

4.2.4 Participant D

Below are the plots observed and analysis for participant D in both Static(without vection) and Dynamic(with vection) setups.

1. Time & X coordinate

Some deviation can be noted for X coordinate for the participant D.

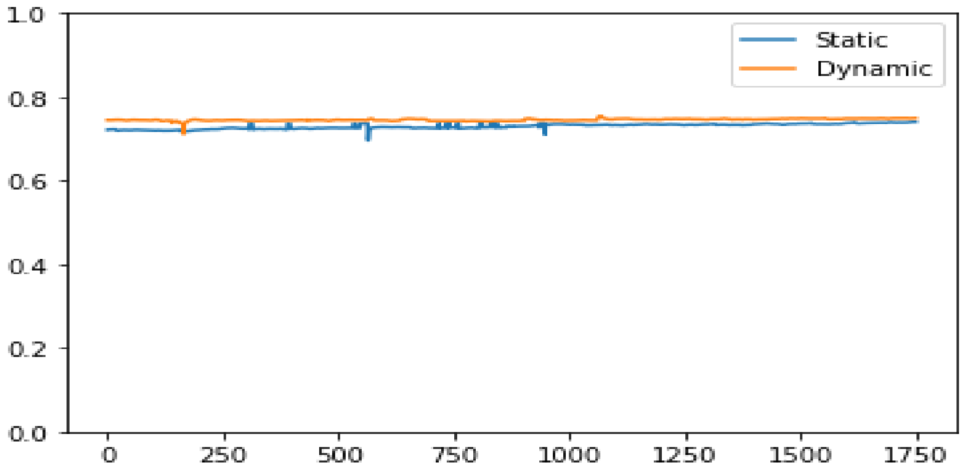

Figure31. Time vs X coordinate plot in static and dynamic setups

2. Time & Y coordinate

In the following graph for the participant C the Y coordinates during the static and dynamic setup are quite similar as there shouldn't be any vertical movement.

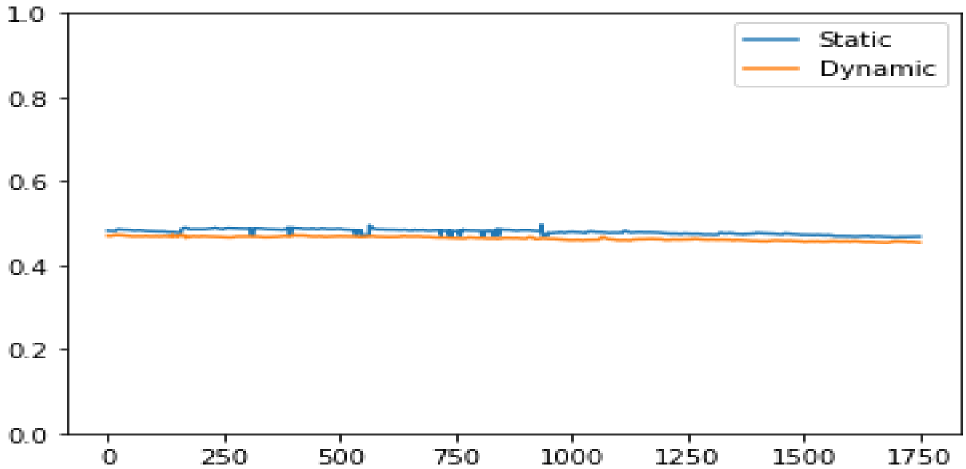

Figure32. Time vs Y coordinate plot in static and dynamic setups

3. X coordinate & Y coordinate

In the below graph the X and Y coordinates have been analyzed for both the setups and a slight deviation can be notice in target point point.

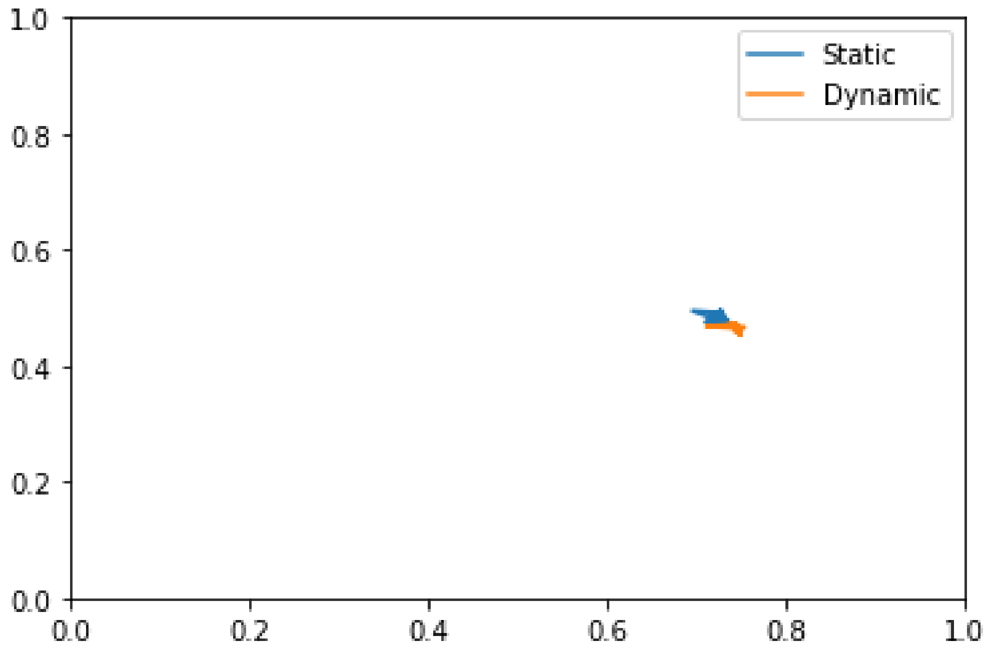

Figure32. X vs Y coordinate plot in static and dynamic setups

4. MSE

Mean Squared Error was calculated for the X&Y coordinates for both static and dynamic setups, the error in the X coordinate is quite noticeable which shows the effect of vection.

MSE

```
In [22]: X_mean = pupil_normal['norm_pos_x'].mean()
         mse = ((X_mean - pupil_normal['norm_pos_x']) ** 2).mean(axis = None)
         mse
Out[22]: 3.5879978270831306e-05

In [25]: Y_mean = pupil_normal['norm_pos_y'].mean()
         mse = ((X_mean - pupil_normal['norm_pos_y']) ** 2).mean(axis = None)
         mse
Out[25]: 0.06307303416166023

In [26]:
         mse = ((X_mean - pupil_dynamic['norm_pos_x']) ** 2).mean(axis = None)
         mse
Out[26]: 0.0002562866918652932

In [27]:
         mse = ((Y_mean - pupil_dynamic['norm_pos_y']) ** 2).mean(axis = None)
         mse
Out[27]: 0.000258031493757714
```

Figure33. Mean Squared Errors for participant D

5.Ttest

Ttest was conducted for the participant B, our null hypothesis was rejected as the pvalue came below the significance level that is 0.05. Therefore, it's quite evident that there was some no similarity between the two samples tested.Below is the snipped image of the result.

```
In [21]: from scipy.stats import ttest_ind
         ttest_ind(pupil_normal['norm_pos_y'], pupil_dynamic['norm_pos_y'],equal_var = False)

Out[21]: Ttest_indResult(statistic=82.1240815508042, pvalue=0.0)
```

Figure33. Mean Squared Errors for participant D

Chapter 5: Conclusion & Future Work

Eye tracking technology is still in the making, and technological advancements will always open more wide range of possibilities in understanding the eye movements. Low cost Devices with high sampling rates can aid the process; as it will allow to take forward the researches with better results, without spending a lot on the eye tracking device.[21]

We can see from the above analysis that the results could have been better for the Static Vs Dynamic setups if a high end eye tracking device was introduce providing high sampling rate and thus detecting the pupil positions more appropriately.

Further analysis on different virtual reality environment can bring out useful outcomes, which will definitely provide help to improve the virtual reality world.

Chapter 6: Reference

[1] N. J. Wade, “Pioneers of eye movement research.,” *Iperception.*, vol. 1, no. 2, pp. 33–68, 2010.

[2] H. Kenneth, “Sampling frequency and eye-tracking measures: how speed affects durations, latencies, and more,” *J. Eye Mov. Res.*, vol. 3, no. 3, pp. 1–12, 2010.

[3] P. Rosa, “History of eye tracking,” *INT.J.PSYCHOL.RES*, vol. 8, no. 2, pp. 90–103, 2015.

[4] “Analysis of Eye Movements in the Context of e-Learning Recommendations for Eye-efficient User Interfaces.”

[5] D. C. Richardson and M. J. Spivey, “Eye-Tracking: Characteristics and Methods Eye-Tracking: Research Areas and Applications Eye-Tracking: Characteristics and Methods.”

[6] “Pupil Labs - Pupil.” [Online]. Available: https://pupil-labs.com/pupil/. [Accessed: 01-Sep-2018].

[7] M. Kassner, W. Patera, and A. Bulling, “Pupil,” in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing Adjunct Publication - UbiComp ’14 Adjunct*, 2014, pp. 1151–1160.

[8] W. R. Kassner, Moritz Philipp; Patera, “PUPIL: constructing the space of visual attention,” pp. 1–181, 2012.

[9] H. Deubel and W. X. Schneider, “Saccade target selection and object recognition: evidence for a common attentional mechanism.,” *Vision Res.*, vol. 36, no. 12, pp. 1827–37, Jun. 1996.

[10] A. Bulling, D. Roggen, and G. Tröster, “Wearable EOG goggles: Seamless sensing and context-awareness in everyday environments,” *J. Ambient Intell. Smart Environ.*, vol. 1, no. 2, pp. 157–171, 2009.

[11] “The Python Standard Library — Python 2.7.15 documentation.” [Online]. Available: https://docs.python.org/2/library/. [Accessed: 01-Sep-2018].

[12] D. D. Wackerly, W. Mendenhall, and R. L. Scheaffer, *Mathematical statistics with applications*, 7th ed. Belmont, CA, USA: Thomson Brooks/Cole, 2008.

[13] “pandas.Series.interpolate — pandas 0.23.4 documentation.” [Online]. Available: https://pandas.pydata.org/pandas-docs/stable/generated/pandas.Series.interpolate.html. [Accessed: 01-Sep-2018].

[14] “Signal Processing – Apply Median Filters (Python) | machinelearning1.” [Online]. Available: https://machinelearning1.wordpress.com/2014/07/13/signal-processing-apply-median-filters-python/. [Accessed: 01-Sep-2018].

[15] “pandas.rolling_mean — pandas 0.17.0 documentation.” [Online]. Available: https://pandas.pydata.org/pandas-docs/version/0.17.0/generated/pandas.rolling_mean.html. [Accessed: 01-Sep-2018].

[16] V. Selimis, “Eye-Tracking Technology in Vehicles: Application and Design,” 2015.

[17] “THE PROBABLE ERROR OF A MEAN,” *Biometrika*, vol. 6, no. 1, pp. 1–25, Mar. 1908.

[18] “scipy.stats.ttest_ind — SciPy v1.1.0 Reference Guide.” [Online]. Available: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_ind.html. [Accessed: 01-Sep-2018].

[19] “T Test (Student’s T-Test): Definition and Examples - Statistics How To.” [Online]. Available: http://www.statisticshowto.com/probability-and-statistics/t-test/. [Accessed: 01-Sep-2018].

[20] “sklearn.metrics.mean_squared_error — scikit-learn 0.19.2 documentation.” [Online]. Available: http://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_squared_error.html. [Accessed: 01-Sep-2018].

[21] D. Li, J. Babcock, and D. J. Parkhurst, “openEyes: a low-cost head-mounted eye-tracking solution,” *Proc. 2006 Symp. Eye Track. Res. Appl.*, pp. 95–100, 2006.